Decoding non-invasive brain activity with novel deep-learning approaches

Richard Csaky

Christ Church College

University of Oxford

A thesis presented for the degree of

Doctor of Philosophy

Michaelmas 2023

I dedicate this to the memory of my grandma, Zori

It may be that our role on this planet is not to worship God, but to create him.

— Arthur C. Clarke

Acknowledgements

I would primarily like to acknowledge my supervisor Mark Woolrich. I had no idea of the kind of journey I would embark during my 3 PhD years, but Mark has given me the research freedom that few can enjoy. While at times I may have needed more guidance, as I learned later, stumbling in the darkness is part of the process. Mark always provided support and an intellectual clarity which I am still in awe of.

I am grateful to my two co-supervisors, Oiwi and Mats, for their support and involvement in my research. Without their sacrifice of long hours to help with experiments, Chapter 6 would have not been possible. Here I would also like to thank Anna Camera, and all OHBA members who helped with my experiments. I am humbled by their selflessness.

I am eternally indebted to Gabor Recski, my Bachelor's and Master's supervisor, who instilled in me fundamental research techniques that I used throughout my PhD. It is under his guidance that I ~~mastered~~ was introduced to python, deep learning, latex, and countless other tools, elemental to my research.

Thanks to Arun, Antara, Coby, Arina, Nati, Shu, Ryan, and Clara, for making the UK more homely, and for putting up with my ~~daily~~ hourly complaints about the weather. While I lost touch with some of them, I do not forget our time together.

Finally, I would like to thank my parents, and my friends from home, Adam, Janos, Come, and Peter, who have helped me through some dark times during my PhD. It's difficult to put into words the significance of their friendship.

Abstract

This thesis delves into the world of non-invasive electrophysiological brain signals like electroencephalography (EEG) and magnetoencephalography (MEG), focusing on modelling and decoding such data. The research aims to investigate what happens in the brain when we perceive visual stimuli or engage in covert speech (inner speech) and enhance the decoding performance of such stimuli. The findings have significant implications for the development of brain-computer interfaces (BCIs), leading to assistive communication technologies for paralysed individuals. The thesis is divided into two main sections, methodological and experimental work. A central concern in both sections is the large variability present in electrophysiological recordings, whether it be within-subject or between-subject variability, and to a certain extent between-dataset variability.

In the methodological sections, we explore the potential of deep learning for brain decoding. The research acknowledges the urgent need for more sophisticated models and larger datasets to improve the decoding and modelling of EEG and MEG signals. We present advancements in decoding visual stimuli using linear models at the individual subject level. We then explore how deep learning techniques can be employed for group decoding, introducing new methods to deal with between-subject variability. Finally, we also explores novel forecasting models of MEG data based on convolutional and Transformer-based architectures. In particular, Transformer-based models demonstrate superior capabilities in generating signals that closely match real brain data, thereby enhancing the accuracy and reliability of modelling the brain's electrophysiology.

In the experimental section, we present a unique dataset containing high-trial inner speech EEG, MEG, and preliminary optically pumped magnetometer (OPM) data. We highlight the limitations of current BCI systems used for communication, which are either invasive or extremely slow. While inner speech decoding from non-invasive brain signals has great promise, it has been a challenging goal in the field with limited decoding approaches, indicating a significant gap that needs to be addressed. Our aim is to investigate different types of inner speech and push decoding performance by collecting a high number of trials and sessions from a few participants. However, the decoding results are found to be mostly negative, underscoring the difficulty of decoding inner speech.

In conclusion, this thesis provides valuable insight into the challenges and potential solutions in the field of electrophysiology, particularly in the decoding of visual stimuli and inner speech. The findings could pave the way for future research and advancements in the field, ultimately improving communication capabilities for paralysed individuals.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

BCI	Brain-Computer Interface
MVPA	Multivariate Pattern Analysis
RSA	Representational Similarity Analysis
MEG	Magnetoencephalohraphy
EEG	Electroencephalography
OPM	Optically-pumped Magnetometer
SNR	Signal-to-Noise Ratio
fMRI	functional Magnetic Resonance Imaging
PCA	Principal Component Analysis
ICA	Independent Component Analysis
LDA	Linear Discriminant Analysis
HMM	Hidden Markov Model
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
GPT	Generative Pre-trained Transformer
AR	Auto-regressive
PFI	Permutation Feature Importance
STFT	Short-time Fourier Transform
LOSO	Leave-one-subject-out

1 | Introduction

1.1 Electrophysiology of the brain

1.1.1 Recording brain activity

Understanding the intricacies of the human brain remains one of the grand challenges of science, but what does it mean to understand? Does it entail accurately simulating certain functions computationally, as pursued in computational neuroscience (Dayan and Abbott, 2005)? Is it synonymous with prediction, as suggested by predictive coding theory (Friston, 2010)? Does understanding pertain to characterising an individual brain, an average brain, or a collection of distinct brain types (Kanai and Rees, 2011)? Might understanding enable novel treatments for brain disorders and enhancement of human abilities, aligning with the mission of translational neuroscience (Insel and Landis, 2013)?

As with most doctoral theses, this work does not attempt to resolve such expansive questions. Rather, it aims to provide incremental advancements in select research domains, with the aspiration that these innovations may one day contribute to a more comprehensive understanding. For our purposes, understanding may constitute elucidating particular processes in the brain and linking them to cognitive, emotional, or behavioural phenomena (Pessoa, 2022). Frequently this involves mathematical models that approximate the underlying biology (Izhikevich, 2007). We can equate these models themselves with understanding (Kriegeskorte and Douglas, 2018), although the use of deep learning in the models can complicate this notion, causing the model itself to require an additional interpretative effort (Arrieta et al., 2020).

The brain contains upwards of 86 billion neurons with quadrillions of synaptic connections (Azevedo et al., 2009). To achieve tractable levels of understanding, approximations and abstractions are imperative. We can delineate types of understanding by spatial and temporal scale (Panahi et al., 2021), reflecting the spatiotemporal physical essence of the brain. For example, modelling neurotransmitter dynamics aids comprehension of learning, motivation, and rewards - pivotal constructs in cognitive and behavioural neuroscience (Schultz, 2002). Single neuron models specify input-output characteristics on precise (millisecond) timescales, providing insights at the core of computational neuroscience (Izhikevich, 2004). Interconnecting such models allows understanding of local microcircuits. Validating models against experimental data reveals, for instance, which neuronal populations represent different parts of the visual field (retinotopic maps) (Wandell et al., 2007; Nasiotis et al., 2017). Such models also elucidate the rapid temporal propagation of activity across the hierarchical visual system (Carlson et al., 2013).

Deriving and evaluating models requires actual brain data. For fine spatial scales, this often comes from intracranial electrodes measuring individual neuron spiking or local field potentials (LFPs) (Buzsáki et al., 2012, 2015). Intracranial electrocorticography (ECoG) provides real-time population-level (10,000s of neurons) activity (Miller et al., 2009) by placing electrode grids directly on the brain surface. However, such invasive procedures carry risks associated with surgery and are typically only employed in clinical settings, or in research with patients who already require surgery for medical reasons (Waldert, 2016). This inevitably constrains data quantity and variety. At larger scales, neural mass models enable whole-brain biophysical simulations (Deco et al., 2008; Hadida et al., 2018), while machine learning can model diverse non-invasive recording modalities like electrophysiology or blood-oxygen-level-dependent (BOLD) imaging (Friston, 2005).

BOLD techniques including functional magnetic resonance imaging (fMRI) and functional near-infrared spectroscopy (fNIRS) offer the poorest temporal resolution, on the order of seconds. This is because they detect changes in blood flow resulting from variations in local neuronal activity, reflecting the understanding that the body cannot, and does not need to regulate blood flow on the order of milliseconds (Buxton, 2013). However, their spatial resolution is unparalleled, producing dynamic 3D brain images with hundreds of thousands to millions of voxels (Buxton, 2013). Voxel volumes are around $0.5mm^3$, providing localised activity estimates in the form of BOLD changes. On the downside, MRI scanners are sensitive to head motion and place the participant in a constrained noisy space, potentially influencing brain function (Van Dijk et al., 2012).

1.1.2 Non-invasive electrophysiology

This thesis is concerned with non-invasive electrophysiological recording modalities, the modelling of such data, and the kind of understanding and real-world applications that these models might facilitate. The modality inherently limits the spatial and temporal scale of modelling and understanding (Kiebel et al., 2008). Non-invasive electrophysiology like electroencephalography (EEG) and magnetoencephalography (MEG) offers millisecond temporal resolution akin to intracranial recordings (Cohen, 1968; Berger, 1929). This results from measuring near-instantaneous electromagnetic fields generated by neuronal activity (Nunez and Srinivasan, 2006). However, limited spatial resolution remains a key challenge, especially for EEG which measures electrical currents. Signal distortion by the skull and scalp restricts spatial specificity (Nunez and Srinivasan, 2006). While MEG is less affected due to measuring magnetic fields, its few hundred sensors still average over millions of neurons (Hämäläinen et al., 1993). MEG also necessitates costly specialised equipment in magnetically shielded rooms (Baillet, 2017). Emerging optically pumped magnetometers (OPMs) may improve MEG sensitivity and flexibility by enabling on-scalp measurements (Wens, 2023).

Their lack of required cooling could expand MEG accessibility (Boto et al., 2018), potentially enabling brain-computer interface (BCI) applications, although current OPM technology does require the devices to be housed in a magnetically shielded room.

Despite advances, modelling and decoding brain signals from non-invasive electrophysiology remains challenging. Models often fail to accurately decode complex, variable signals across and within individuals (Saha and Baumert, 2020). While some variability naturally arises from morphological and dynamical diversity (Michel and Brunet, 2019; Wainio-Theberge et al., 2021), noise also contributes (Faisal et al., 2008). Signal-to-noise ratio is thus crucial (Nenonen et al., 2007), with artefacts originating from external sources (e.g. power lines, Earth's magnetic field) or internal ones like breathing, blinks, heartbeats, and muscle activity (Urigüen and Garcia-Zapirain, 2015). In later chapters, we discuss artefact reduction via signal processing and machine learning (Makeig et al., 1995). Attention, fatigue, anatomy, and functional differences also modulate brain recordings (Saha and Baumert, 2020), complicating generalisation across individuals or sessions. Managing variability in electrophysiological data is an integral theme of this thesis.

One may rightly question how such noisy, spatially-coarse signals can inform understanding or applications. However, the real-time nature, direct neural basis, and non-invasiveness of M/EEG enable massive datasets with exquisite temporal resolution (Gifford et al., 2022). This makes EEG uniquely suited for BCIs in healthy and ill populations (Murguialday et al., 2011). MEG and OPMs currently remain confined to research due to the shielding requirement, and fNIRS (Naseer and Hong, 2015), while up and coming in the BCI field is too slow to be used in the kinds of BCI applications we are interested in the experimental work part of this thesis.

As our research is not concerned with the development of new kinds of recording technology, we will have to carefully consider the limitations inherent in non-invasive electrophysiology in our methodological work. These constraints inform the kind of questions we can ask (and hopefully answer), and the kind of understanding we can gain (Kiebel et al., 2008). The whole-brain view of M/EEG facilitates studying dispersed cognitive processes that recruit vastly different brain regions like vision and language, as we will see. We can characterise spatiotemporal dynamics with millisecond precision relative to events (Baillet et al., 2001), albeit with limited spatial specificity. Critically, only synchronised activity across tens of thousands of neurons overcomes noise to manifest in M/EEG (Singer, 1999; Buzsaki, 2006; Buzsaki and Draguhn, 2004). As we will see this consideration is particularly important when studying subtle cognitive processes such as inner speech (Alderson-Day and Fernyhough, 2015). To reiterate an important point, one way to deal with the natural variability of ongoing brain signals (Wainio-Theberge et al., 2021) and artefactual contamination is to collect as much data as possible from a single individual (Boudewyn et al., 2018). Repeated measurements allow both better spatial and better temporal specificity, and consequently a deeper level of understanding (Hebart et al., 2022). However, better methods are needed to deal with variability and fully capitalise on the rich information in M/EEG data (Quinn et al., 2022a; Hebart et al., 2022; van Vliet and Salmelin, 2020).

A major opportunity with non-invasive electrophysiology is accumulating large datasets. However, between-subject variability hinders understanding brain activity beyond the individual level (Olivetti et al., 2014). The success of deep learning with large sets of data motivates the need to unlock the full potential of electrophysiology datasets by developing new methods for dealing with between-participant variability (Défossez et al., 2022; Kostas et al., 2021).

This thesis aims to address the above challenges by exploring the potential of deep learning techniques in the decoding and modelling of electromagnetic brain signals. We delve into these issues through methodological and experimental work seeking to advance electrophysiology. In addition to providing a deeper understanding of the brain (or brains), we hope our findings will contribute to the development of more accurate and reliable BCIs.

1.2 Visual and language processing

Besides ongoing neural activity during rest (Raichle et al., 2001), the brain must also process and react to external stimuli, including complex inputs such as vision and language. Elucidating the neural dynamics underlying these faculties is fundamental in neuroscience and is important for brain-computer interfaces (BCIs) (İşcan and Nikulin, 2018; Akbari et al., 2019). However, decoding brain signals during such tasks poses multiple challenges that this thesis tackles. As discussed previously, these stem principally from various forms of variability.

Visual processing involves a hierarchical cascade that originates in the retina, travelling along the optic nerve to the primary visual cortex (V1) to extract basic features such as orientation and spatial frequency (Hubel and Wiesel, 1962). This early stage, focused on low-level attributes such as edges and colour, has been thoroughly characterised (Livingstone and Hubel, 1988). Information then flows to extrastriate areas including V2, V3, V4, and inferotemporal cortex for processing complex features like object identity (Tanaka, 1996), motion (Albright, 1984), and spatial location (Maunsell and Newsome, 1987).

The spatiotemporal dynamics and spectral signatures of visual processing can be examined using electrophysiological techniques like EEG and MEG (Baillet, 2017). For example, visual stimuli can evoke specific amplitude peaks in the signal precisely time-locked to stimulus onset (evoked activity). Due to time-locking, by averaging over individual trials the evoked peaks become more prominent and ongoing oscillations are cancelled out. Evoked responses triggered by changes in the visual field manifest very rapidly, within 100 ms post-stimulus (Cichy et al., 2014, 2016). Stimuli can also more broadly modulate oscillatory activity in the alpha (8-12 Hz) and gamma (30-80 Hz) frequency bands, reflecting processes like attentional modulation (Jensen and Mazaheri, 2010) and object recognition (Tallon-Baudry and Bertrand, 1999; Gruber et al., 1999), respectively. This induced activity plays a critical role in coordinating neural processing and integration of visual information (Klimesch, 2012; Herrmann et al., 2010).

In contrast, language processing relies on a distributed network that includes classical perisylvian language areas such as Broca's area for speech production and Wernicke's area for comprehension, among others (Friederici, 2011; Hickok and Poeppel, 2007). These regions are interconnected via white matter tracts, forming an integrated system for linguistic processing. Like visual processing, language tasks also elicit specific oscillatory activity patterns. Theta band (4-8 Hz) oscillations, for instance, have been linked to syllable segmentation and sentence parsing (Bastiaansen and Hagoort, 2006), while gamma oscillations reflect phonetic and semantic processing (Obleser and Kotz, 2011). The precise timing of these oscillations is believed to play a critical role in the coordination of neural dynamics during language tasks (Peelle and Davis, 2012).

(Dikker et al., 2020) have presented the detailed spatiotemporal dynamics evoked in language processing. Spoken word processing starts in Heschl's gyrus and the superior temporal gyrus 50-100 ms after stimulus. Written word processing starts in the occipital lobe 100 ms post-stimulus and goes on to the posterior and anterior fusiform gyrus for orthographic and morphological segmentation. Modality-independent processing happens 300-500 ms post-stimulus with lexical access and word meaning in the middle temporal gyrus. Semantic processing takes place 350-500 ms post-stimulus in orbito-frontal areas, and finally syntactic processing at around 600 ms in the inferior frontal gyrus. This late syntactics related activity reflects sentence-level integration.

We could conclude that the spatiotemporal dynamics of language and vision are quite well understood based on hundreds of studies with both invasive and non-invasive recordings. As we are motivated by BCI applications, we wanted to design methods that can improve decoding performance while still offering the kind of spatiotemporal understanding that has been established. A major issue is the limited performance of BCIs that allow a subject to communicate. Current systems are either invasive, posing risks to the user (Chaudhary et al., 2016), or non-invasive but extremely slow, limiting their practical use. Non-invasive BCIs often rely on slow and effortful control signals, such as the P300 wave or cortical potentials, which can be difficult to control and may require extensive training (Birbaumer et al., 1999; Farwell and Donchin, 1988; Lebedev and Nicolelis, 2006; Wolpaw, 2013). To provide improvements in BCI applications we wanted to tackle both the fundamental decoding methods (Lotte et al., 2018), and offer new ways of using BCIs, such as inner speech (Martin et al., 2018).

1.3 Thesis outline

One of the main challenges in decoding brain signals is the variability and complexity of these signals (Saha and Baumert, 2020). Multivariate pattern analysis (MVPA) of MEG and EEG data can be a valuable tool for understanding how the brain represents and discriminates between different stimuli (Guggenmos et al., 2018; King and Dehaene, 2014). However, traditional decoding models, such as linear, pairwise, sliding window decoding models, can be computationally intensive and may have limited decoding performance (Higgins et al., 2022b,a). These models typically focus on identifying the spatial and temporal signatures of stimuli, but they may not fully capture the complex patterns of brain activity during visual and language tasks. In contrast, full-epoch decoding models, commonly used for BCI applications, can provide better decoding performance but lack methods to interpret the contributions of spatial and temporal features (Haufe et al., 2014; Lotte et al., 2018). To address these challenges, we propose an approach that combines a multiclass, full epoch decoding model with supervised dimensionality reduction. This approach allows us to reveal the contributions of spatiotemporal and spectral features using permutation feature importance, while achieving higher decoding accuracy than traditional sliding window decoders.

Moving to multi-subject datasets will require decoding methods to be able to deal with high amounts of between-subject variability (Varoquaux et al., 2017; Poldrack et al., 2009). Decoding is typically subject-specific and does not generalise well over subjects (Olivetti et al., 2014; Dash et al., 2020a). Naive group modelling approaches have been proposed where a single model is trained on the data from multiple subjects. Due to high amounts of between-subject variability these methods typically perform much worse than subject-dependent modelling (Olivetti et al., 2014; Li et al., 2021; Saha and Baumert, 2020). Techniques that can overcome this will not only provide richer neuroscientific insights but also make it possible for group-level models to outperform subject-specific models. Here, we propose a method that uses subject embedding, analogous to word embedding in Natural Language Processing (NLP) (Mikolov et al., 2013a), to learn and exploit the structure in between-subject variability as part of a decoding model. We apply this method to MEG data from a visual task and show that the combination of deep learning and subject embedding can close the performance gap between subject and group-level decoding models.

The final kind of variability lies in the types of datasets collected and experimental conditions used. If we truly want to leverage big data in a brain decoding context, then we will have to deal with this additional layer of variability. Despite the potential of deep learning techniques in decoding brain signals (Schirrmeister et al., 2017a), there is little work on training large unsupervised models on brain data and then fine-tuning these for specific decoding tasks (Kostas et al., 2021). This approach has seen massive success in the deep learning field, for various kinds of data, e.g. images, language, audio (Devlin et al., 2019a; Krizhevsky and Sutskever, 2012; Hinton et al., 2012). We hypothesise that Wavenet (van den Oord et al., 2016) and Transformer (Vaswani et al., 2017) models can more accurately predict future timesteps than a linear model, and that these models can capture the spectral properties and long-range spatiotemporal dynamics of the data more accurately. Thus, they serve as solid foundation models to deal with between-dataset and between-task variability. We also propose that pre-trained forecasting models can be used to improve downstream decoding performance, much like in NLP (Radford et al., 2018).

We hope the aforementioned methodological advancements can contribute to improving BCI systems. However, we also wanted to ask whether faster BCI communication can be achieved with previously untapped modalities, such as inner speech (Brumberg and Guenther, 2010). Inner speech refers to the inner voice inside the head that governs our thoughts, specifically when thoughts take the form of language (Morin, 2005; Alderson-Day and Fernyhough, 2015). Despite the prevalence of inner speech in everyday life, research on this has been limited, particularly when it comes to non-invasive methods (Panachakel and Ramakrishnan, 2021; Dash et al., 2020a). Our proof of concept work aims to fill this gap by using EEG and MEG to collect data from three different inner speech paradigms and by conducting an initial decoding analysis. We aim to investigate the decoding performance of inner speech in EEG and MEG with a large number of per-participant trials, the transferability of decoders across sessions and tasks, and the comparison of OPM decoding performance and spatiotemporal dynamics to EEG and MEG.

In conclusion, besides analysing a new modality for BCI communication, inner speech, we address the various types of variability that hinder BCI decoding performance and applicability. We aim to leverage large datasets and deep learning to deal with within-participant, between-participant, and between-dataset variability.

The remainder of this thesis is organised as follows:

> **Chapter 2** introduces key concepts in electrophysiological data processing and modelling. This includes signal processing, the use of machine learning in unsupervised modelling, encoding, and decoding, and methods for interpreting such models.
>
> **Chapter 3** introduces several solutions for unifying the fields of multivariate pattern analysis and brain computer interface decoding, mainly focusing on linear full-epoch multiclass models and the uncovering of spatiotemporal and spectral information through permutation feature importance. This chapter is part of a published paper (Csaky et al., 2023a).
>
> **Chapter 4** presents a new method termed subject embedding to deal with between-subject variability in group-level decoding models. It investigates how this subject embedding and deep learning contribute to better group-level modeling. This chapter is part of a published paper (Csaky et al., 2023b).
>
> **Chapter 5** introduces deep learning methods for unsupervised modelling (forecasting) of MEG data. It is shown that specifically Transformer-based models are capable of accurately generating the spatiotemporal dynamics of real data. We investigate how these capabilities arise through a series of

ablations.

Chapter 6 presents our proof of concept work on inner speech with EEG and MEG. We discuss our experimental pipeline, data collection, data analysis, and decoding results, offering new insights into the decoding of inner speech. Preliminary OPM data is also presented along with a comparison of decoding performance with more standard modalities.

Chapter 7 discusses the implications of our findings and possible future directions for this research and the field of non-invasive brain decoding.

2 | Modelling and decoding electrophysiology

2.1 Machine learning

Machine learning (ML) refers to the automated discovery of models from data. Instead of manually specifying model parameters based on prior knowledge, as with conventional modelling approaches, ML algorithms learn (infer) these parameters directly from data. By learning from data, ML models can capture complex patterns and relationships that may be difficult to a priori specify. Thus, they are well suited to the complexities of high-dimensional whole-brain electrophysiology data. Compared to more biologically inspired modelling, ML is abstract and does not necessarily explain the underlying physical phenomena. For predictive power and abstraction, we are trading interpretability. This has enabled breakthroughs across various domains, from computer vision (Krizhevsky and Sutskever, 2012) to natural language processing (Vaswani et al., 2017).

2.1.1 Key components

At its core, ML comprises four key components: (1) data, (2) model specification, (3) learning objective, and (4) learning algorithm (Richards et al., 2019). Carefully considering the interplay between these components is crucial for successfully applying ML. We discuss each of these in turn below.

Data The data used for training is the primary basis for everything an ML model learns. For neuroimaging applications, this typically comprises multivariate time series data reflecting brain activity across multiple spatial locations. Data quality and curation is thus critical. Brain data is often accompanied by synchronised behavioural or task-related data. Electrophysiology data exhibits substantial within-subject variability from noise and artefacts (Bigdely-Shamlo et al., 2015), as well as between-subject variability in anatomical and functional characteristics (Bijsterbosch et al., 2018). Capturing this variability sufficiently during training is key for the model to generalise well across time and subjects. Preprocessing to clean, normalise and extract relevant signals is also important.

Model The model defines the computational architecture relating inputs to outputs. Choosing an appropriate model class is guided by domain expertise about expected relationships in the data, as well as trade-offs between flexibility, interpretability and trainability. For example, linear models have limited flexibility but are simple and interpretable. Deep neural networks are extremely flexible functional approximators (Hornik et al., 1989b) but lack interpretability. Recent work has shown the efficacy of deep networks for learning spatiotemporal relationships from neuroimaging data (Gohil et al., 2022).

Learning objective The objective quantifies the model's performance on a desired task, providing feedback to drive learning. Objectives may balance various constraints like accuracy, interpretability, and biological plausibility. For example, in regression tasks, a common objective is minimising the mean squared error between predicted and true outputs. In classification, maximising accuracy or minimising cross-entropy loss are typical. Additional objectives can be added to impose further constraints, e.g. regularisation (Gohil et al., 2022).

Learning algorithm The algorithm optimises model parameters (iteratively) to improve the objective. Algorithms like (stochastic) gradient descent (Bottou, 2010), genetic algorithms (Goldberg and Deb, 1991), and reinforcement learning (Sutton et al., 1998) have proven effective for various models and tasks. Factors like scalability, speed of convergence, and avoidance of local optima guide algorithm selection. For deep networks, backpropagation with stochastic gradient descent underpins most state-of-the-art approaches (Goodfellow et al., 2016).

Together, these components define the ML modelling approach. In the following, we discuss key concepts and strategies for effectively applying ML to brain data.

2.1.2 Fundamentals

ML offers a data-driven approach for uncovering structure in complex, multivariate brain data. However, simply throwing large models and datasets at a problem does not guarantee success. Thoughtfully considering the bias-variance trade-off, model capacity, regularisation, and cross-validation strategies is important for robust, generalisable models (Hastie et al., 2009).

Perhaps the most important component is the learning objective as this prescribes our goals in mathematical form. Learning objectives can often also be used as a direct metric of goodness, i.e. how well the model accomplishes the prescribed task. An important concept in arriving at such an assessment is the bias-variance trade-off, which provides us with a statistical framework (Vapnik, 1999). From a probabilistic point of view, data represents some probability distribution. Thus, machine learning can be framed as coming up with a model that best captures the training data distribution. Often, we want this model either to be as simple as possible, or to generalise to examples which are not part of the training data. These examples can be either within or outside of the training distribution, prescribing different levels of generalisation (Geirhos et al., 2018).

Bias-variance trade-off The bias-variance trade-off represents the opposing goals of fitting the training data well (variance error) while being able to generalise to new examples (bias error). Models with insufficient capacity are prone to

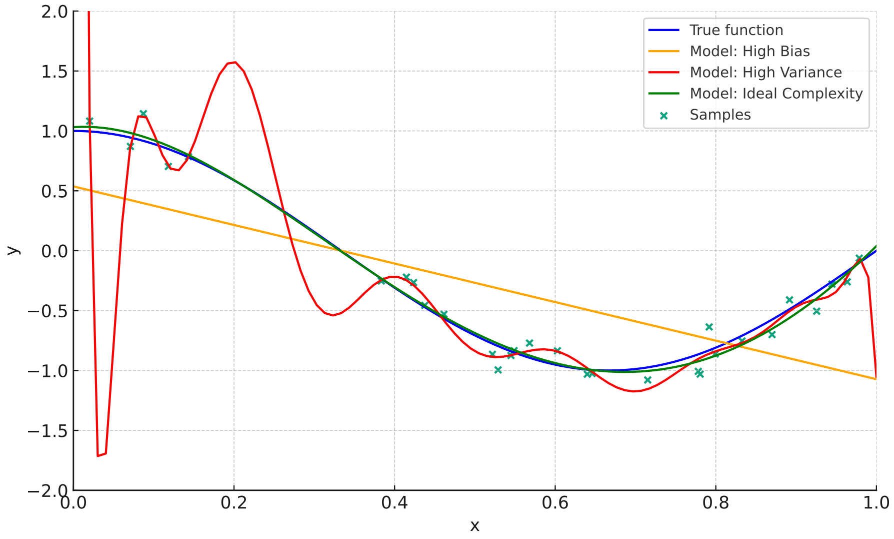


Figure 2.1: Visualisation of the bias-variance trade-off through a polynomial regression problem. The high-degree polynomial (red) fits the noise in our training samples (green cross) and thus has high variance. In contrast the high-bias linear model (orange) cannot capture the underlying distribution. The optimal model (green) achieves a trade-off between bias and variance and fits the true function (blue) well, however some differences may remain due to the irreducible noise in our samples.

high bias. Overly complex models overfit the noise in the training data, causing high variance. The ideal model balances bias and variance while also minimising irreducible error from noise (Geman et al., 1992). This trade-off is visualised through a simple regression problem in Figure 2.1.

Overfitting Overfitting happens when the model complexity is so high that it can model the noise as well as the signal in the training data (Ying, 2019). Noise means that the samples in our training data are not fully representative of the true underlying distribution we are trying to model. This is sometimes a consequence of not having the right data, i.e. EEG signals are affected by various noise sources, and always a consequence of not having enough data. Fully defining the underlying distribution would require an infinite set of examples, but our training data is always a subsample of this.

Model capacity To deal with noise, overfitting and generalisation, in practise we must employ certain assumptions about the underlying distributions, or collect better and more data. Capacity reflects the model's ability to fit diverse functions. High capacity can improve the fit to training data, but risks overfitting. Capacity depends on factors such as number of parameters and nonlinearity (Belkin et al., 2019). Deep neural networks derive immense capacity through multiple nonlinear layers (Raghu et al., 2017). But this needs to be carefully controlled, often requiring large training sets. Simpler linear models have limited capacity. Model selection aims to find the optimal capacity for a given dataset size. The effects of model capacity, overfitting, and regularisation are shown in Figure 2.2.

Regularisation Constraining model complexity through regularisation or early stopping helps to control variance. Simplicity can be achieved by limiting the number of parameters a model employs, putting mathematical constraints such as linearity, or other forms of regularisation. The latter discourages overfitting by penalising model complexity (Goodfellow et al., 2016). ℓ_2 regularisation adds a penalty proportional to the sum of squared parameters to the objective. This shrinks parameters toward zero, effectively constraining the capacity. Dropout randomly omits subsets of activations in neural networks during training as a form of implicit regularisation (Srivastava et al., 2014). Other approaches limit parameter ranges or enforce smoothness. The degree of regularisation is tuned to balance under and overfitting.

Cross-validation Rigorously testing ML models on held-out data is key to ensuring that they generalise beyond the training set. Cross-validation divides the

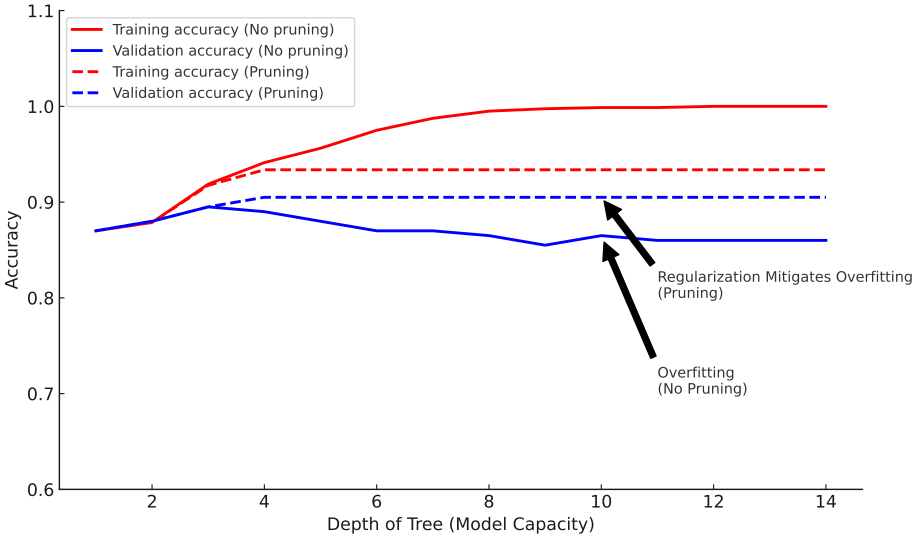


Figure 2.2: Visualisation of generalisation, overfitting, model capacity, and regularisation on a toy decision tree classification problem. As model capacity increases (depth of tree), both train and validation (generalisation) accuracy improve, up to a certain point after which overfitting occurs, and generalisation performance worsens. When adding regularisation (pruning) to the models (dashed lines), overfitting can be avoided even with larger models.

data into training and test sets multiple times to assess performance across different partitions (Arlot and Celisse, 2010). More sophisticated validation schemes include nested cross-validation for hyperparameter tuning and leaving out entire subjects for evaluating generalisability across individuals.

These are important considerations in our quest for models that can deal with the various types of variability inherent to electrophysiology data. Variability within an individual arises from noise contamination, and we would like to design models capable of generalising to similar events across time. Variability between individuals arises from subtle differences in data distributions, necessitating the need for capturing the underlying (more complex) distribution over multiple brains. Variability over datasets and tasks again expands the distribution of electrophysiological data in non-obvious ways.

2.1.3 Model categories

Various categorisations of modelling approaches can be made based on the learning paradigm, model flexibility, model form, and nature of predictions. These categorisations provide a useful framework for selecting appropriate techniques for modelling brain data.

In terms of learning paradigm, models can be *supervised*, *unsupervised*, or *self-supervised* (Goodfellow et al., 2016). In supervised learning, the model is trained on a set of input-output pairs with the goal of learning a mapping from inputs to outputs. In contrast, unsupervised learning involves only inputs, with the aim of uncovering latent structure such as clusters or factors (Ghahramani, 2003). Self-supervised learning employs inputs to generate proxy labels that are then used for doing supervised learning. For instance, autoregressive modelling uses past brain activity to predict future activity (Oord et al., 2018).

Another categorisation considers model *flexibility*: *linear* versus *nonlinear* models. Linear models assume a linear relationship between inputs and outputs, while nonlinear models make no such assumption. Although nonlinear models are more flexible and can capture complex relationships, they risk overfitting and reduced interpretability.

Generative models learn the joint distribution $p(x, y)$ over data x and target labels y, and can synthesise new data samples. Linear discriminant analysis (LDA) is a simple linear generative model. *Discriminative* models learn the conditional distribution $p(y|x)$ to predict outputs from inputs. Generative models (e.g., LDA) can also be employed for prediction in a classification task, by applying Bayes' theorem to the learned joint probability distribution (Murphy, 2012). Neural networks are most often formulated as discriminative models. Discriminative modelling is advantageous when the relationship between inputs and outputs is well-defined and there is abundant amount of data to avoid overfitting. However, with increasing amounts of noise, variability, and data scarcity, modelling the (generative) joint probability may provide useful assumptions and constraints.

Together, these categorisations delineate the extensive range of modelling techniques applicable to brain data. Careful consideration of the properties of the data and scientific question can guide selection of suitable approaches. In the following sections we detail common machine learning architectures and their application to M/EEG data.

2.2 Electrophysiological data analysis

2.2.1 Data characteristics

Electroencephalography (EEG) and magnetoencephalography (MEG) are two of the most prevalent non-invasive techniques for measuring brain activity with high temporal resolution. EEG electrodes placed on the scalp detect microvolt-level electrical potentials generated by synchronised postsynaptic potentials in neural populations (Teplan et al., 2002). Standard clinical EEG uses the 10–20 system with 19 electrodes, while high-density EEG utilises up to 256 electrodes to achieve higher spatial resolution (Fiedler et al., 2022). Typical EEG frequency bands include delta (1-4 Hz), theta (4-8 Hz), alpha (8-12 Hz), beta (12-30 Hz), and gamma ($>$30 Hz). Different bands have been linked to various cognitive and behavioural states. For instance, alpha waves reflect relaxed or idle cortical states whereas gamma activity is involved in active processing (Wang, 2010).

MEG detects femtotesla-level magnetic fields induced by postsynaptic currents, providing millisecond temporal resolution. It uses superconducting quantum interference devices (SQUIDs) to detect the minute neuromagnetic fields emanating from the head (Baillet, 2017). MEG sensor arrays typically contain around 300 sensors housed in a liquid helium Dewar. Magnetically shielded rooms are required to attenuate environmental magnetic noise and enable MEG systems to detect the brain's weak magnetic signals (Hämäläinen et al., 1993). EEG and MEG offer complementary information, with MEG more sensitive to tangential sources and EEG to radial sources (Baillet, 2017). Both modalities provide millisecond temporal resolution critical for tracking rapid neural dynamics.

Recently, optically pumped magnetometers (OPMs) have emerged as a novel MEG sensor technology. OPM-MEG systems utilise an array of OPM sensors that can be placed directly on the scalp, providing higher sensitivity and spatial resolution compared to traditional SQUID sensors (Boto et al., 2018).

2.2.2 Typical preprocessing steps

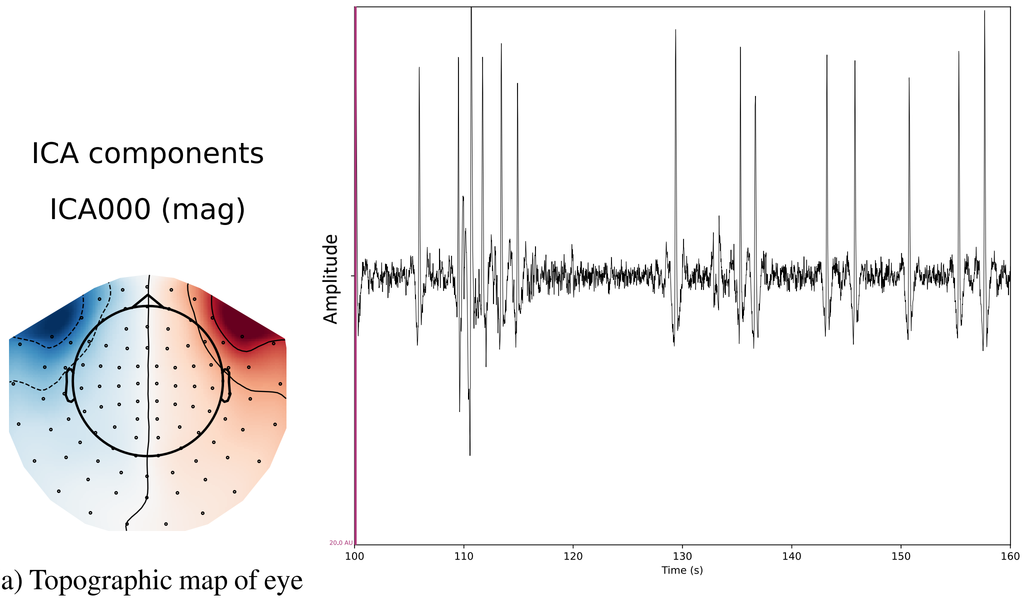
Standard preprocessing of electrophysiological data removes noise and artefacts while retaining brain signals. Bandpass filtering eliminates slow drifts below 0.1 Hz from skin potentials and high frequencies above 100 Hz containing muscle noise. Downsampling then reduces the data rate after lowpass filtering. This reduces computation time and the number of features when applying machine learning models to the timeseries. Notch filtering targets removal of 50/60 Hz power line noise and harmonics that can obscure lower amplitude brain signals (Widmann et al., 2015). Narrow stopbands centered on the noise frequencies are applied, e.g. 59-61 Hz to remove 60 Hz.

Noisy channels and segments are identified and repaired or discarded. Bad channel detection utilises statistical thresholds (e.g., too much or too little variance) to find excessively noisy channels for interpolation from surrounding good channels or removal. In EEG a potential cause for bad channels can be high impedance or bad contact with the scalp. Eye blinks, muscle activity, and motion yield large artefacts detected via amplitude, gradient, variance or visual inspection (Fatourechi et al., 2007). Thresholds should retain valid data while excluding only clear artefacts.

Independent component analysis (ICA) decomposes the sensor data X into a set of maximally independent components S, assuming a linear relationship between the components and the observed signal X (Jung et al., 2000). For further details see Section 2.3.1. Components corresponding to artefacts like eye blinks or heart beats can be identified from their spatial, temporal, and spectral signatures and removed before reconstructing the data. This cleans artefacts while preserving brain signals. Specifically, components matching eye blinks have corresponding spatial maps with activity focused in frontal lateral channels (Figure 2.3), while heartbeat components exhibit a characteristic repeating temporal profile around 1Hz (Figure 2.4).

While the preprocessing steps mentioned so far are sufficient for this thesis, often an additional step is mapping sensor-space data to brain sources. Volume conductor modelling creates a head model specifying conductivities of the brain, skull, and surrounding layers for accurate EEG/MEG source localisation (Vorwerk et al., 2014). Realistic models built from subject MRIs can optimise localisation accuracy and spatial interpretation. Source reconstruction enables better alignment with underlying brain geometry and is an active research area (see Timms (2022) for an in-depth review).

In summary, these preprocessing steps clean electrophysiological data, remove artefacts, and prepare multichannel timeseries for further analysis. Careful preprocessing improves signal quality and enhances the fidelity of subsequent analytic



(a) Topographic map of eye blink component

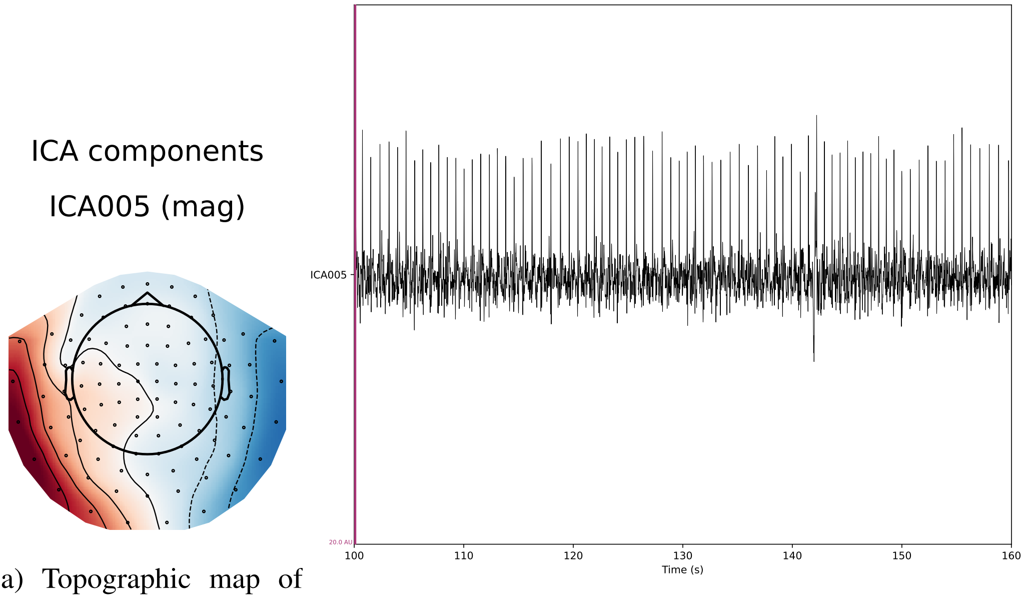
(b) Timeseries of eye blink component

Figure 2.3: An independent component matching eye blink signatures from the ICA decomposition of a MEG recording. Eyeblinks materialise in frontal lateral channels (left) as short large amplitude deviations (right).

approaches. However, most of these steps can only be applied offline, and thus have limited use in BCI systems. A visual comparison of the MEG timeseries before and after applying the aforementioned preprocessing steps is presented in Figure 2.5.

2.2.3 Analysis methods

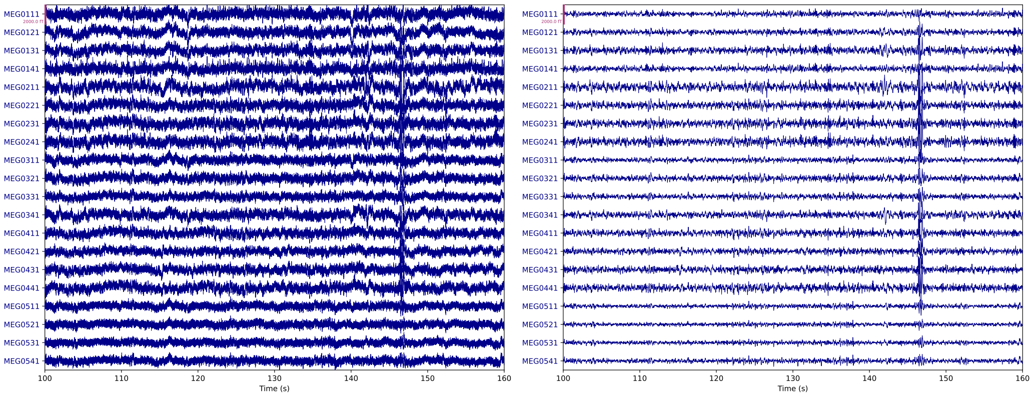
Analyzing electrophysiological data presents unique challenges due to the high-dimensional, multi-channel time series nature of EEG and MEG recordings. In MEG and EEG, a clear 10 Hz oscillation appears that can be observed in real-time recordings when a subject closes their eyes, and gross sleep phases can also be identified in the signal by eye (Schulz, 2008). In more general cases the neuronal signals are stochastic and unintelligible to human eyes. This section reviews



(a) Topographic map of heartbeat component

(b) Timeseries of heartbeat component

Figure 2.4: An independent component matching heartbeat signatures from the ICA decomposition of a MEG recording. Heartbeats show lateral spatial activity (left) and have a consistently repeating high-amplitude pulse-like timeseries. (right)



(a) raw MEG timeseries

(b) preprocessed MEG timeseries

Figure 2.5: Comparing 20 MEG channels before (left) and after (right) running a typical preprocessing pipeline. The data is less noisy, and low and high-frequency activity has been removed. Some artefacts may remain.

common methods to interrogate M/EEG data by examining spatial, temporal, or spectral properties. These techniques provide an initial foothold when confronted

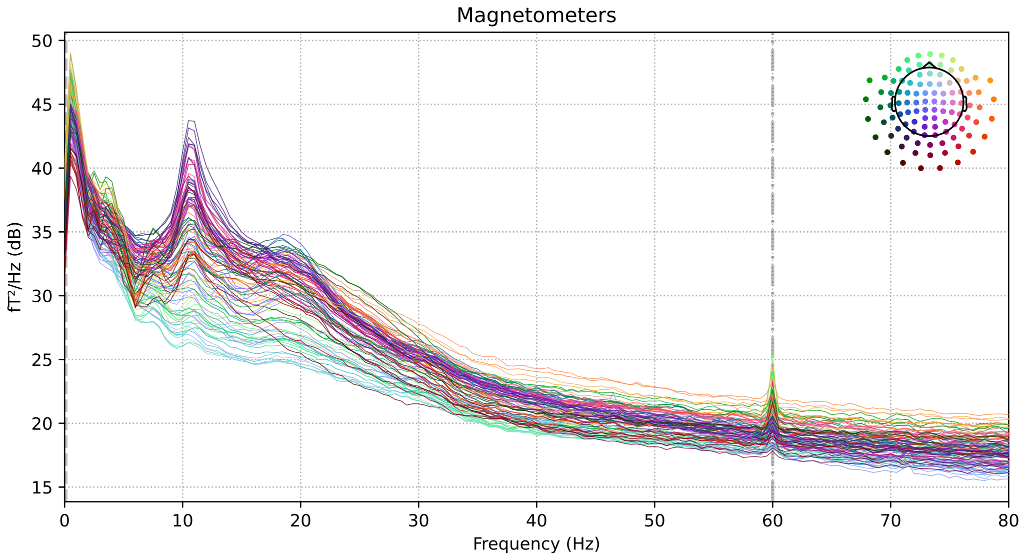


Figure 2.6: PSD of a typical MEG recording. Each line represents a separate channel. The $1/f$ shape is apparent, as well as a prominent peak around 10 Hz, and a sharp peak at 60 Hz (power line noise).

with new electrophysiological recordings.

Power spectral analysis decomposes the signal into constituent frequencies using Fourier-based methods (Cohen, 2014). The power spectral density (PSD) quantifies power within frequency bands of interest. Specifically, the power P at frequency f is computed from the Fourier transform $X(f)$ as:

$$P(f) = |X(f)|^2 \tag{2.1}$$

Band power in canonical delta, theta, alpha, beta, and gamma ranges can be compared across experimental conditions (Buzsaki, 2006). Typical M/EEG spectra follow a $1/f$ distribution with peaks at ~10 and ~20 Hz, reflecting dominant alpha and beta rhythms in the awake brain (Nunez, 2000). A typical MEG power spectra is shown in Figure 2.6.

Time-frequency analysis such as wavelet and Hilbert transforms reveal spectral dynamics over time (Link et al., 2002). This elucidates task-related changes in oscillatory activity, like suppression or enhancement of alpha during visual processing (Klimesch et al., 2007). Topographic mapping of the sensor-level PSD highlights the spatial signature of different rhythms. For instance, alpha power localised to visual cortex decreases during visual tasks (Brüers and VanRullen, 2018). Intrinsic brain networks, like the default mode, also exhibit characteristic spatial spectral patterns (Mantini et al., 2007).

For stimulus-driven experiments, evoked responses are obtained by averaging short data segments time-locked to each event (Luck, 2014). Assuming similar brain responses across trials, this averaging cancels non-stimulus-dependent ongoing brain activity and noise while retaining time-locked signals. Studying peak latencies and amplitudes of visual, auditory, or cognitive components (e.g. P100, N200, P300) provides insights into perceptual and cognitive processes.

Topographic mapping visualises the spatial distribution of brain activity on the scalp (Tzovara et al., 2012). Combined with evoked analysis, this reveals spatiotemporal dynamics (Figure 2.7). For better spatial localisation, inverse solutions like minimum norm estimation and beamforming combine sensor data with head models (Baillet et al., 2001). This allows sensor data to be linked to the brain regions from which signals originate.

To summarise, non-invasive brain activity can be investigated in terms of temporal, spatial and spectral signatures. We can observe each aspect in isolation, e.g. the average PSD across all channels at timepoints, the average evoked response across all channels, or the mean topographic map across all timepoints. Alternatively, the data can be decomposed into pairs such as temporal-spatial, temporal-spectral, and spatial-spectral, or even across all three dimensions. This latter view would

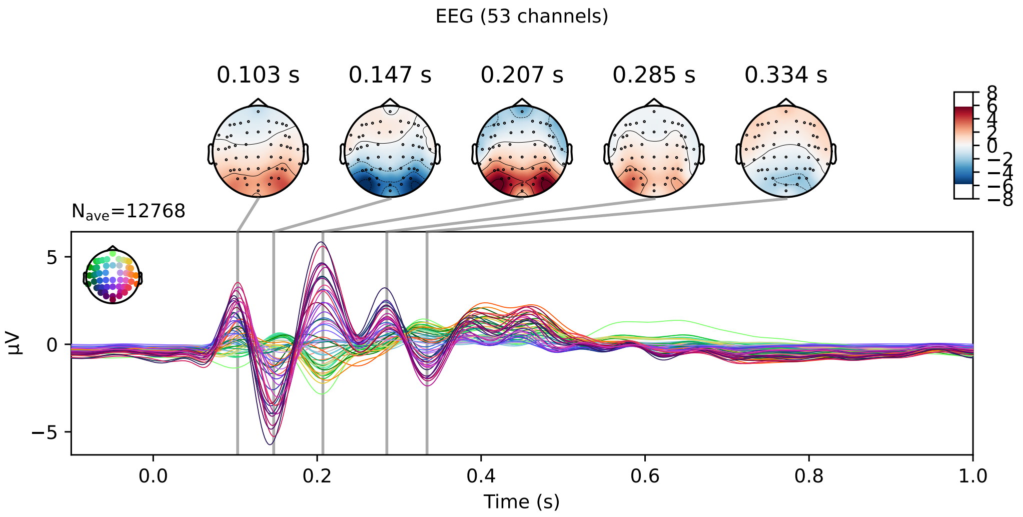


Figure 2.7: Spatiotemporal evoked activity from a visual EEG dataset. The evoked response can be observed as soon as 100 ms after stimulus presentation, followed by several peaks in the timeseries. Topographic maps show that the evoked response appears in channels over the visual area of the brain.

correspond to plotting the wavelet transform as a function of time and space, a topographic PSD map evolving in time. These analyses and views allow various neuroscientific investigations, in both resting-state and task-related brain activity.

Functional connectivity (FC) analyses statistical relationships between the activity in different brain regions, such as coherence (Sakkalis, 2011), and is therefore most straightforwardly carried out in source space. A simpler metric is channel covariance, which reveals spatial relationships in the data. Examining time-varying covariance provides insights into ongoing and stimulus-driven whole-brain dynamics (Vidaurre et al., 2018c; Gohil et al., 2022). Discussed next, modelling covariance structure using generative models provides useful ways to understand M/EEG data.

In conclusion, electrophysiology provides millisecond insights into human brain function. Typical preprocessing removes noise and artefacts enabling further analysis like spectral decomposition, source localisation, functional connectivity, and evoked response characterisation for understanding brain dynamics and evaluating data quality.

2.3 Unsupervised modelling

EEG and MEG provide rich temporal information about brain dynamics. However, these signals represent an aggregate measure of the activity from many neurons and are challenging to interpret. Unsupervised machine learning techniques, such as Principal Component Analysis (PCA), Independent Component Analysis (ICA), and autoregressive (AR) modelling, have proven useful for extracting meaningful representations of neural dynamics from M/EEG (Makeig et al., 1999).

The main challenge these techniques address is the reduction of the high-dimensional (channels × time points) and noisy raw data into a lower-dimensional, human-interpretable latent space. This section and the next will focus on common machine learning techniques used for analysing electrophysiological data. The last section of this chapter will then describe how these models can be leveraged to understand brain activity in more complex ways than the basic signal processing and statistical methods discussed in Section 2.2.3.

2.3.1 PCA and ICA

As mentioned in Section 2.2.2, ICA can be used to remove non-brain artefacts from M/EEG data. To better understand ICA, it is useful to first introduce Principal Component Analysis (PCA). PCA and ICA are commonly used for dimensionality reduction and source separation of M/EEG recordings (Hyvarinen and Oja, 2001; Delorme and Makeig, 2004).

Let us assume we have M/EEG data in the form of a matrix $\mathbf{X} \in \mathbb{R}^{C \times T}$, with C channels and T time points. PCA finds a set of orthogonal basis vectors that capture directions of maximum variance in the data. Let $\mathbf{U} = [\mathbf{u}_1, ..., \mathbf{u}_D] \in \mathbb{R}^{C \times D}$ be the PCA basis vectors, with $D \leq C$. The PCA decomposition is given by:

$$\mathbf{X} = \mathbf{UDV} \tag{2.2}$$

where the rows of $\mathbf{V} \in \mathbb{R}^{D \times T}$ give the PCA component time courses and $\mathbf{D} \in \mathbb{R}^{D \times D}$ is a diagonal matrix containing the eigenvalues. PCA provides a compressed representation of the data by retaining only the top $K < D$ components that explain the most variance. The choice of K is a trade-off between capturing as much variability in the data as possible while maximally reducing the feature space dimension. Using the matrix $\mathbf{U}$ we can map the decomposed data back to the original channel space. Metrics for evaluating a PCA decomposition include reconstruction error and percentage of variance explained in the original channel space data.

Unlike PCA, ICA aims to find statistically independent latent sources $\mathbf{S} \in \mathbb{R}^{N \times T}$ underlying the observed data:

$$\mathbf{X} = \mathbf{AS} \tag{2.3}$$

where $\mathbf{A} \in \mathbb{R}^{C \times N}$ is the mixing matrix. Common ICA algorithms include infomax (Bell and Sejnowski, 1995) and FastICA (Hyv"arinen, 1999).

ICA can better isolate neural and artefact components compared to PCA. The estimated ICA sources do not always have a clear mapping to underlying cortical generators (Makeig et al., 1999). Source localisation, which maps sensor data to cortical sources, requires more sophisticated techniques than PCA or ICA. Thus, these methods all provide complementary information. Due to its data-driven nature, PCA is often used for feature reduction before training machine learning models like Hidden Markov Models or linear classifiers for decoding tasks. By retaining key components explaining most variance, PCA helps mitigate the *curse of dimensionality* when working with limited electrophysiological data.

Typically, PCA and ICA are used to reduce spatial dimensionality while retaining the full temporal resolution. However, we may also want to reduce the temporal (potentially together with spatial) dimensionality, and have a coarser representation of the temporal dynamics. Let us explore such models in the next section.

2.3.2 Hidden Markov Models

To achieve temporal dimensionality reduction, a simple solution is downsampling. However, this can remove useful high-frequency signal. Another option for temporal dimensionality reduction is to classify time points into a limited set of classes, or states. The simplest way to achieve this is by using sliding windows. First, a window of length T (usually 100-200 ms) is slid along the multivariate time series data with a step size of 1 time step. Within each window, statistics like the mean, power spectra, or local (in time) covariance can be derived. Finally, K-means clustering (Hartigan and Wong, 1979) can then be applied on the window statistics (e.g. covariance) to arrive at a discrete set of high-level states governing the dynamics.

States are useful for temporal reduction since empirically their average lifetime is around 100-200 ms (Baker et al., 2014). Using a small set of states (e.g. 10-20) to explain long recordings acts as a bottleneck, summarising the salient recurring patterns. Studies show state statistics like covariance, power spectral density, and

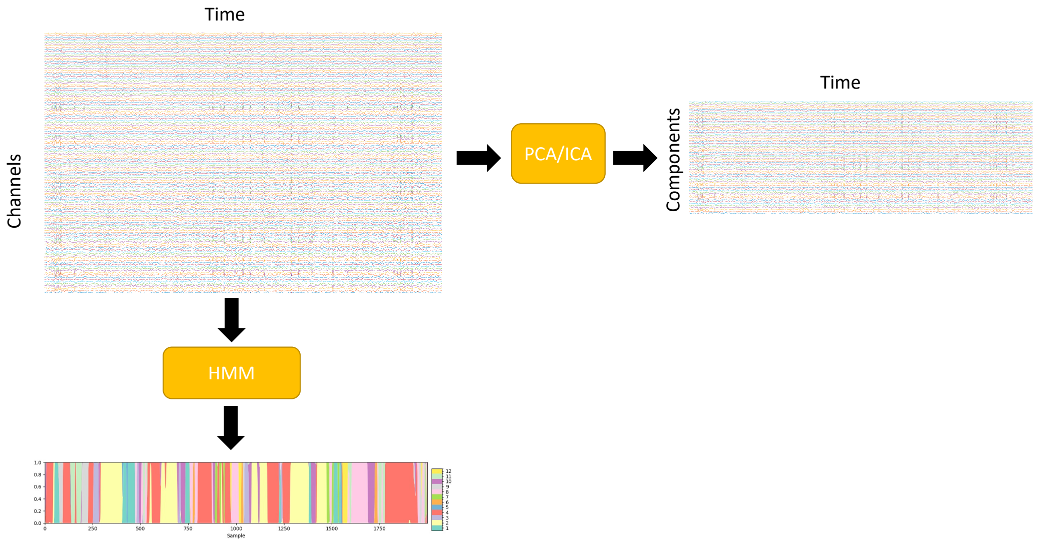


Figure 2.8: The original high-dimensional (channels × time) M/EEG data can be transformed with PCA/ICA to reduce spatial dimensionality. The HMM can be used to infer a set of states governing brain dynamics. The state time course provides both a spatial and temporal dimensionality reduction.

connectivity correspond to identifiable networks and activation patterns during both rest and tasks (Vidaurre et al., 2018a).

However, using sliding windows to find states has limitations. An alternative is to directly learn the states and model state switching dynamics in a data-driven manner. Hidden Markov models (HMMs) are well-suited for modelling sequential data and discovering recurring patterns (Rabiner, 1989a). A conceptual visualisation of HMM, and PCA/ICA for temporal and spatial reduction is given in Figure 2.8.

In an HMM, the observed multivariate time series $\mathbf{x}_1, ..., \mathbf{x}_T$ is assumed to be generated by an underlying sequence of hidden states $z_1, ..., z_T$ with Markov dynamics:

$$p(z_t|z_{t-1},...,z_1) = p(z_t|z_{t-1}) \tag{2.4}$$

$$p(\mathbf{x}_t|z_t,\mathbf{x}_{t-1},...,\mathbf{x}_1) = p(\mathbf{x}_t|z_t) \tag{2.5}$$

The HMM is parameterized by initial state distribution π, transition matrix $\mathbf{A}$, and observation model parameters $\theta = \mathbf{B}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$:

$$\pi_i = p(z_1 = i) \tag{2.6}$$

$$\mathbf{A}_{ij} = p(z_t = j|z_{t-1} = i) \tag{2.7}$$

$$\mathbf{B}_i = p(\mathbf{x}_t|z_t = i) \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{2.8}$$

The transition matrix gives state switching probabilities. The observation model is usually a Gaussian with learned mean and covariance per state. Importantly, the spatial dimension is retained as $\boldsymbol{\Sigma}_i \in \mathbb{R}^{C\times C}$, where C is number of channels. Thus, the HMM provides a high-level state description while able to reproduce the data through the observation model (Figure 2.9). To be clear the state time course is both a spatial and temporal reduction of the original high-dimensional data. HMM parameters can be estimated from the data using the Baum-Welch algorithm (Baum et al., 1970). The Viterbi path gives the optimal hidden state sequence (Forney, 1973). Note that the HMM is a linear model, and the Gaussian observation model can be quite limiting. More realistic observation models like mixtures of Gaussians have been proposed in the literature (Bilmes et al., 1998), but these also usually complicate the inference process.

To aid interpretation, state covariances $\boldsymbol{\Sigma}_i$ can be visualised to characterise re-

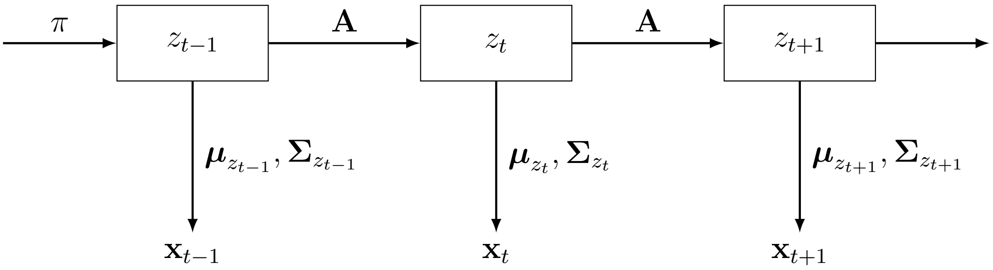

π
z_{t-1}
$\mathbf{A}$
z_t
$\mathbf{A}$
z_{t+1}
$\boldsymbol{\mu}_{z_{t-1}}, \boldsymbol{\Sigma}_{z_{t-1}}$
$\boldsymbol{\mu}_{z_t}, \boldsymbol{\Sigma}_{z_t}$
$\boldsymbol{\mu}_{z_{t+1}}, \boldsymbol{\Sigma}_{z_{t+1}}$
$\mathbf{x}_{t-1}$
$\mathbf{x}_t$
$\mathbf{x}_{t+1}$

Figure 2.9: A graphical illustration of the HMM. See text for parameters.

curring brain patterns (Vidaurre et al., 2018a). Section 2.5 provides more details on interpreting HMMs. More powerful deep generative models can also replace HMMs, as will be discussed in Chapter 5.

2.3.3 Linear autoregressive models

As we have seen the HMM is an unsupervised generative sequential model well-suited for M/EEG data. Another method of self-supervision is sequential (time-series) forecasting, where the aim is to model the conditional probability of the future given the past. Autoregressive (AR) models are a powerful class of self-supervised generative models well-suited for forecasting. AR models aim to directly model the conditional probability distribution of future timesteps given the past, i.e. $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_{t-2}, ..., \mathbf{x}_{t-P})$ where $\mathbf{x}_t \in \mathbb{R}^C$ is a multivariate time series with C channels at time t. This formulation allows AR models to learn complex temporal dynamics from the data in a fully unsupervised manner.

The AR modelling approach shares similarities with Hidden Markov Models (HMMs), with a couple key distinctions. First, AR models do not enforce a low-dimensional latent state space, instead operating directly on the observed data. Second, AR models do not in general make the Markov assumption, allowing dependence on multiple past timesteps rather than just the previous state (Equation 2.4).

More formally, the log-likelihood $\log p(\mathbf{x}_t|\mathbf{x}_{t-1}, ..., \mathbf{x}_{t-p}; \theta)$ is maximised with respect to model parameters θ. Using Gaussian noise assumptions, this is equivalent to minimising the squared error between predictions and targets:

$$\underset{\theta}{\operatorname{argmin}}(\mathbf{x}_t - f(\mathbf{x}_{t-1}, \mathbf{x}_{t-2}..., \mathbf{x}_{t-P}; \theta))^2 \tag{2.9}$$

The simplest AR model specification uses a linear function for f:

$$\mathbf{x}_t = \sum_{i=1}^{P} \mathbf{A}_i \mathbf{x}_{t-i} + \boldsymbol{\epsilon}_t \tag{2.10}$$

where $\mathbf{A}_i \in \mathbb{R}^{C \times C}$ are the autoregressive coefficients at time lag i and $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \boldsymbol{\Sigma})$ is Gaussian noise. P controls the model order, which determines the length of temporal memory or receptive field. C is the number of input channels. To be precise, this is known as a multivariate AR (MAR) model, as it captures cross-channel interactions (Schlögl and Supp, 2006). A simpler (univariate) approach is to avoid modelling the channel interactions in $\mathbf{A}_i$ by letting it be a scalar for each channel ($\mathbf{A}_i \in \mathbb{R}^C$). This decouples the conditional dependence between channels, as the conditional probability $p(x_{t,c}|x_{t-1,c}, ..., x_{t-P,c})$ is modelled with a separate univariate model for each channel c. AR models can be fit via ordinary least squares.

MAR models are able to capture linear temporal autocorrelations as well as cross-channel relationships in M/EEG data. They can be interpreted from a signal processing lens as infinite impulse response filters applied to the input (Takalo et al., 2005). This enables analysing model dynamics in the frequency domain using tools from spectral analysis.

A key advantage of MAR models is the ability to generate new data recursively for any length by feeding back previous outputs. However, linearity remains a limitation, which motivates exploring nonlinear AR models in Chapter 5. We conclude this section by briefly introducing basic nonlinear models.

2.3.4 Neural network autoregressive models

Neural networks are powerful function approximators that have proven effective for time series modelling and forecasting. They comprise multiple layers of nonlinear transformations with learned parameters that enable extracting hierarchical features from the input data. Fully-connected neural networks with sufficiently wide hidden layers can approximate any continuous function, as established by the universal approximation theorem (Hornik et al., 1989a). However, architectural constraints like weight sharing and recurrence are often incorporated to exploit structure in sequential data. Fully connected networks consist of affine transformations followed by nonlinear functions, such as the sigmoid or ReLU function (Nair and Hinton, 2010), stacked on top of each other. To learn the parameters of such models, the backpropagation algorithm (Rumelhart et al., 1985) is used to differentiate an objective function (e.g. sum of squared errors), and an optimisation algorithm such as stochastic gradient descent is used to minimise this function.

Convolutional neural networks (CNNs) leverage local spatial or temporal correlations through weight sharing across space/time (Fukushima and Miyake, 1982). They comprise convolutional layers that convolve input representations with learned kernels followed by nonlinearities. This equates to extracting finite impulse response (FIR) filters whose coefficients are optimised via gradient descent. The learned kernels act similarly to the autoregressive coefficients in modelling temporal dependencies. Through hierarchical feature extraction, deep CNNs can construct complex spatiotemporal representations (Krizhevsky and Sutskever, 2012; Chambon et al., 2018). The weights of a MAR model are equivalent to a CNN with a single layer without any nonlinearities.

For an M/EEG input $\mathbf{X} \in \mathbb{R}^{C \times T}$, a CNN with L layers produces an output representation $\mathbf{H}^{(L)} \in \mathbb{R}^{M \times T}$:

$$\mathbf{H}^{(l+1)} = [\mathbf{h}_1^{(l+1)}; \mathbf{h}_2^{(l+1)}; ...; \mathbf{h}_{M^{(l+1)}}^{(l+1)}] \tag{2.11}$$

$$\mathbf{h}_m^{(l+1)} = f^{(l)}(\mathbf{b}_m^{(l)} + \sum_{i=1}^{M^{(l)}} \mathbf{w}_{m,i}^{(l)} * \mathbf{h}_i^{(l)}) \tag{2.12}$$

$$\mathbf{H}^{(0)} = \mathbf{X} \tag{2.13}$$

where $[;]$ denotes concatenation across the channel dimension, $*$ is convolution, $\mathbf{W}_m^{(l)}$, $\mathbf{b}_m^{(l)}$ are the learned weights/biases corresponding to output channel m in layer l. Each layer can have a different nonlinearity $f^{(l)}$, such as ReLU. $M^{(l)}$ and $M^{(l+1)}$ are the number of input/output channels in layer l.

By concatenating across channels we can concisely denote the full bias and weight tensors of layer l by $\mathbf{B}^{(l)} \in \mathbb{R}^{M^{(l+1)} \times M^{(l)}}$ and $\mathbf{W}^{(l)} \in \mathbb{R}^{M^{(l+1)} \times M^{(l)} \times K^{(l)}}$ where $K^{(l)}$ is the kernel size in layer l. For forecasting, the CNN can be trained to minimise error between model outputs $\mathbf{H}^{(L)}$ and future timesteps. Thus, the target labels are the inputs shifted by 1 timestep.

Stacking layers enables hierarchical feature learning. Temporal downsampling can be achieved via strided convolution or pooling layers like max pooling. This simply takes a time window as input and outputs the maximum value from this window, sliding over the timeseries. Architectural innovations like dilated convolutions (van den Oord et al., 2016) also enable expanding the receptive field to capture longer-range dependencies. CNNs are widely used for end-to-end M/EEG modelling (Schirrmeister et al., 2017b; Chambon et al., 2018).

Recurrent neural networks (RNNs) are the nonlinear counterpart to HMMs. They comprise recurrent connections that enable maintaining a state over time and modelling long-term temporal contexts (Rumelhart et al., 1986). An RNN takes inputs $\mathbf{X} \in \mathbb{R}^{C \times T}$ and outputs a hidden state $\mathbf{h}_t$ and optional outputs $\mathbf{y}_t$ per timestep:

$$\mathbf{h}_t = f(\mathbf{U}\mathbf{x}_t + \mathbf{W}\mathbf{h}_{t-1} + \mathbf{b}) \tag{2.14}$$

$$\mathbf{y}_t = g(\mathbf{S}\mathbf{x}_t + \mathbf{V}\mathbf{h}_t + \mathbf{c}) \tag{2.15}$$

Here, f and g are nonlinearities like Gated Recurrent Unit (GRU) or Long Short-Term Memory (LSTM) cells (Hochreiter and Schmidhuber, 1997). $\mathbf{U}$, $\mathbf{W}$, $\mathbf{S}$, $\mathbf{V}$ are learned projections and $\mathbf{b}$, $\mathbf{c}$ are biases. The fixed weights at each timestep coupled with recurrent state enables modelling complex dynamics. The matrix $\mathbf{S}$ corresponds to skip connections as the hidden state is bypassed in the information flow from inputs to outputs. In standard RNN formulations $\mathbf{S}$ is not used. By removing the nonlinearities and bias terms one can recognise a linear state-space system in these equations. Indeed RNN dynamics may be studied in terms of a dynamical state-space system when certain assumptions are met.

RNNs can model multivariate AR dynamics by predicting future values of the timeseries. We can enforce this through the objective function (e.g. mean-squared error) applied to output $\mathbf{y}_t$:

$$\text{MSE}(\mathbf{y}_t, \mathbf{x}_{t+1}) = \frac{1}{C}\sum_{i=1}^{C}(y_{t,i} - x_{t+1,i})^2 \tag{2.16}$$

While the above description is accurate for a 1-layer RNN, these operations can

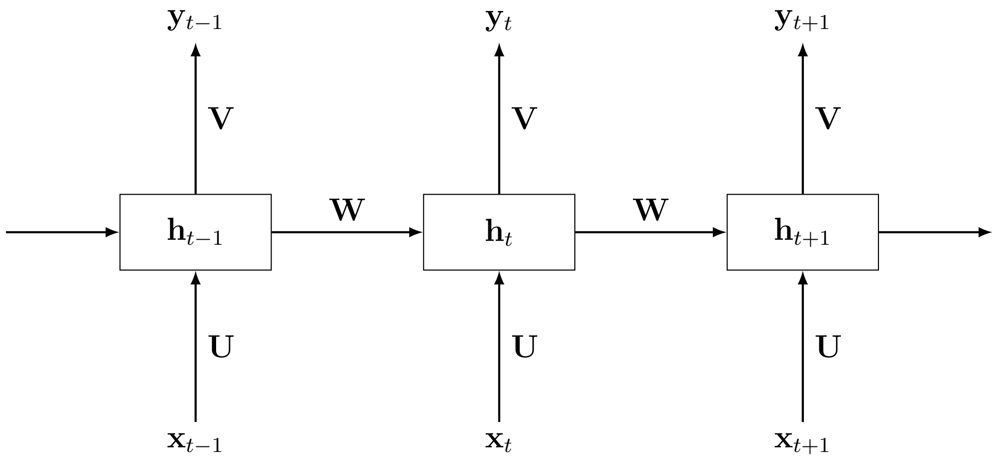


Figure 2.10: Graphical illustration of the recurrence in an RNN layer. Nonlinearities, biases, and the projection $\mathbf{S}$ are omitted.

be stacked on top of each other by feeding the output representations $\mathbf{y}_t$ as input to the next layer. Multiple layers allow deeper feature extraction/representation capabilities. CNNs and RNNs may look somewhat similar, however a CNN layer is state-less and exploits weight-sharing across time, whereas an RNN layer applies the same operation at each timestep while carrying a continually updating hidden state (Figure 2.10). In practice several extensions have been proposed to both types of architectures, even combining them into a single model (Bashivan et al., 2015a). These and other, more recent modelling approaches (such as Transformers) will be explored in Chapter 5.

2.4 Encoding and decoding

In the previous section, we have seen how unsupervised modelling of M/EEG data can uncover intrinsic brain dynamics. While these models elucidate spontaneous neural processes, they do not account for external stimuli and behaviour. By incorporating such external events, we can study the relationship between brain dynamics and the outside world. This is also necessary for applying our models in

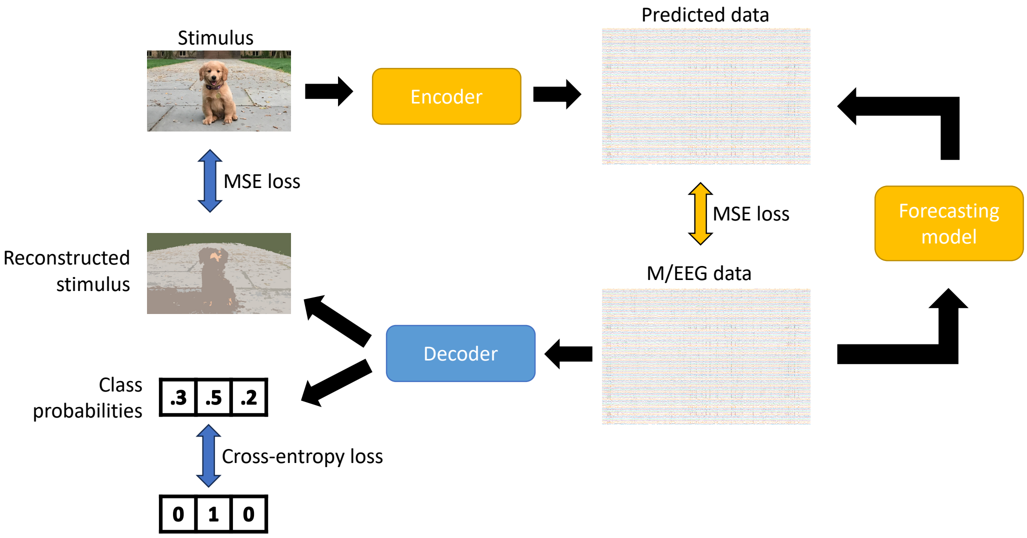


Figure 2.11: Conceptual comparison of forecasting, encoding, and decoding. A typical forecasting model is fed some brain data to predict future timesteps and trained through the MSE loss. A typical encoder is similar to the forecasting model except that it is fed the stimulus. A typical decoder is similar to the forecasting model except that it has to either reconstruct the stimulus (trained with MSE loss), or predict the stimulus class, trained with the cross-entropy loss. Note that each modelling approach may involve standard feature extraction steps, and thus map features to outputs, instead of raw stimuli or brain data.

brain-computer interfaces (BCIs).

If we conceptualise the brain as a dynamic system that receives inputs (e.g., visual, auditory) from the environment and generates outputs (e.g., movement, emotion, speech) there are several approaches for investigating input-output relationships. In this thesis, we specifically focus on links between external inputs and resultant brain dynamics. This relationship can be studied bidirectionally, termed encoding and decoding. For a high-level visualization comparing forecasting, encoding, and decoding see Figure 2.11.

Encoding refers to predictive modelling of brain activity evoked by stimuli. Such models elucidate how external inputs are processed and represented in the brain

(Naselaris et al., 2011). Decoding models predict, classify, or reconstruct stimuli based on elicited brain activity, providing insights into neural representations and importantly enabling BCI applications. Key considerations in encoding models include accurately mapping temporally evolving, stimulus-evoked responses and their spatial propagation across regions. Decoders can also infer these representational dynamics. For BCIs, harnessing knowledge of evoked spatiotemporal dynamics can inform input constraints.

An important aspect of both encoding and decoding is the level of generalisation. As discussed, M/EEG variabilities necessitate particular forms of generalisation. External stimuli pose an additional dimension. The simplest case is predicting responses to identical stimuli in new trials. Increased difficulty arises in generalising to unseen stimuli from the training distribution (e.g., novel dog images). Further generalisation may predict responses to any within-distribution sample (any image). The generalisation levels can be similarly formulated for decoding.

For a thorough introduction to electrophysiological encoding and decoding, Holdgraf et al. (2017) provides an excellent resource to interested readers.

2.4.1 Encoding

Linear encoding characterises relationships between stimuli or task conditions $\mathbf{s}_t \in \mathbb{R}^K$ and resultant M/EEG response $\mathbf{x}_t \in \mathbb{R}^C$ at timepoint t:

$$\mathbf{x}_t = \mathbf{W}_t \mathbf{s}_t + \boldsymbol{\epsilon}_t \tag{2.17}$$

where $\mathbf{W}_t$ is a weight matrix and $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \boldsymbol{\Sigma})$ is zero-mean Gaussian noise with covariance $\boldsymbol{\Sigma}$. Noise can be static or time-varying. K is stimulus feature dimensionality. Weights can also be static or dynamic over time. For static weights, the overall M/EEG response $\mathbf{X} \in \mathbb{R}^{C \times T}$ can be predicted by:

$$\mathbf{X} = \mathbf{W}\mathbf{S} + \boldsymbol{\epsilon} \tag{2.18}$$

where $\mathbf{S} \in \mathbb{R}^{K \times T}$ contains the stimulus features at each timepoint. This models the conditional distribution $p(\mathbf{X}|\mathbf{S})$. Given N labelled trials $(\mathbf{S}_n, \mathbf{X}_n)$, linear encoding is fit by least squares if the error is assumed to have a symmetric distribution:

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmin}} \sum_{n=1}^{N} |\mathbf{X}_n - \mathbf{W}\mathbf{S}_n|_2^2 \tag{2.19}$$

minimising squared error between predicted and actual responses across trials.

Once fit, encoding models can be analysed to study neural representations. Weights $\mathbf{w}_c = \mathbf{W}[c, :]$ for channel c indicate its selectivity for stimulus features in $\mathbf{S}$. Spatial mapping of weight magnitudes localises brain areas selective for different features. Comparing weights across channels and stimuli reveals distributed sensory representations.

Linear encoding has limited flexibility in capturing complex relationships between stimuli and responses. Simplicity also restricts insights into nonlinear neural computations/dynamics. Nonlinear approaches, such as neural networks, provide greater modelling flexibility for encoding. A multi-layer fully-connected neural network can be formulated as:

$$\mathbf{h}^{(1)} = f^{(1)}(\mathbf{W}^{(1)}\mathbf{s} + \mathbf{b}^{(1)}) \tag{2.20}$$

$$\mathbf{h}^{(l+1)} = f^{(l)}(\mathbf{W}^{(l+1)}\mathbf{h}^{(l)} + \mathbf{b}^{(l+1)}) \tag{2.21}$$

$$\hat{\mathbf{x}} = g(\mathbf{W}^{(L)}\mathbf{h}^{(L-1)} + \mathbf{b}^{(L)}) \tag{2.22}$$

where $f(\cdot)$ and $g(\cdot)$ are nonlinear activations, $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are the learned weights/biases of layer l, and $\hat{\mathbf{x}}$ is the predicted response. This can be applied per-timestep with $\mathbf{s} = \mathbf{s}_t$, $\hat{\mathbf{x}} = \hat{\mathbf{x}}_t$, and static or time-dependent $\mathbf{W}$. Alternatively, the model can be applied to the whole stimulus $\mathbf{S} \in \mathbb{R}^{K \times T}$ in which case $\mathbf{s} \in \mathbb{R}^{KT}$ is the flattened vector form of $\mathbf{S}$, and similarly for $\hat{\mathbf{x}}$. In this latter case $\mathbf{W}$ is by default time-dependent, since the time dimension is present in the input.

Stacking layers enables learning hierarchical nonlinear feature dynamics, capturing complex stimuli-response relationships. Note that T can be freely chosen and indeed it can range from 1 timestep up to a larger stimulus window, e.g., several seconds. When setting it to a small number of timesteps (1-20) and prescribing $\mathbf{W}$ to be time-dependent we call this approach sliding window encoding. Training these models is achieved similarly to forecasting models, i.e., using the MSE loss.

The fully-connected network can be replaced by both CNNs or RNNs to provide constraints and introduce more complexity in feature extraction and modelling. Compared to the equations described in Section 2.3.4 one simply needs to replace $\mathbf{X} \in \mathbb{R}^{C \times T}$ with $\mathbf{S} \in \mathbb{R}^{K \times T}$ to define an encoding model. Often we can exploit the structure in the stimulus with specialized models, such as CNNs for images, or RNNs for audio or language stimuli.

As shown in Section 2.3.4, CNNs and RNNs models are also well suited for forecasting timeseries. As the goal of encoding is also to predict brain data, it comes as no surprise that providing both past timesteps and stimulus features as input to CNNs and RNNs leads to better encoders. This is also called conditioning the forecasting model on stimulus information. This improves encoding/forecasting performance as instead of predicting open-ended brain activity relying solely on past brain activity, the prediction is constrained/conditioned on the respective external stimulus. A straightforward approach would be to concatenate the flattened stimulus vector $\mathbf{s}$ to the timeseries data $\mathbf{x}_t$ at each timepoint. This also allows for modelling variability in the evoked response by taking into account the past state of the brain. Simple, deterministic encoders predict identical responses to repeated stimuli, unlike real variable data. The high-dimensional output space of the encoder is also better constrained by past brain activity.

Chehab et al. (2022) combined CNNs and RNNs to predict MEG data from pre-stimulus activity and word features, finding timing differences in feature importance. Deep encoders enable studying neural representations and computations underlying M/EEG responses to complex stimuli. Learned features can be visualised and analysed to reveal encoding transformations (Kriegeskorte, 2015).

2.4.2 Decoding

Decoding refers to the inverse (non-causal) process of encoding, in which task conditions or stimuli are inferred from recorded brain activity data. In contrast to encoding models which predict brain activity from stimuli, decoding models estimate stimuli or behaviours from brain activity patterns. Decoding analyses have become increasingly prevalent in neuroimaging research (Kay et al., 2008; Guggenmos et al., 2018; Cichy et al., 2016), with applications ranging from brain-computer interfaces (Lotte et al., 2018; Willett et al., 2021a) to basic neuroscience inquiries (Haynes and Rees, 2006; Kay et al., 2008). Decoding approaches can be applied to diverse experimental paradigms and neural measurement modalities, including the decoding of visual stimuli (Cichy et al., 2016), phonemes and words (Mugler et al., 2014; Hultén et al., 2021; Cooney et al., 2019b), imagined speech (Dash et al., 2020a), and motor movements (Willett et al., 2021a; Dash et al., 2020b; Elango et al., 2017).

Stimulus decoding may refer to reconstructing the full sensory stimulus (Anumanchipalli et al., 2019; Shen et al., 2019), estimating stimulus features or categories (Cichy et al., 2016; Kay et al., 2008), or simply classifying among a predefined set of stimuli (Mugler et al., 2014; Hultén et al., 2021). While reconstructing arbitrary novel stimuli poses considerable challenges, classification provides a straightforward decoding objective when experiments utilise fixed stimulus sets. Hence, in this dissertation decoding is carried out primarily through supervised classification.

Conceptually, decoding can be framed as inverting the encoding model from brain activity to stimuli (Equation 2.18):

$$\mathbf{S} = \mathbf{W}\mathbf{X} + \boldsymbol{\epsilon} \tag{2.23}$$

where $\mathbf{S}$ represents the stimulus, $\mathbf{X}$ is measured brain activity, $\mathbf{W}$ is a weight matrix, and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$ is zero-mean Gaussian noise. This expresses the conditional distribution $p(\mathbf{S}|\mathbf{X})$ that enables inferring stimuli from brain patterns. While this model can be trained via regression, classification models are more commonly employed for decoding tasks.

For example, consider an MEG recording $\mathbf{X} \in \mathbb{R}^{C \times T}$ comprising C channels and T timepoints. Let $\mathbf{x}_t \in \mathbb{R}^C$ denote the spatial topography across channels at time t. Given class labels $y \in 1, \ldots, K$, where K is the number of classes, linear discriminant analysis (LDA) models the class conditional densities $p(\mathbf{x}_t|y_k)$ as

Gaussians $\mathcal{N}(\boldsymbol{\mu}_k, \Sigma)$ and applies Bayes' rule to predict labels (Lemm et al., 2011):

$$p(y_k|\mathbf{x}_t) = \frac{p(\mathbf{x}_t|y_k)p(y_k)}{\sum_{l=1}^{K} p(\mathbf{x}_t|y_l)p(y_l)} \tag{2.24}$$

$$\log(p(y_k|\mathbf{x}_t)) = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k\mathbf{x}_t - \frac{1}{2}\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \log(p(y_k)) \tag{2.25}$$

where the class covariance matrix $\Sigma_k = \Sigma$ is assumed equal. LDA provides interpretable discriminative spatial patterns through the model weights $\mathbf{W} = \Sigma^{-1}\boldsymbol{\mu}$. However, LDA's linear decision boundaries limit flexibility for complex data.

For decoding time-varying signals, sliding window LDA trains classifiers on short time segments $\mathbf{x}_{t:t+P}$, where P is the window length. This reveals how decodable information evolves over time (Grootswagers et al., 2020), but overlooks long-range dependencies beneficial for decoding. Similar to encoding, the entire trial $\mathbf{X}$ can be flattened into a vector $\mathbf{x} \in \mathbb{R}^{CT}$ for LDA, though high dimensionality may hinder learning. For all of the following models we will denote the inputs with $\mathbf{x}$, which can either refer to 1 timestep, a sliding window, or the full flattened trial, depending on the problem at hand.

Beyond LDA, logistic regression and support vector machines comprise other widely used linear decoding models (Lotte et al., 2018). Logistic regression models the class conditional logits using an affine projection, and then applies the softmax function to obtain probabilities for prediction:

$$\mathbf{l} = \mathbf{W}\mathbf{x} + \mathbf{b} \tag{2.26}$$

$$p(\mathbf{y}|\mathbf{x}) = \frac{e^{\mathbf{l}}}{\sum_{i=1}^{K} e^{l_i}} \tag{2.27}$$

where $\mathbf{l}$ is the logit vector. The softmax function is the extension of the logistic function to more than 2 classes. Logistic regression is probabilistic like LDA but does not assume Gaussian densities, and it is discriminative instead of generative.

Linear models provide interpretable mappings from features to predictions. However, their simplicity can limit decoding complex spatiotemporal brain patterns. As in encoding, multi-layer fully-connected neural networks extend logistic regression via nonlinear hidden layers. The input $\mathbf{x}$ propagates through fully-connected layers to produce class predictions:

$$\mathbf{h}^{(1)} = f^{(1)}(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) \tag{2.28}$$

$$\mathbf{h}^{(l+1)} = f^{(l)}(\mathbf{W}^{(l+1)}\mathbf{h}^{(l)} + \mathbf{b}^{(l+1)}) \tag{2.29}$$

$$\hat{\mathbf{y}} = \mathrm{softmax}(\mathbf{W}^{(L)}\mathbf{h}^{(L-1)} + \mathbf{b}^{(L)}) \tag{2.30}$$

where $f^{(l)}$ denotes a nonlinearity like ReLU, and $\hat{\mathbf{y}}$ gives predicted class probabilities. Cross-entropy loss trains the network for classification:

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = -\mathbf{y}\log\hat{\mathbf{y}} \tag{2.31}$$

where $\mathbf{y}$ is the true class label distribution. This is also called a one-hot vector since it consists of a 1 at the target class index and zeros everywhere else.

Classification objectives are common for decoding, although regression losses can enable reconstructing richer stimulus representations and provide wider generalisation. However, this comes with considerable challenges as the output dimensionality is much higher than in the case of classification.

Similarly, CNNs and RNNs can be adapted for decoding by changing the targets and loss function compared to forecasting. At a high-level, CNNs insert temporal convolutions before a classifier:

$$\mathbf{H} = \mathrm{CNN}(\mathbf{X}) \tag{2.32}$$

$$\hat{\mathbf{y}} = \mathrm{softmax}(\mathbf{W}\mathrm{flatten}(\mathbf{H}) + \mathbf{b}) \tag{2.33}$$

While RNNs insert temporal recurrences:

$$\mathbf{h}_T = \mathrm{RNN}(\mathbf{X}) \tag{2.34}$$

$$\hat{\mathbf{y}} = \mathrm{softmax}(\mathbf{W}\mathbf{h}_T + \mathbf{b}) \tag{2.35}$$

Early work showed fully-connected and convolutional neural networks can decode motor intentions from EEG better than linear SVMs (Tabar and Halici, 2016). CNNs have proven effective for motor decoding (Schirrmeister et al., 2017b) and robust generalization across subjects (Lawhern et al., 2018). The learned filters provide insight into discriminative patterns. Some variants use time-convolutional layers before recurrent layers to extract local temporal features (Bashivan et al., 2015b). RNNs directly model the temporal hierarchy of neural processes (Kubilius et al., 2019).

It is important to mention that an alternative and more traditional way to improve on linear models (instead of using nonlinear models), is to first apply some nonlinear transformation to the input data, e.g. the wavelet transform, and then train the linear decoder (Higgins et al., 2022b; Hu et al., 2011). In general any encoding or decoding model can operate on a transformed feature space instead of the raw stimuli or brain data.

Next we will provide an introduction to leveraging unsupervised, encoding, and decoding models for understanding the brain.

2.5 Interpretability methods

While the analysis methods presented in Section 2.2.3 focused on utilising signal processing and statistical techniques to characterise brain activity, we can also leverage some of the more advanced modelling approaches described in the previous sections in our quest to understand noninvasive electrophysiology.

2.5.1 HMM statistics

Hidden Markov models (HMMs) have proven to be a powerful tool for modelling sequential M/EEG data and uncovering recurring brain states (Vidaurre et al., 2018b). Once an HMM is fit to M/EEG observations, the estimated model parameters and hidden state sequences can be analysed in various ways to elucidate the spatiotemporal dynamics of brain activity. This section outlines common techniques for interpreting trained HMMs on M/EEG data.

The HMM transition matrix $\mathbf{A} \in \mathbb{R}^{K \times K}$ describes the Markovian dynamics between states, where K is the number of states:

$$A_{ij} = p(z_t = j | z_{t-1} = i) \tag{2.36}$$

The overall transition structure characterises dynamic reconfigurations of large-scale brain networks (Vidaurre et al., 2018b). The spatial covariance patterns $\mathbf{\Sigma}_i$

characterise functional connectivity networks associated with each state (Vidaurre et al., 2018b).

The power spectral density (PSD) of each state reveals its oscillatory profile. Importantly, we only look at PSD when using the HMM method in conjunction with time-delay embeddings (TDE). This method augments the input to the HMM with multiple lagged versions of the time series. The PSD for state i can be estimated from data segments assigned to that state based on the Viterbi path. Different states often exhibit distinct spectral signatures related to underlying cognitive processes. For example, alpha desynchronisation may indicate active sensory processing (Klimesch, 2012). Visualising PSD topographically links specific large-scale networks to spectral features like alpha/beta desynchronisation or theta/gamma synchronisation (Vidaurre et al., 2018b).

The Viterbi path $z_1^*, \ldots, z_T^*$ provides the optimal hidden state sequence explaining the observations. With task data, evoked responses can be computed over the state timecourse. This reveals which states are activated by certain events, and their temporal evolution within the trial.

General statistics can also be calculated from the state timecourses:

- Fractional occupancy: fraction of time spent in each state.
- Mean lifetime: average duration of each state visit.
- Mean interval: average time between consecutive state visits.
- Switching rate: rate of transitions into each state.

Comparing these metrics across states, task conditions, subjects, datasets, or models can reveal insights into differences in brain dynamics. The distribution of the statistics within and across states can also be quantified to characterise variability. Thus HMMs provide a rich set of analysis tools beyond standard spectral analysis, functional connectivity, and evoked responses estimated from brain data.

2.5.2 AR generation

Autoregressive (AR) models provide a powerful framework for modelling the dynamics of multivariate M/EEG time series. Once an AR model is fit to M/EEG data, the estimated model parameters and generated data samples can be analysed in various ways to elucidate both local and global spatiotemporal characteristics of brain activity. This section outlines techniques for interpreting trained AR models on M/EEG data.

A multivariate autoregressive (MAR) model characterises how a multivariate time series $\mathbf{x}_t \in \mathbb{R}^C$ evolves linearly over time. The AR coefficients $\mathbf{A}_p \in \mathbb{R}^{C \times C}$ directly encode the autoregressive dynamics, with elements $A_p[i, j]$ relating channel j at lag p to channel i at the current time step. Non-zero off-diagonal elements in $\mathbf{A}_p$ indicate cross-channel interactions, while a diagonal structure reflects independence (Chiarion et al., 2023). The overall cross-channel coefficient structure provides insight into functional brain networks.

The power spectral density (PSD) describes the distribution of power across frequencies. For a linear multivariate AR process, the PSD matrix is (Schlögl and Supp, 2006):

$$\mathbf{S}(f) = \frac{1}{2\pi}\mathbf{\Sigma}|\mathbf{I} - \sum_{p=1}^{P} \mathbf{A}_p e^{-i2\pi f p}|^{-2} \tag{2.37}$$

where $\mathbf{I}$ is the identity matrix, and $\mathbf{\Sigma}$ is the noise covariance. Diagonal PSD elements $\mathbf{S}_{ii}(f)$ give the spectrum for each channel i. The PSD decompositions provide insight into the oscillatory characteristics captured by the model, including peak frequencies, spectral shape (e.g. $1/f$), and spatial topographies of different rhythms.

A key advantage of AR models is their ability to generate new data points by feeding back their own predictions:

$$\hat{\mathbf{x}}_t = \sum_{p=1}^{P} \mathbf{A}_p \hat{\mathbf{x}}_{t-p} + \boldsymbol{\epsilon}_t \tag{2.38}$$

$$\hat{\mathbf{x}}_{t-p} = \text{past generated data} \tag{2.39}$$

This allows synthesising multidimensional time series with similar dynamics as the empirical data. Comparing real and simulated data based on spectral, spatial, and temporal properties helps validate that the model accurately captures the characteristics of interest. Note that recursive generation is straightforward for any of the neural network models described in Section 2.3.4, not only linear models.

Long generative runs enable assessing model stability through metrics like divergence from the empirical covariance matrix (Schlögl and Supp, 2006). If outputs diverge over time, the model may be underconstrained. Fitting to more data until predictions remain accurate ensures proper characterisation of the system dynamics.

Key empirical measures for comparing real and generated data include:

- **Spectral content**: The PSD reveals whether the model accurately captures oscillations in specific frequency bands.
- **Covariance**: The overall covariance matrix measures global correlation structure.

- **Evoked responses**: Generating data related to stimuli, e.g. by initialising the generation with the start of the evoked response tests modelled event-related dynamics.
- **HMM statistics**: Comparing the statistics of HMMs (discussed in Section 2.5.1) trained on real and generated data reveals how well the AR model captures higher-level metrics of brain dynamics.

In summary, AR modelling provides a concise framework for capturing the spatiotemporal dynamics in multivariate M/EEG recordings. Analysing model coefficients, spectral characteristics, and generated samples gives insights into oscillatory content and event-related attributes of brain activity. Comparisons against real data validate how accurately the model reproduces empirical dynamics, supporting further analysis of both spontaneous and task-related neural processes.

2.5.3 Multivariate pattern analysis

Multivariate pattern analysis (MVPA) refers to a set of techniques that apply machine learning algorithms to neural data (e.g. M/EEG, fMRI) to uncover distributed neural representations and decode mental states (Haxby et al., 2001; Haynes and Rees, 2006). In contrast to univariate methods that examine individual sensors or voxels, MVPA examines multivariate patterns of activity across multiple channels or voxels to discriminate between experimental conditions. This provides insights into information encoding in the brain that are not accessible with univariate approaches.

Encoding models can be analysed to study model weights $\mathbf{W}$ reflecting channel tuning, and feature importance for model prediction. This reveals how stimuli are transformed and represented in brain activity.

Once fit, both encoding and decoding models can be analysed to reveal discriminative spatial, temporal, and spectral patterns. This provides insights into neural coding underlying perceptions and behaviours. Similarly to evoked analyses we are interested in the temporal, spatial, and spectral signatures of evoked activity. Leveraging multivariate decoding models for this is the central aim of MVPA.

Sliding-window analysis divides continuous data into short overlapping segments and extracts features from each window separately to characterise temporal dynamics (Higgins et al., 2022b). Windows may be from 10 ms up to 200 ms long and slide in small steps (10-20 ms). Compared with evoked response analysis, machine learning decoding models trained on time windows provide a complementary view of the temporal activity, by plotting cross-validated accuracy (across trials) for each time window. Such a sliding window decoding approach may reveal the temporal evolution of stimulus-related information and thus discriminability in the brain. Shorter windows provide finer temporal resolution of representative dynamics at the expense of less discriminative power. Longer windows allow achieving higher decoding accuracy and a smoother temporal profile.

For linear sliding window models like LDA, the model weights $\mathbf{W}$ reflect channel contributions. While in encoding models these weights directly quantify the importance of features to brain data, this is not the case for decoding models. Instead, the Haufe transform can be used to quantify these contributions (Haufe et al., 2014).

The Haufe transform works as follows. Let $\mathbf{W}$ be the $C \times K$ matrix of extraction filters from the linear decoding model, where C is the number of channels and K is the number of latent factors. Let $\mathbf{\Sigma_x}$ be the $C \times C$ data covariance matrix and $\mathbf{\Sigma_{\hat{s}}}$ the $K \times K$ covariance matrix of the estimated latent factors ŝ.

Then the $M \times K$ matrix $\mathbf{A}$ of activation patterns is given by:

$$\mathbf{A} = \mathbf{\Sigma}_{\mathbf{x}} \mathbf{W} {\mathbf{\Sigma}_{\hat{\mathbf{s}}}}^{-1} \tag{2.40}$$

If the estimated latent factors are uncorrelated, this simplifies to:

$$\mathbf{A} \propto \mathbf{\Sigma}_{\mathbf{x}} \mathbf{W} \tag{2.41}$$

where $\propto$ denotes proportionality. Plotting $\mathbf{A}$ as a sensor-space topography reveals spatial patterns related to stimulus discriminability. By plotting the activation maps for each sliding window decoding model within a trial, the spatiotemporal stimulus-discriminability can be jointly characterised.

An alternative method of finding spatial patterns of stimulus-discriminability would be to train a separate decoding model on the full trial of each channel, and plotting the accuracies as scalp topographies. This is similar to the temporal sliding window method but across space.

Finally, for spectral investigations one can fit separate decoding models on individual frequency bands from the wavelet transform of the data. Once trained, cross-validation accuracies can be computed to assess which frequency bands contain the most information related to stimuli. Alternatively one can create separate datasets by filtering around bands of interest and training separate models on each band-filtered dataset. These methods can also be combined with sliding window analysis or the Haufe transform to jointly characterise spectro-temporal, and spectro-spatial patterns of stimulus-discriminability.

Another useful concept is temporal generalisability (King and Dehaene, 2014), where a decoding model is trained on one time window and tested on all other time windows (within the trial). This provides a matrix of size timepoints × timepoints, elucidating which parts of the evoked response contain similar/shared information. This can be similarly extended for spatial and spectral generalisability.

2.5.4 Permutation feature importance

Permutation feature importance (PFI) is a model-agnostic approach that can quantify the contribution of input features to model performance for any black-box model (Fisher et al., 2019). By systematically disrupting features in the input and measuring the resultant change in model performance, PFI reveals how much the model relies on each part of the input. This provides an alternative to the multivariate pattern analysis (MVPA) methods described earlier, with the advantage that PFI can be applied to nonlinear models without needing to constrain the input domain or dimensionality. Ablation approaches such as MVPA involve completely removing a feature from the model and re-evaluating performance. While this gives a more direct measure of the impact of that feature, it requires retraining models multiple times. PFI and ablation methods methods can be viewed as complementary and useful in their own right.

For a trained encoder model g with parameters θ, the PFI approach quantifies each feature's contribution Δp_f by measuring the change in mean squared error when feature f is randomly permuted across trials:

$$\Delta p_f = \mathbb{E}_{\mathbf{X},\mathrm{s}} \left[|\mathbf{X} - g(\mathrm{s}_{\perp f}; \theta)|_2^2 \right] - \mathbb{E}_{\mathbf{X},\mathrm{s}} \left[|\mathbf{X} - g(\mathrm{s}; \theta)|_2^2 \right] \tag{2.42}$$

where $\mathrm{s}_{\perp f}$ denotes random permutation of feature f in input s. A higher Δp_f indicates feature f is more important for encoding model performance.

PFI has been applied to M/EEG data to reveal how linguistic properties such as word length and frequency affect the encoding of language responses at different latencies and sensors (Chehab et al., 2022). It demonstrated that word length impacts early encoding, while word frequency affects later encoding, matching known stages of linguistic processing.

For a trained decoder model g, PFI can localise which parts of the M/EEG input are most relevant for predicting specific evoked responses. It is model-agnostic and therefore can be applied to nonlinear deep neural networks that may better capture complex stimulus-response mappings.

The importance Δp_j of each M/EEG feature j is revealed by randomly shuffling it across M/EEG trials:

$$\Delta p_j = \mathbb{E}_{\mathbf{X},\mathbf{y}} \left[\mathcal{M}(\mathbf{y}, g(\mathbf{X}_{\perp j}; \theta)) \right] - \mathbb{E}_{\mathbf{X},\mathbf{y}} \left[\mathcal{M}(\mathbf{y}, g(\mathbf{X}; \theta)) \right] \tag{2.43}$$

where $\mathcal{M}$ is the evaluation metric (e.g. accuracy for classification), and $\mathbf{X}_{\perp j}$ denotes shuffling of feature j in $\mathbf{X}$. Here, j can represent either a timepoint vector $\mathbf{x}_t \in \mathbb{R}^C$ or a channel vector $\mathbf{x}_c \in \mathbb{R}^T$ within the full input $\mathbf{X} \in \mathbb{R}^{C \times T}$.

Higher values of Δp_j indicate that feature j is more relevant for the decoding objective. Applying PFI to $\mathbf{x}_t$ across all timepoints $t \in 1, \ldots, T$ yields an accuracy loss timecourse, revealing discriminative temporal dynamics. Similarly, applying it to $\mathbf{x}_c$ across all channels $c \in 1, \ldots, C$ creates channel accuracy loss map, localising spatial importance.

Spatiotemporal PFI jointly shuffles feature windows spanning both time and space, $\mathbf{j} = \mathbf{X}[c : c + k, t : t + l]$. This is repeated across all windows by indexing across channels c, with spatial window length k, and timepoints t, with temporal window length l. Since the channel ordering in $\mathbf{X}$ may not follow a locality-sensitive layout, methods that incorporate 3D sensor locations could improve the spatial windowing. Overall, spatiotemporal PFI reveals the joint relevance of time and space for stimulus discriminability.

To be clear PFI does not solve the issues raised by Haufe et al. (2014), as the absence of influence of a feature on decoding performance does not necessarily imply that that feature does not contain stimulus-related information. Furthermore if two features are correlated the model might use both to drive decoding, when in reality only one of the features may reflect stimulus-related activity. PFI is often used throughout the thesis and it is further detailed in Chapter 3.

2.5.5 Interpreting neural networks

Interpreting deep learning models becomes more challenging because of the nonlinear functions and high-dimensional nature of the parameter space. Their complexity poses challenges for interpreting what is learned by the models and relating this to neuroscientific principles. This section describes techniques to extract insights from trained deep encoder and decoder models into the representations and computations captured by the networks. Gaining such understanding helps link the models to underlying neural mechanisms. As mentioned, PFI can be used to investigate the input feature importances in any nonlinear model.

A common approach for understanding model decisions is to backpropagate from the output to the input layer to identify salient input patterns via gradients (Simonyan et al., 2013). Consider a deep neural network decoder $f(\mathbf{X};\theta)$ that maps an input MEG trial $\mathbf{X} \in \mathbb{R}^{C \times T}$ to predicted stimulus features $\hat{\mathbf{y}} \in \mathbb{R}^{D}$, where θ are learned model parameters, and D is the dimensionality of predicted stimulus, or the number of classes in case of classification.

The gradient of the loss $\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y})$ (e.g., cross-entropy) between the predicted ($\hat{\mathbf{y}}$) and true stimulus ($\mathbf{y}$) quantifies how small changes in the model input affect the loss. By backpropagating these gradients to the input layer (Yosinski et al., 2015), we can compute the input gradient image:

$$\mathbf{G} = \frac{\partial \mathcal{L}}{\partial \mathbf{X}} = \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{X}} \tag{2.44}$$

where $\mathbf{G} \in \mathbb{R}^{C \times T}$. This gradient image highlights channels and timepoints most relevant for the model's predictions (Sturm et al., 2016). Alternatively, one can compute saliency maps by taking the absolute gradient magnitude $\mathbf{S} = |\mathbf{G}|$ (Simonyan et al., 2013). Visualising the gradient image or saliency map reveals spatiotemporal patterns the model relies on for stimulus decoding.

For CNN decoders or forecasting models, visualising the learned filters reveals localised temporal patterns the network detects in the input. Consider a 1D temporal CNN with the following convolutional layer:

$$\mathbf{h}_m^{(l+1)} = f^{(l)} \left(\mathbf{b}_m^{(l)} + \sum_{i=1}^{M^{(l)}} \mathbf{w}_{m,i}^{(l)} * \mathbf{h}_i^{(l)} \right) \tag{2.45}$$

where $\mathbf{h}_i^{(l)} \in \mathbb{R}^T$ is the i^{th} input channel, $\mathbf{w}_{m,i}^{(l)} \in \mathbb{R}^K$ is a learned 1D kernel of length K, $*$ denotes convolution, $\mathbf{b}_m^{(l)}$ is a bias term, and $f^{(l)}$ is a nonlinearity.

Visualising the kernel weights $\mathbf{w}_{m,i}^{(l)}$ shows what patterns the network extracts from each input channel i to form output feature channel m. Analysing filter activations on input examples highlights what specific features are detected (Zeiler and Fergus, 2014). Comparing filters across layers reveals hierarchical feature learning that transforms raw signals to higher-level stimulus representations (Yosinski et al., 2015).

The internal representations learned within deep neural networks can be analysed by examining activations $\mathbf{H}^{(l)} \in \mathbb{R}^{M^{(l)} \times T^{(l)}}$ at each layer l. Dimensionality reduction techniques like PCA can visualise the geometry of high-dimensional activations (Rauber et al., 2016).

Representational similarity analysis characterises the geometry of learned neural representations using pairwise distances between activity patterns (Kriegeskorte et al., 2008). Consider a decoder $f(\mathbf{X}) = \hat{\mathbf{y}}$ predicting stimulus class $\hat{\mathbf{y}} \in 1, \ldots, K$ from M/EEG input $\mathbf{X}$. Its layer-wise representations can be analysed by first computing the mean representation pattern $\bar{\mathbf{h}}_k^{(l)}$ for each layer l for each class k by averaging over trials. Then the representational dissimilarity matrix between mean patterns ($\forall k, m \in 1, \ldots, K$) can be constructed:

$$\mathrm{RDM}^{(l)}(k, m) = d(\bar{\mathbf{h}}_k^{(l)}, \bar{\mathbf{h}}_m^{(l)}) \tag{2.46}$$

where d is a distance metric like Euclidean distance. Visualising and comparing RDMs across layers provides insight into the decoding model's discriminative space (Kriegeskorte et al., 2008). Comparing to brain RDMs (constructed from evoked responses for example) tests representational alignment between models and neural data (Kietzmann et al., 2019).

The frequency sensitivity of neural networks can be examined by analysing their spectral characteristics. For a 1D temporal convolutional layer, its filters $\mathbf{w}_{m,i}^{(l)} \in \mathbb{R}^K$ act as finite impulse response (FIR) filters on the input. Taking the discrete Fourier transform gives the frequency response:

$$W_{m,i}^{(l)}(f) = \mathcal{F}\{\mathbf{w}_{m,i}^{(l)}\} \tag{2.47}$$

Comparing spectral profiles across layers and kernels reveals what oscillations the model captures. Power spectral analysis can also be applied to network activations

$\mathbf{H}^{(l)}$ to examine what oscillations emerge internally, providing insights into the model's spectral representations.

Interpreting encoding and decoding models is crucial for relating predictions to underlying neural processes. Techniques like input backpropagation, layer analysis, representational geometry, and spectral analysis enable understanding hierarchical computations and representations learned by deep neural networks applied to M/EEG data. This links data-driven models with neurophysiological principles to advance mechanistic understanding of the brain. Ongoing research in interpretable deep learning will further bridge predictive and explanatory modelling of brain function (Samek et al., 2019).

3 | Interpretable full-epoch decoding

As described in the introduction the aim of the thesis is to deal with the various variability issues in M/EEG data. In this first content chapter we aim to tackle within-subject variability, and push the performance of within-subject decoding while also providing methods for neuroscientific interpretability. In later chapters we will build on these methods and investigate the use of deep learning to deal with more challenging types of variabilities.

Multivariate pattern analysis (MVPA) of MEG and EEG is a valuable tool for understanding how the brain represents and discriminates between different stimuli. Identifying the spatial and temporal signatures of stimuli is typically a crucial output of these analyses. Such analyses are mainly performed using linear, pairwise, sliding-window decoding models. These allow for relative ease of interpretation, e.g. by estimating a time-course of decoding accuracy, but are computationally intensive and can have limited decoding performance. On the other hand, full epoch decoding models, commonly used for brain-computer interface (BCI) applications, can provide better decoding performance. However, they lack methods for interpreting the contributions of spatial and temporal features.

In this chapter, we propose an approach that combines a multiclass, full epoch decoding model with supervised dimensionality reduction, while still being able to reveal the contributions of spatiotemporal and spectral features using permutation feature importance. Crucially, we introduce a way of doing supervised dimensionality reduction of input features within a neural network optimised for the classification task, improving performance substantially. We demonstrate the approach on 3 different task MEG datasets using image presentations. Our approach consistently achieves higher accuracy than the peak accuracy of a sliding window decoder while estimating the relevant spatiotemporal features in the MEG signal. Finally, we show that our multiclass model can also be used for pairwise decoding (in Appendix A.1.1), eliminating the computational burden of training separate models for each pairwise combination of stimuli.

Note: Most of this chapter is part of a published paper (Csaky et al., 2023a). All of the work has been carried out by the thesis author. Most experiments in this chapter can be reproduced using the associated GitHub repository[1].

3.1 Introduction

Decoding studies tend to prioritise increasing the discriminatory power (accuracy) between stimuli, e.g., in brain-computer interface (BCI) applications (Koizumi et al., 2018; Cooney et al., 2019a; Défossez et al., 2022), or gaining interpretable insights as to where and when stimuli are represented in the brain (Cichy et al., 2014, 2016). These latter approaches are often referred to as multivariate pattern analysis (MVPA), and typically make use of linear, sliding-window decoders. This allows for the extraction of the interpretable spatiotemporal features that drive the decoding; for example, allowing for the estimation of a decoding accuracy time course (Cichy et al., 2014, 2016; Cichy and Pantazis, 2017; Lappe et al., 2013; Higgins et al., 2022b,a). However, it has been demonstrated that, as one would expect, discriminatory power is also important for the effectiveness of MVPA (Guggenmos et al., 2018). Hence, there is a need in MVPA for decoding methods that improve decoding performance, while maintaining the ability to reveal the spatiotemporal features that underlie the decoding.

One possibility for increasing decoding performance is to abandon the use of sliding window approaches and instead use full epoch decoding. Here, we re-

[1] https://github.com/ricsinaruto/MEG-transfer-decoding

fer to the 500ms following stimulus presentation as the full-epoch. While it is generally good to increase the time window for decoding, as we will later show in the results, using a longer window than 500ms might actually be detrimental. Decoding full-epoch trials has been explored most typically within the context of potential (asynchronous) brain-computer interface (BCI) applications, for example in language tasks (Koizumi et al., 2018; Cooney et al., 2019a,b; Hultén et al., 2021; Dash et al., 2020a; Défossez et al., 2022) and motor tasks (Schirrmeister et al., 2017a; Dash et al., 2020b; Elango et al., 2017). It should be noted that in synchronous BCI paradigms sliding window decoding might be preferable to minimise prediction lag.

In contrast with the decoding employed in MVPA, BCI applications often use nonlinear multiclass models (Lawhern et al., 2018). These will generally have good discriminatory power (accuracy), but this comes at the expense of poor interpretability, and are thus not directly useful for MVPA.

Within BCI research, dimensionality reduction is often done with established supervised methods such as Common Spatial Patterns (CSP) (Blankertz et al., 2007), or Riemannian classifiers (Barachant, 2014). Supervised variants of PCA have also been introduced, but not for MEG data (Kobak et al., 2016). A gold standard approach to designing BCI decoders is the use of a Riemannian classifier that also performs a supervised class separation (Barachant, 2014). Importantly, these methods rely on a separate feature extraction step before applying the classifier. We wanted to include both steps in a single neural network to allow end-to-end optimisation for the classification task. We have found that the features learned by the neural network can be used to also train a standard LDA model, increasing performance substantially over either unsupervised feature reduction or the supervised Riemannian method.

Some promising approaches have been investigated recently to make full-epoch models more interpretable, such as the linear forward transform (Haufe et al., 2014). However, this approach can only be applied to linear models. Another option is to apply full-epoch and sliding window decoding on the same data in order to get both perspectives, e.g. in (Ling et al., 2019). Nonetheless, it would be hugely beneficial if a single decoding approach could be used without a loss in performance on both BCI and MVPA.

An additional consideration is computational efficiency. MVPA of MEG data is commonly performed using pairwise decoding methods, i.e. they decode between just two classes at a time (Cichy et al., 2014, 2016; Cichy and Pantazis, 2017; Higgins et al., 2022b,a). When the number of classes gets large, this becomes computationally burdensome. Here, we propose to overcome this through the use of multi-class decoding.

Taking together the aforementioned issues, we propose an approach that can improve decoding accuracy through the use of full-epoch multi-class decoding, while still being able to reveal the underlying spatiotemporal features that drive the decoding. This allows us to consider and investigate the use of neural network decoding models, and we also show the benefit of using supervised feature reduction. We limit our investigations to linear models, leaving nonlinear models for future work. Importantly, to allow access to interpretable features, we make use of permutation feature importance (PFI).

We assess the proposed approach by systematically comparing it with sliding window decoding on three MEG datasets with visual tasks, finding that our full-epoch decoding outperforms sliding window decoding in terms of accuracy. We then compare PFI with standard alternatives and find that PFI is able to extract the same kind of dynamic temporal, spatial, and spectral information. In addition, we show that pairwise accuracies can easily be gained from a single multiclass model and that these accuracies are on-par with a direct pairwise classification approach. Please see Appendix A.1.1 for these results.

In short, the aforementioned contributions achieve the best of both worlds: a single decoding model trained on full epochs, empirically good performance, and clear interpretability from an MVPA viewpoint. This approach promises to be useful for both the BCI researcher and the neuroscientist trying to gain insight into the underlying brain activity in a particular task and external stimuli set.

3.2 Methods

3.2.1 Data

Here, we used three visual MEG datasets: two similar datasets from Cichy et al. (2016) and one additional dataset from Liu et al. (2019). The datasets have been collected with appropriate consent from participants and ethical review by Cichy et al. (2016) and Liu et al. (2019), and do not contain any personal information. 15 subjects view 118 and 92 different images, respectively in the first two datasets, with 30 repetitions for each image. The third dataset is part of a larger replay study, and we only use the portion of the data where images are presented in random order for 900ms. Here, 22 subjects view 8 different images, with 20-30 repetitions for each image (depending on the subject). The image sets used in the three datasets are different.

We obtained the raw MEG data directly from the authors to run our preprocessing pipeline with MNE-Python (Gramfort et al., 2013). The 118-image and 92-image data are also available publicly in epoched form[2]. We bandpass filtered raw data

[2] http://userpage.fu-berlin.de/rmcichy/fusion_project_page/main.

between 0.1 and 25Hz and downsampled to 100Hz. As recommended by prior work the sampling rate is 4 times higher than the lowpass filer (Higgins et al., 2022b). This is done so that representational alias artefacts are eliminated from the sliding window decoding time courses. We also applied whitening, which involved transforming the data with PCA to remove covariance between channels while retaining all components. The PCA was fit on the training set only but applied to both training and test sets.

Many papers have shown that visual information processing in the brain primarily operates in lower frequency ranges. Specifically, theta (4-7Hz), alpha (8-12Hz), and beta (13-30Hz) bands have been implicated in various aspects of visual processing, including object recognition, visual attention, and perceptual decision-making (Klimesch, 1999; Engel and Fries, 2010; Zoefel and VanRullen, 2017). Therefore, a lowpass filter of 25Hz captures these important frequency bands while reducing the influence of higher frequency signals that are less likely to be related to visual processing.

MEG data, like all bioelectrical signals, are often contaminated by various sources of noise. High-frequency noise, particularly above 30Hz, often originates from sources outside the brain, such as muscle activity or environmental electromagnetic fields (Gross et al., 2013). By using a 25Hz lowpass filter, we can significantly reduce these non-brain noise contributions, thereby improving the signal-to-noise ratio and enhancing the detectability of the brain's visual responses.

While there are meaningful neuronal signals at frequencies above 30Hz (e.g., gamma-band activity), decoding these high-frequency signals from MEG data can be challenging due to lower signal-to-noise ratios. Therefore, unless the specific research question involves high-frequency bands, applying a 25Hz lowpass fil-

html

ter simplifies the data and focuses the analysis on the most relevant and easily interpreted signals. It also allows reducing the sampling rate, and thus the dimensionality of the data which is an important factor for achieving good classification performance with machine learning. In Chapter 4 we also train decoding models on this dataset using a higher lowpass filter and do not observe improvement in classification accuracy.

In the first two datasets, image presentation lasted for 500ms with an average inter-trial interval of 0.95 seconds. In order to analyse the data using machine learning models, we created two versions of each dataset. The first version consisted of full epochs, with input examples having a shape of [50, 306] (or [90, 273] for the 8-image dataset), where 306 and 273 correspond to the number of MEG channels and 50 and 90 correspond to the number of time points during image presentation. The second version consisted of sliding windows, with input examples having a shape of [10, 306] (or [10, 273] for the 8-image dataset). In this case, we partitioned each trial into overlapping 100ms time windows between 0 and 1000ms post-stimulus and trained separate models on each time window partition as is normally done in the MVPA literature. The difference between consecutive windows was 1 timestep (10ms). As a result, 90 independent sliding window models were trained for each dataset. In the rest of this chapter we use the term *raw* to refer to the pre-processed time domain signal, as opposed to other non-time domain input features.

As opposed to some previous work using a wavelet transform of the trial as features for sliding window decoding (Higgins et al., 2022b), here we use the raw set of timepoints within the respective 100ms window. This means that we rely more on the decoder to extract relevant frequency information rather than directly providing such information in the input. A more recent approach, termed superlets transform (Moca et al., 2021; Jorntell and Kesgin, 2023) has been shown to improve classification results by mitigating the time vs. frequency resolution problem (Bârzan et al., 2022). However, our main comparison between sliding-window and full-epoch decoding is performed at the raw data level. Training decoding models on raw data is also beneficial for our goals of using deep learning later in the thesis. By supplying raw data we do not make any assumptions about the types of features that should be used, but rather delegate this task to the model.

3.2.2 Neural network

The Neural network (NN) model in this chapter is a four-layer, fully-connected linear neural network which is only run on the full-epoch dataset (Figure 3.1). The first layer performed a learnable dimensionality reduction, where the full epoch data $\mathbf{X} \in \mathbb{R}^{T \times C}$ was multiplied by a weight matrix $\mathbf{W} \in \mathbb{R}^{C \times K}$, where K was set to 80. This is similar to the projection in principal component analysis, but in this case, the projection and the decoding model are trained simultaneously; therefore, the dimensionality reduction is optimised for the classification objective. To be clear, the input size to the first layer, and thus the dimensionality of this layer, depends on the time window size and number of channels which can be different for each dataset. After the first layer, the data was flattened and three affine transformations were applied in sequence (see Figure 3.1 for dimensionalities). The final layer had an output dimension equal to the number of classes, and the logits from this layer were passed through a softmax function for classification. We chose the intermediate hidden sizes (1000 and 300) to be roughly equally distanced (multiplicatively) between the input and output dimensions of the network (4000 and 118). This rationale was employed for the 118-image dataset primarily and we did not change the hidden sizes for the other two datasets.

The model was trained using cross-entropy loss (Good, 1952) for multiclass classification and included dropout between layers during training (Srivastava et al., 2014). It is worth noting that, as no nonlinearities were used, the model could be replaced

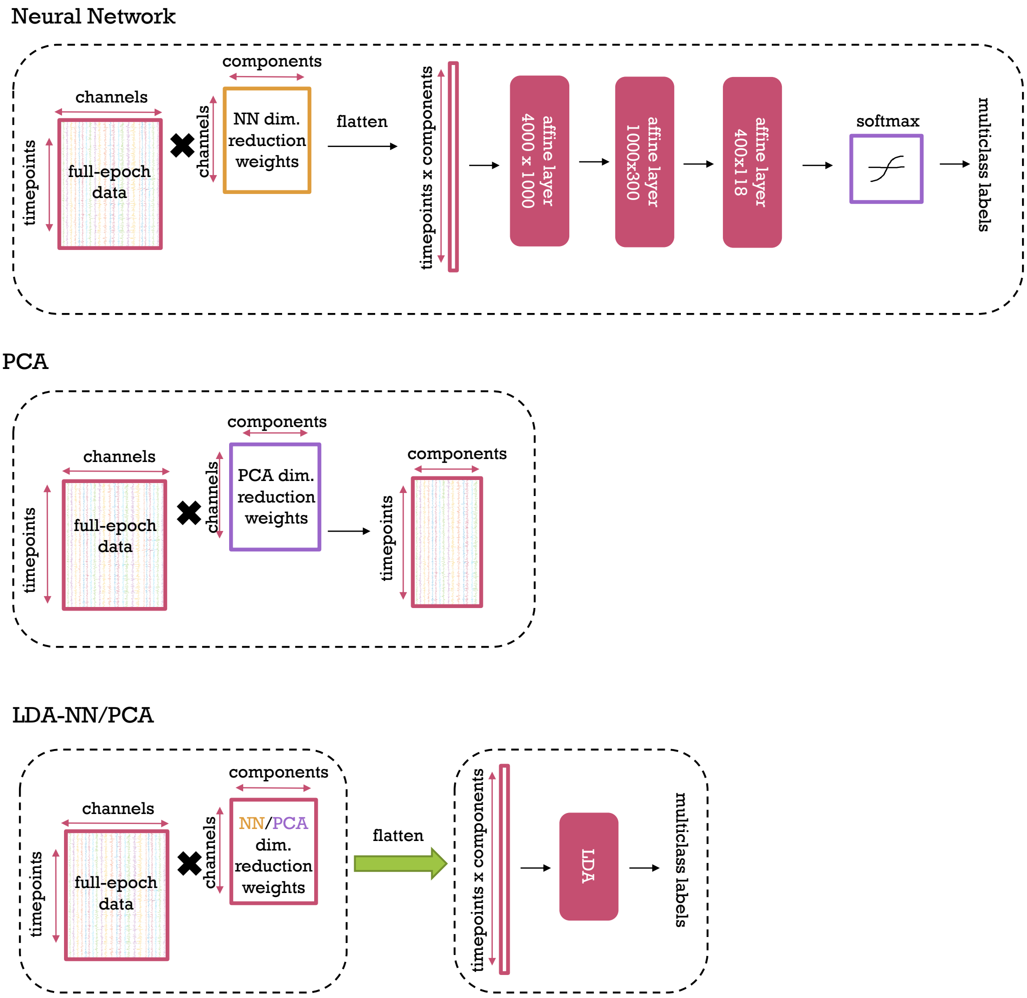


Figure 3.1: Our Neural Network, PCA, and LDA-NN/PCA methods from top to bottom. Dashed boxes represent separate processing steps, i.e. in the case of LDA-NN and LDA-PCA the respective dimensionality reduction is first used to compute the input features, which are then used to train the LDA model.

with a single affine transformation during evaluation. However, deep linear neural networks are known to have nonlinear gradient descent dynamics that change with each additional layer (Saxe et al., 2013); both the learnable dimensionality-reduction layer and the use of dropout impose additional constraints on the weight matrix during learning.

3.2.3 LDA-PCA

The LDA-PCA approach has two variants: one that is full-epoch, and one that uses a sliding window. In the full-epoch version, PCA is used to do unsupervised dimensionality reduction on the channel dimension of the full-epoch data as an initial, separate step (Figure 3.1). The resulting PCA-reduced data matrix $\mathbf{H} \in \mathbb{R}^{T \times K}$ is flattened and then used to train a multiclass classifier using LDA.

In the sliding window version, the $\mathbf{H}$ matrix is separated into sliding window matrices $\mathbf{H}_t = \mathbf{H}[t : t + d, :]$, where d was set to 100 ms, and $t \in 1 \ldots T - d$. The data within each window is then flattened in the same manner as in the full-epoch version and fed into separate LDAs that are distinct to each window. The window size of 100 ms was selected due to its previous use in the MVPA literature (Higgins et al., 2022b), and because it provides a good trade-off between resolution and accuracy. However, we also explore different window sizes in the results.

3.2.4 LDA-NN

In the LDA-NN method, the dimensionality-reducing weight matrix from PCA is replaced with the use of the dimensionality-reducing weight matrix extracted from the pre-trained NN approach (Figure 3.1). As in LDA-PCA, this projection is then applied to the input data $\mathbf{X}$. The LDA model is applied to the resultant (flattened) features $\mathbf{H} \in \mathbb{R}^{T \times K}$. In the same manner, as LDA-PCA, LDA-NN also has full-epoch and sliding window versions.

3.2.5 Permutation feature importance

To investigate the temporal dynamics of visual information processing, we utilised permutation feature importance (PFI) on our trained models. Specifically, we applied PFI to a trained full-epoch LDA-NN by using sliding windows of 100ms with 1 time point shift for each trial. The information in each window was disrupted by permuting the data across the channel dimension separately for each time window. For instance, if the window was centred around 50ms post-stimulus, the information within that window would be disrupted from 0 to 100ms post-stimulus compared to the original trial, while the rest of the timepoints in the trial remained unchanged. We then evaluated the trained LDA-NN on each of these disrupted trials and compared the accuracy to the original accuracy obtained with the original trials. The greater the accuracy decrease for a trial with disrupted information in a specific time window, the more crucial that time window is to the model's performance and, therefore, the more information it contains relevant to the model's objective of discriminating between images. By repeating this analysis for all time windows, we obtain a temporal profile of the information content, similar to the method of training separate models on individual time windows.

In terms of assessing spatial information content, we followed a similar methodology, albeit with modifications. Here, the disruption involved permuting the data across time points within each channel individually. The outcome of this operation is a sensor space map detailing the decrease in accuracy, which serves as a metric for the visual information content. This map was then compared with others generated by evaluating the per-channel accuracy of individual LDA models trained on the full epoch of each respective channel. Conceptually, this method can be seen as sliding a window (or "search-light") across the spatial domain, similar to the previous time-based approach. In practice, we ran spatial PFI across sensors (2 gradiometers and 1 magnetometer in the same position) instead of channels, thus permuting these 3 channels together and obtaining a single metric for them. This allows for more robust results.

An alternative would be to permute the gradiometers and magnetometers separately but using a spatial neighbourhood of nearby sensors for smoothing. Due to differences in spatial sensitivity between magnetometers and gradiometers this approach could result in spatial maps with higher resolution. Magnetometers measure the magnitude of the magnetic field and are most sensitive to sources directly below the sensor. Gradiometers on the other hand measure the difference (spatial gradient) of the field and thus are more sensitive to sources slightly to the sides of the sensor. Because they measure a difference, the signal is less affected by magnetic fields coming from outside the head (if spatially homogeneous), and thus decoding models trained on gradiometers usually perform better than those trained on magnetometers.

Additionally, we illustrated the extraction of spatiotemporal information by utilising PFI. The method involved choosing a window that spanned both space (across 4 sensors with 2 gradiometers and 1 magnetometer each, totalling 12 channels) and time (a 100ms window) simultaneously. The spatial window contains the 2 gradiometers and 1 magnetometer on three sides of the sensor in question. By sliding the window and performing the shuffling across all timepoints and channels we obtained a spatiotemporal discriminative information content profile. This comprehensive profile allowed us to understand how the disruption of specific spatiotemporal windows impacts the performance of the trained model, therefore highlighting the importance of those windows in discriminating between visual stimuli.

Finally, we introduce *spectral PFI* to assess the effect of different frequency bands on the visual discrimination objective. Let $\mathbf{X} \in \mathbb{R}^{C \times T}$ be the input trial in the time domain. First, we apply the Fourier transform to each channel:

$$\mathbf{Z} = \mathcal{F}(\mathbf{X}) \tag{3.1}$$

to get the frequency domain representation $\mathbf{Z} \in \mathbb{C}^{C \times F}$, with F frequency bands. Then each frequency band f is randomly shuffled across the channel dimension, and the disrupted trial is transformed back to the time domain:

$$\hat{\mathbf{X}} = \mathcal{F}^{-1}(\mathbf{Z}_{\perp f}) \tag{3.2}$$

where $\mathbf{Z}_{\perp f}$ denotes random shuffling of feature (frequency band) f in $\mathbf{Z}$. Note that the frequency band can either refer to a single value in $\mathbf{Z}$ or to a window (l) of frequencies $\mathbf{Z}[:, f : f + l]$. The window length provides a trade-off between smoothness, power, and specificity of the frequency profile we obtain. The importance Δp_f of each frequency band/window f is revealed by comparing the accuracy of disrupted trials with original trials:

$$\Delta p_f = \mathbb{E}_{\hat{\mathbf{X}},\mathbf{y}} \left[\mathcal{M}(\mathbf{y}, g(\hat{\mathbf{X}}; \theta)) \right] - \mathbb{E}_{\mathbf{X},\mathbf{y}} \left[\mathcal{M}(\mathbf{y}, g(\mathbf{X}; \theta)) \right] \tag{3.3}$$

where $\mathcal{M}$ is our metric of goodness, accuracy, g is the trained model with θ parameters, and $\mathbf{y}$ are the target classes.

By applying this method to all frequency bands, we obtained a spectral information content profile, similar to the method of training separate LDA models on features from individual frequency bands (Higgins et al., 2022b). Similar to spatiotemporal PFI we can combine spatial and spectral PFI, by running spectral PFI on a neighbourhood of 4 sensors at a time (spatial window) to assess the spectral information content of individual MEG channels. Thus the shuffled feature is $\mathbf{j} = \mathbf{Z}[c : c + k, f : f + l]$, where c is the channel index, k is the spatial window size, f is the frequency band index, and l is the frequency window width. We call this spatio-spectral PFI.

Previous work applied sliding window decoding in combination with spectral decoding (i.e., training separate models on individual frequency bands), thus assessing the temporo-spectral information content (Higgins et al., 2022b). In order to make comparisons with this work, we developed temporo-spectral PFI.

Let $\mathbf{X} \in \mathbb{R}^{C \times T}$ be the input trial in the time domain. We frst apply the short-term Fourier transform with a (Hamming) window size $w = 100\ ms$ and hop size $h = 1\ timestep$ to get the time-frequency representation matching parameters used in Higgins et al. (2022b):

$$\mathbf{Z} = \mathrm{STFT}(\mathbf{X}, w, h) \tag{3.4}$$

where $\mathbf{Z} \in \mathbb{C}^{C \times N \times F}$, with N time windows and F frequency bins. Then each feature window $\mathbf{Z}[:, n : n + k, f : f + l]$ is randomly shuffled, were n is the timepoint index, k is the temporal window length, f is the frequency band index, and l is the frequency window width. Note that both the temporal and spectral windows can be set to 1, but usually a small window is better to balance specificity and smoothness. The disrupted trial is transformed back to the time domain:

$$\hat{\mathbf{X}} = \mathrm{iSTFT}(\mathbf{Z}_{\perp n,f}) \tag{3.5}$$

where $\mathbf{Z}_{\perp n,f}$ denotes random shuffling of the feature indexed by (n, f) in $\mathbf{Z}$. The importance $\Delta p_{n,f}$ of each time-frequency window is revealed by comparing with the original trials as in Equation 3.3:

$$\Delta p_{n,f} = \mathbb{E}_{\hat{\mathbf{X}},\mathbf{y}} \left[\mathcal{M}(\mathbf{y}, g(\hat{\mathbf{X}}; \theta)) \right] - \mathbb{E}_{\mathbf{X},\mathbf{y}} \left[\mathcal{M}(\mathbf{y}, g(\mathbf{X}; \theta)) \right] \tag{3.6}$$

By repeating this over all frequency bands and time windows we obtain the temporo-spectral PFI profile.

3.2.6 Experimental details

The primary evaluation metric for the three datasets is classification accuracy across the respective number of classes (118, 92, or 8). The main focus of our analysis was on the 118 and 92-image datasets, with the 8-image dataset, included to demonstrate the effects of a much smaller sample size. Note that such a high number of classes are not commonly decodable in BCI and stimulation settings. All of the main results using our decoding methods (NN, LDA-NN, LDA-PCA) are multiclass. For all analyses, separate models were fit to separate subjects. Training and validation splits were created in a 4:1 ratio for each subject and class, with classes balanced across the splits. The NN approach was trained for 2000 epochs (full passes of the training data as opposed to epochs in the sense of MEG trials) using the Adam optimiser (Kingma and Ba, 2015). The high number of epochs was selected as this allowed the training accuracy to converge to almost 100%, while the validation accuracy also converged to a stable value for most participants. The output layer's dimensionality was equal to the number of classes in the corresponding dataset. Dropout was set to 0.7 and applied before each of the three hidden layers.

The dimensionality reduction layer and PCA were both set to 80 components, as it is slightly higher than the inherent dimensionality reduction of MaxFilter (of rank 64) which is applied to the MEG data, and thus contains more than 99% of variance. We briefly tried our pipeline with 60 components as well on 1 subject and found similar results. Thus, as long as the dimensionality reduction is at least the rank of the data there should be no loss of information and similar results should be observed. Projecting to more dimensions than the rank of the data can introduce issues such as linear dependence in the input components. However, we observed that our decoding models are robust and probably due to the large amounts of regularisation used are not affected by such issues.

Validation data was not used for early stopping, and the trained NN dimensionality reduction weight matrix (used in LDA-NN) was extracted after the full 2000 epochs of training on the training data. For the LDA models, the shrinkage parameter was set to *auto* using the sklearn package. Comparisons of interest over methods were evaluated using Wilcoxon signed rank tests, with within-subject pairing and subject-level mean accuracies over validation examples as the samples. We used Bonferroni correction to correct for multiple comparisons. The PyTorch package was used for training (Paszke et al., 2019), and several other packages were utilised for analysis and visualisation (Pedregosa et al., 2011; Virtanen et al., 2020; Harris et al., 2020; Wes McKinney, 2010; Waskom, 2021; Hunter, 2007).

3.3 Results

3.3.1 Full-epoch models better than sliding-window decoding

We set out to test whether full-epoch decoding is better than timepoint-by-timepoint and sliding-window decoding, which are common practices in the M/EEG literature (Carlson et al., 2011, 2013; Su et al., 2012; Ramkumar et al., 2013; Cichy et al., 2017; Grootswagers et al., 2017; Kurth-Nelson et al., 2016; Liu et al., 2019; Higgins et al., 2022b). We wanted to make sure that our classifier of choice, LDA is at least as good as other commonly used models for multiclass decoding, including support vector machines (SVM), linear discriminant analysis (LDA), logistic regression, and Lasso. The results, depicted in Figure 3.2, indicate that LDA and logistic regression exhibited comparable performance (no statistical

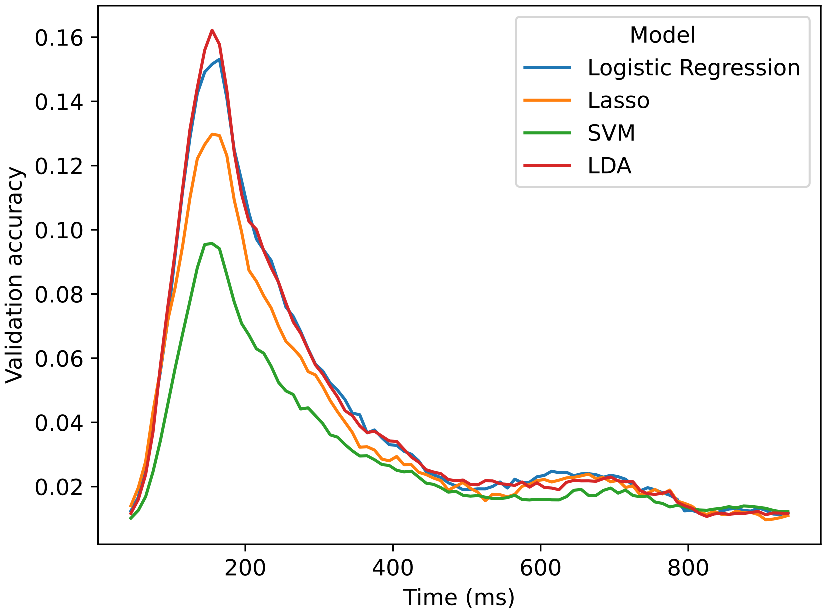


Figure 3.2: Comparing different sliding window models trained on PCA features on the 118-image dataset for multiclass decoding. The sliding window size is 100ms. Results are averaged across subjects.

difference) and performed better than the other 2 examined models. For this reason, and as described in the methods, we used LDA in all further analyses for comparing different classification strategies.

We also wanted to make sure that the choice of using raw time-domain data is not limiting decding performance. Thus we compared our raw sliding window decoding performance with the wavelet approach employed in Higgins et al. (2022b) and found the latter to be significantly worse across most timesteps (Figure 3.4).

The performance of multiclass full-epoch models was compared to that of sliding-window decoding for both LDA-PCA and LDA-NN on the three datasets in Figure 3.3. The peak performance of sliding-window decoding was observed at 150-160 ms post-stimulus for the 92 and 118-image datasets, and at 200 ms post-

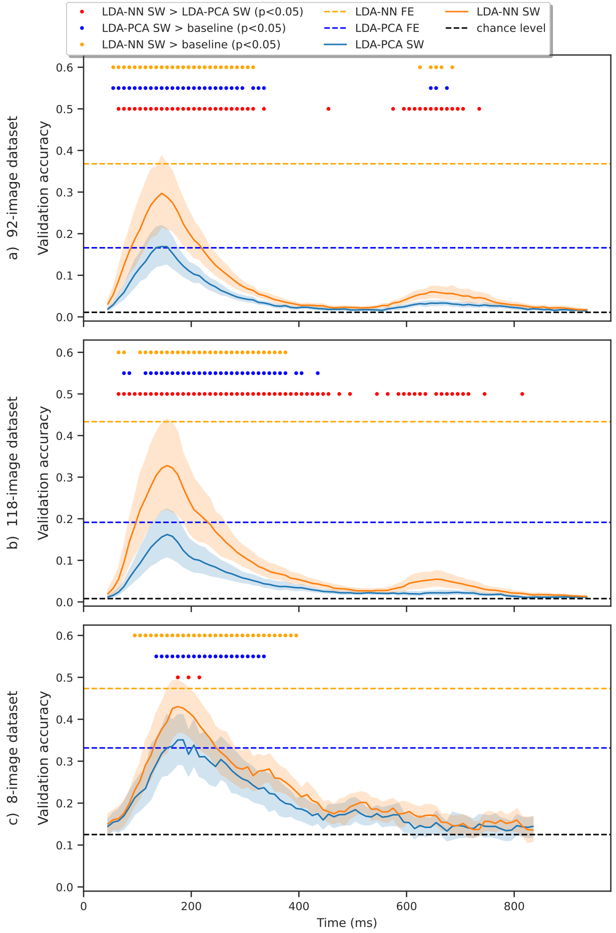


Figure 3.3: Models trained on the sliding-window versions of the 92-class dataset (top), 118-class dataset (middle) and 8-class dataset (bottom) for multiclass decoding. Wilcoxon signed-rank tests are reported between sliding window LDA-NN and LDA-PCA. We also ran Wilcoxon signed-rank tests between the first timepoint of LDA-NN and LDA-PCA and all other timepoints. This shows statistical significance compared to a "baseline" level. FE stands for full-epoch models, and SW stands for sliding window models. The blue and orange dotted lines are placed at the average performance of full-epoch LDA-NN and LDA-PCA, respectively. All statistical tests are Bonferroni corrected for multiple comparisons across all time points (i.e. p-values are multiplied by 90). Shading indicates the 95% confidence interval across subjects. For the full-epoch results, please see Figure 3.6 for distributions across subjects. LDA-NN is better across almost all time points than LDA-PCA, and full-epoch accuracy is higher than peak sliding window accuracy for both LDA-NN and LDA-PCA (except in the 92-class and 8-class datasets).

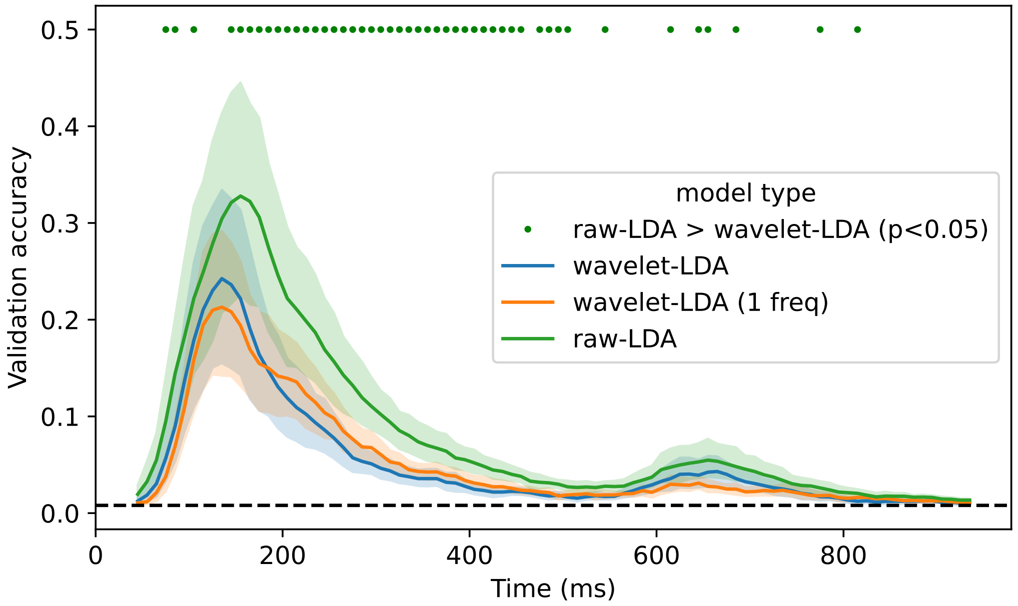


Figure 3.4: Comparison of our sliding window LDA-NN approach with LDA-NN using wavelet features on the 118-image dataset. The wavelet features are computed after the dimensionality reduction, with the same settings as in Higgins et al. (2022b). A hamming window of 10 timesteps was used with an overlap of 9 timesteps. wavelet-LDA corresponds to using a concatenation of all frequency bands for training the LDA model, and wavelet-LDA (1 freq) uses a single frequency band (10Hz). We selected this band based on previous results in Higgins et al. (2022b), achieving the best decoding performance using this band only. Results are averaged across subjects, and shading indicates the 95% confidence interval across subjects.

stimulus for the 8-image dataset. These findings are broadly consistent with previous research on the temporal dynamics of visual information processing in MEG (Cichy et al., 2014, 2016; Cichy and Pantazis, 2017; Higgins et al., 2022b; Liu et al., 2019; Guggenmos et al., 2018). For the 92 and 118-image datasets a second smaller peak was observed around 650-660 ms post-stimulus. As the image presentation is switched off at exactly 500ms, we reason that the second peak is due to the brain reacting to this event. The first peak is observed 150-160 ms post-stimulus onset, while the second peak occurs 150-160 ms post-stimulus offset.

Across subjects, the full-epoch LDA-PCA approach demonstrated significantly higher accuracy than the best sliding-window LDA-PCA approach on the 118-class dataset (3.1% increase, p < 1e-4). On the 92-class dataset, no significant difference was observed between these models, though full-epoch LDA-PCA still outperformed the sliding-window version at most time windows. A similar comparison between full-epoch LDA-NN and peak sliding-window performance showed that full-epoch models had higher accuracy on both the 92- and 118-class datasets (7.1% and 10.5% increase, respectively, p < 1e-4). The tests were corrected for multiple comparisons across time points. These results indicate that training a model on the full epoch generally leads to better performance than using the best sliding-window model, except for the LDA-PCA approach on the 92-image dataset. However, as noted in the following section, it is advisable to use an LDA-NN model in any case.

On the 8-image dataset, the full-epoch model had higher accuracy than the peak sliding-window model, though this difference was not significant. It should be noted that the reduced effectiveness of the full-epoch model on this dataset may be due to both the longer epoch of 900 ms and the smaller amount of data. This can lead to overfitting due to a larger number of features and fewer examples.

Our results could be affected by the choice of window size for the sliding window LDA (100 ms). Thus, we repeated the sliding window LDA for different window sizes, including a window of 1 sample (i.e., timepoint-by-timepoint decoding), and the results are presented in Figure 3.5. We trained models using sliding window sizes of 10ms, 100ms, 200ms, 300ms, and 400ms. As expected, using a single time point (10ms) resulted in lower accuracy compared to a 100ms window. As the window size increased, we observed two trends. First, accuracy improved and the

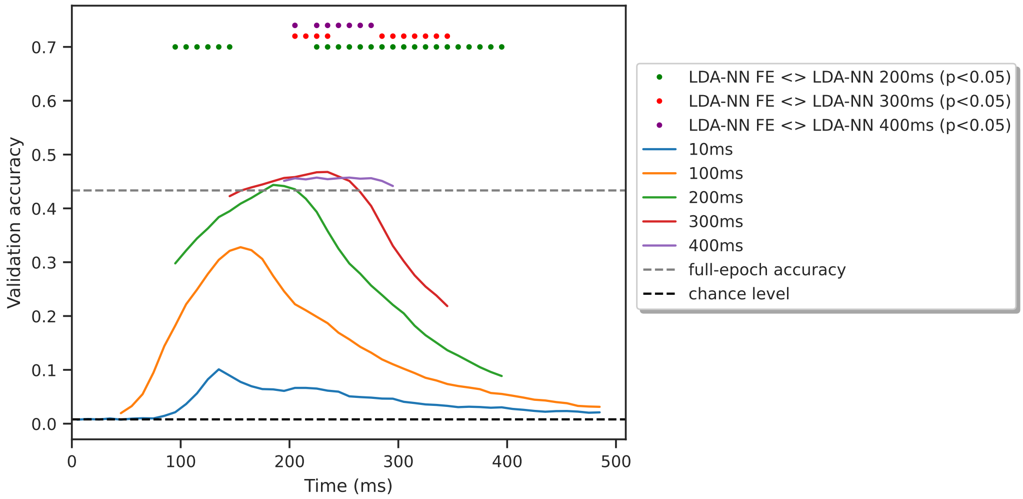


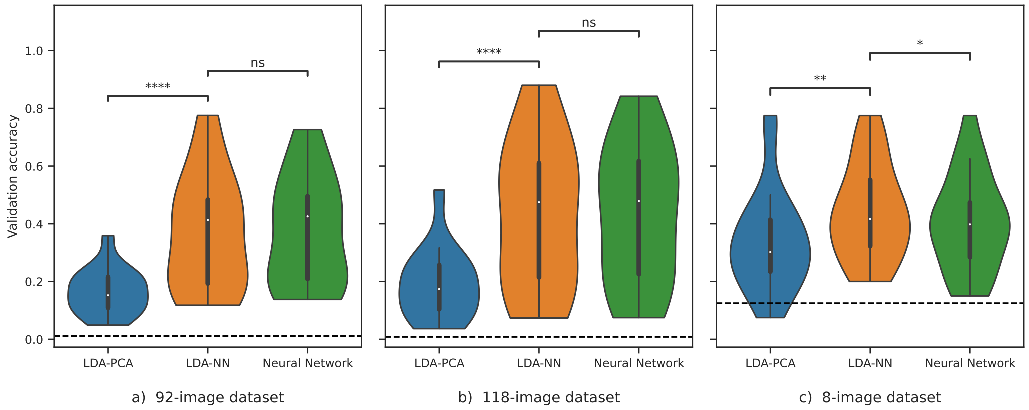
Figure 3.5: Comparing sliding window LDA-NN with different window sizes on the 118-image dataset. Results are averaged across subjects. Wilcoxon signed-rank tests are reported between the sliding window models and the full-epoch model, Bonferroni corrected for all comparisons in the figure.

peak accuracy of a 200ms window already reached the full-epoch level. Second, the accuracy profile became more distorted and the peak shifted compared to the results obtained with a single time point. In some cases, full-epoch performance was even exceeded by a few percentage points with a 300ms window. This may not be surprising, as a larger window that focuses on the most significant part of the input results in fewer features compared to using the full epoch. However, it is advisable to avoid using a window larger than 100ms in sliding window analysis due to its distortion and lower temporal resolution. One potential solution could be to combine the sliding window models with PFI analysis, but this would be inefficient. We can therefore conclude that using a full-epoch model is the optimal solution, even if it results in slightly lower accuracy. Additionally, we expect that with larger datasets, full-epoch models would outperform sliding window models regardless of window size, as the ratio of features to examples would be reduced.

3.3.2 Supervised dimensionality reduction is better than PCA

We next investigated the effect of incorporating a learned, supervised dimensionality reduction layer in our models, i.e. a dimensionality reduction optimised to aid a downstream classification task. We, therefore, modified the LDA-PCA approach by replacing the unsupervised dimensionality reduction performed by PCA with the supervised dimensionality reduction (of equal dimensionality) from the Neural Network (NN) approach, as described in Section 3.2. We refer to this modified approach as LDA-NN. As shown in Figure 3.6, this simple change resulted in a significant improvement in performance (20.2% for the 92-class dataset and 24.2% for the 118-class dataset, $p < 1e-4$). We also assessed the performance of the pure NN model and found that it has a similar performance to LDA-NN. In other words, the supervised dimensionality reduction effectively eliminated the performance gap between the LDA and the Neural Network (NN) approach.

The sliding window versions of LDA-PCA and LDA-NN are also compared in Figure 3.3. Across most time points (and all time points around the 2 peaks), LDA-NN is significantly better than LDA-PCA, when Bonferroni corrected for multiple comparisons across time points. Similar conclusions can be drawn on the 8-image dataset, although LDA-NN is better than the NN approach, possibly due to the reduced performance of neural networks on small datasets in general. In summary, our results suggest that using a full-epoch LDA-NN or a simple linear Neural Network results in the best performance across all datasets and that the feature reduction should be learned in a supervised manner for both the LDA and Neural Network models.



a) 92-image dataset

b) 118-image dataset

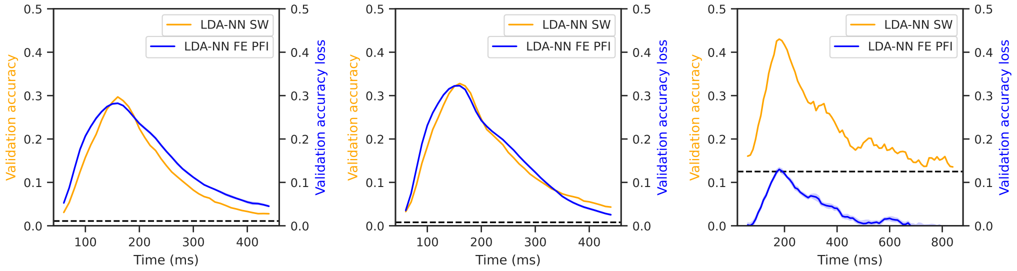
c) 8-image dataset

Figure 3.6: Models trained on the full-epoch versions of the 92-class (left), 118-class (middle), and 8-class (right) datasets for multiclass decoding. The violin plot distributions are shown over the mean individual subject performances. The dashed black line represents the chance level. Wilcoxon signed-rank tests are shown where 4 stars mean p < 1e-4, and 3 stars mean p < 1e-3. “ns” means that the p-value is higher than 0.05.

3.3.3 Temporal PFI

One of the benefits of sliding window or time-point-by-time-point decoding is that it is straightforward to obtain a time course of decoding accuracy (e.g., Figure 3.3), allowing for interpretation of the temporal dynamics of neural representations. Here we show that full epoch decoding in combination with permutation feature importance (PFI) can give the same qualitative information. The results presented in Figure 3.7 indicate that temporal PFI applied to a full-epoch LDA-NN model produces temporal profiles similar to those obtained using sliding window LDA-NN models with a window size of 100ms across all three datasets. The peak sliding window performance also aligns well with the peak accuracy loss for PFI.

We investigated an alternative method of performing PFI, referred to as inverse PFI. This method is not common in the literature, but could be interesting from an MVPA viewpoint. Inverse PFI differs from standard PFI in that it shuffles values outside a specified time window, rather than within it. Standard PFI assesses the



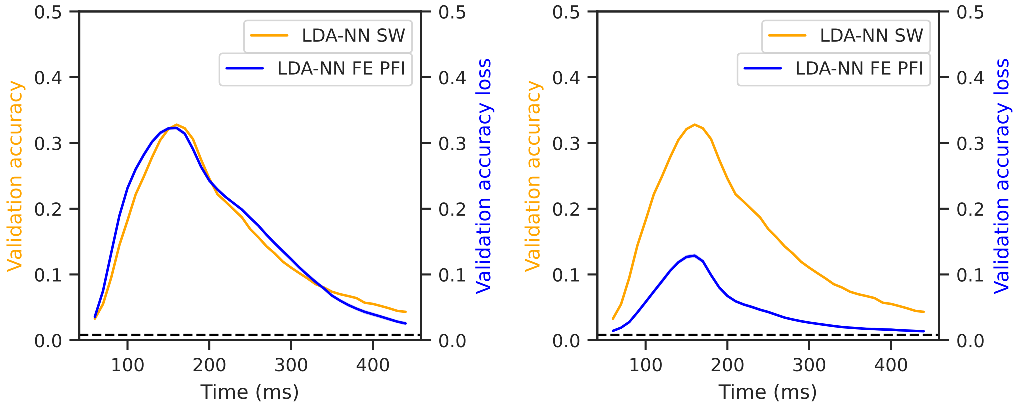
(a) 92-image dataset (b) 118-image dataset (c) 8-image dataset

Figure 3.7: Comparison of multiclass sliding window LDA-NN (orange) and the temporal PFI of multiclass full-epoch LDA-NN (blue) across the three datasets. Results are averaged across all subjects in the respective datasets, and shading indicates 95% confidence interval across permutations for PFI. Chance level for LDA-NN SW is indicated with a dashed line.

impact of disrupting information within a specific window on performance and therefore reveals the importance of that window for discriminating between images. In contrast, inverse PFI investigates performance when all information outside a specified window is disrupted, thereby providing insight into the performance that can be achieved using only the information contained within the time window. The temporal PFI results for both standard and inverse PFI are presented in Figure 3.8. While both approaches are similar to the standard sliding window LDA profile, there are some differences as well.

3.3.4 Spatial PFI

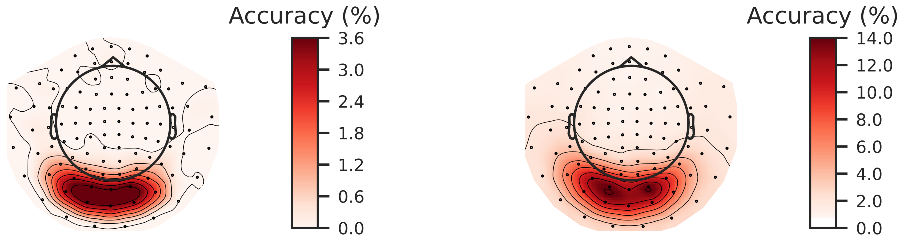
We investigated the ability of PFI to accurately capture spatial information by applying it to a full-epoch LDA-NN model on the 118-image dataset. To do this, we permuted time points from the gradiometers and magnetometers located at the same position in the MEG data simultaneously to obtain a single sensor space map. We compared these to the maps obtained by training separate LDA models on the full epoch of the same three sensors (2 gradiometers and 1 magnetometer). This



(a) Standard PFI

(b) Inverse PFI

Figure 3.8: Comparison of multiclass sliding window LDA-NN (orange) with standard temporal PFI (a) and inversed temporal PFI (b) using a trained LDA-NN model on the 118-image dataset. Results are averaged across subjects, and shading indicates the 95% confidence interval across permutations for PFI. Chance level is indicated by the dashed line.

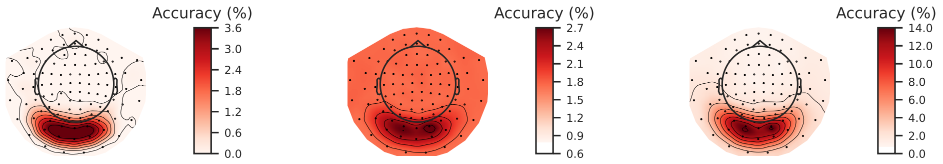


(a) Spatial PFI

(b) Channel-wise LDA

Figure 3.9: Comparison of multiclass channel-wise LDA model (b) with the spatial PFI of multiclass full-epoch LDA-NN (a). Spatial maps are averaged across all 15 subjects on the 118-image dataset. Both PFI and the channel-wise LDA model are run on 3-channels in the same location at a time (1 magnetometer and 2 gradiometers).

approach can be viewed as a sliding window across space. All PFI results are averaged over the accuracy losses of individual subjects, which can somewhat smear



(a) Standard PFI (b) Inverse PFI (c) Channel-wise LDA

Figure 3.10: Comparison of channel-wise LDA model (c) with the standard spatial PFI (a) and inverse spatial PFI (b) of full-epoch multiclass LDA-NN. Results are averaged across all 15 subjects on the 118-image dataset. Both PFI and the channel-wise LDA model are run on 3-channels in the same location at a time (1 magnetometer and 2 gradiometers).

both spatial and temporal profiles. The results, shown in Figure 3.9, demonstrate good alignment between the accuracy loss of spatial PFI and per-sensor accuracy of LDA-NN, indicating that PFI can effectively recover spatial information content.

We also conducted the inverse PFI analysis in the spatial domain, the results of which are shown in Figure 3.10. In this domain, the inverse PFI approach exhibits less contrast between visual channels and other channels but appears to distinguish between visual channels more similarly to channel-wise LDA than standard PFI. It is not the aim of this study to determine which approach is superior, as both seem to have their merits.

3.3.5 Spatiotemporal PFI

We also employed PFI to extract spatiotemporal information jointly from a trained full-epoch LDA-NN model on the 118-image dataset. Specifically, we used a 100 ms time window and a 4-channel spatial window (i.e., the 2 gradiometers and 1 magnetometer on three sides of the sensors in question) for each time point and channel, shuffling the values within these blocks. This allowed us to unravel the temporal and spatial information simultaneously, showing that only channels

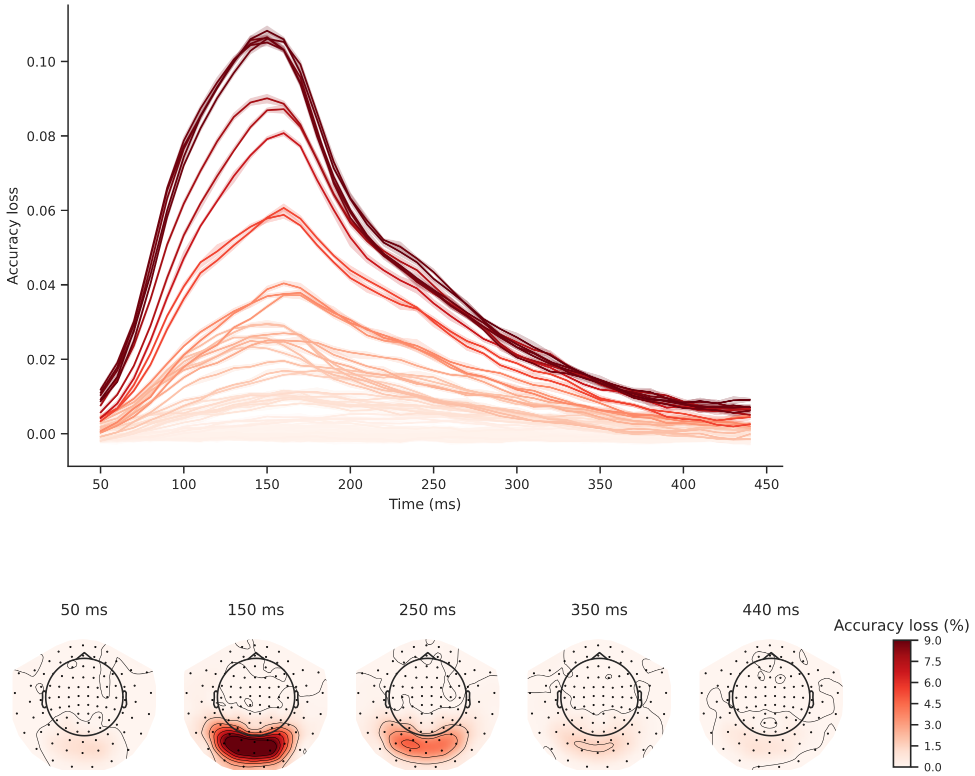


Figure 3.11: Spatiotemporal PFI of multiclass full-epoch LDA-NN on the 118-image dataset. Blocks of 4-channel neighbourhoods and 100ms time windows are shuffled to obtain a spatial and temporal profile jointly. Each line in the temporal profile corresponds to a sensor, and each sensor space map is obtained with a time window centred around the respective time point. The color map of the upper plot is based on the coloring of sensors at 150ms in the lower plot. The shading in the upper plot is across the 10 permutations used for PFI and indicates the 95% confidence interval. Both temporal and spatial profiles are averaged over subjects.

located in the visual area exhibited the characteristic temporal profile and that there was a gradient with channels further from the visual area displaying progressively lower peak accuracy loss (Figure 3.11).

We observed that the temporal evolution of the sensor space maps showed the visual area sensors to be consistently the most important for the decoding objective across all time points. In theory, the sliding window LDA and the per-channel LDA approach could be combined to get a similar spatiotemporal profile, where each LDA model is trained on the sliding window of 4 channels at a time. However, in practice accuracy might suffer substantially with so few input features, and it would be computationally taxing considering the amount of LDA models required to train. Overall, PFI proved to be a useful technique for investigating full-epoch data and obtaining spatiotemporal information similar to what can be obtained from individual sliding window models.

3.3.6 Spectral PFI

Figure 3.12 presents our spectral PFI results averaged over subjects. This shows a clear peak of spectral information content at 4Hz, after which the power rapidly declines with increasing frequency. However, it should be noted that, because of the sampling rate of the data and the size of the epochs, the frequency resolution is only 2Hz. This means that the apparent 4Hz peak is due to the 1Hz highpass used for preprocessing the data, and so in actuality there is simply a 1/f characteristic, as is expected in MEG data. We have confirmed this by plotting the psd of the raw (bandpassed) data with a matched frequency resolution, and found the same peak at 4Hz, which shows that this is an artefact of the frequency resolution.

We present temporospectral PFI in Figure 3.13, which reveals temporal information content within individual frequency bands, in an alternative manner to using separate LDA models trained on wavelet features (Higgins et al., 2022b). All frequency values represent bands centred around the respective frequencies, except the 0Hz band which represents the true 0Hz signal, i.e. the average over the time window. For computing the STFT we followed the same setup as in Higgins et al. (2022b). Because we are using a 100ms window (10 timesteps) for computing the STFT the frequency resolution is 10Hz. When permuting a specific time window,

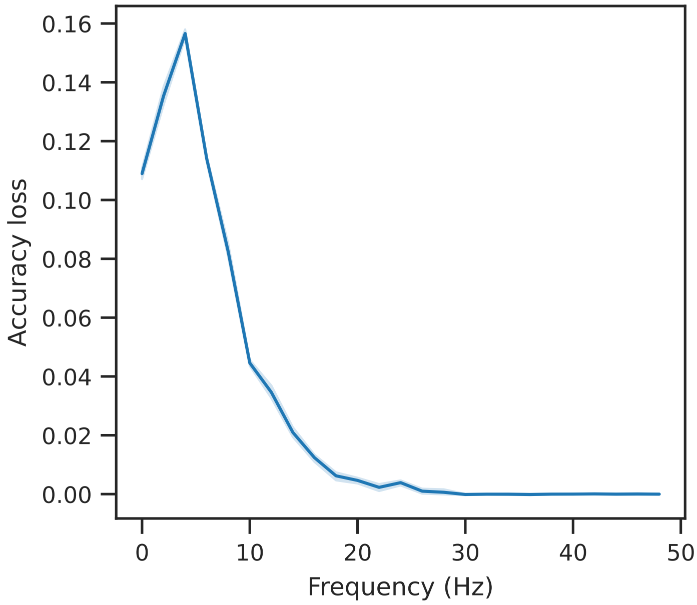


Figure 3.12: Spectral PFI of multiclass full-epoch LDA-NN on the 118-image dataset. Shading indicates 95% confidence interval across permutations. Results are averaged across subjects.

we also permuted the frequency content of the time window right before and after, to obtain a smoother temporal profile.

As expected from the standard temporal PFI, the temporal peak is between 100 and 150ms. Spectrally, higher frequency bands tend to be less and less useful to the decoding objective, confirming the observations of Higgins et al. (2022b). However, we think both the figure in Higgins et al. (2022b), and the temporospectral PFI analysis are slightly misleading, as they could be interpreted as having a peak in information content in the 10Hz band. As observed in Figure 3.12 this effect is explained simply by the $1/f$ characteristic. Because of the poor frequency resolution, both lower and higher frequencies are represented in the 10Hz band, thus all it shows is the $1/f$ characteristic, and the reason why it is higher than the "0Hz" band is because the 0Hz band contains solely the true 0Hz content. A potentially better approach to disentangling time-frequency information content

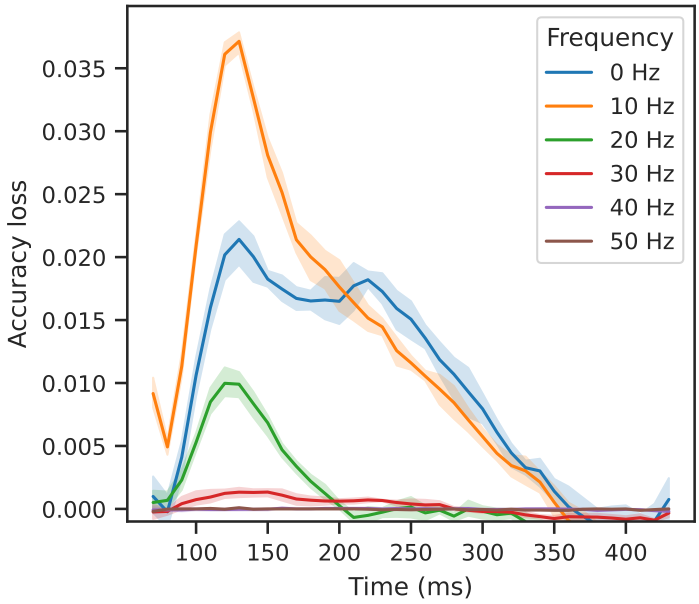


Figure 3.13: Temporospectral PFI of multiclass full-epoch LDA-NN on the 118-image dataset. Shading indicates 95% confidence interval across permutations. Results are averaged across subjects.

would be to bandpass the data first into specific frequency bands, then train our decoding model and compute the temporal PFI on each bandpassed data version.

We note that it is expected that we would find little to no signal above 25Hz, because of the lowpass filter we have employed. In later timepoints (>200ms) the 0Hz band seems to be slightly more important than the 10Hz band, potentially meaning that the classifier relies more on average rather than oscillatory activity after the visual peak. Similar to spatiotemporal PFI we can combine spatial and spectral PFI to assess the spectral information content of individual MEG channels (see Figure 3.14).

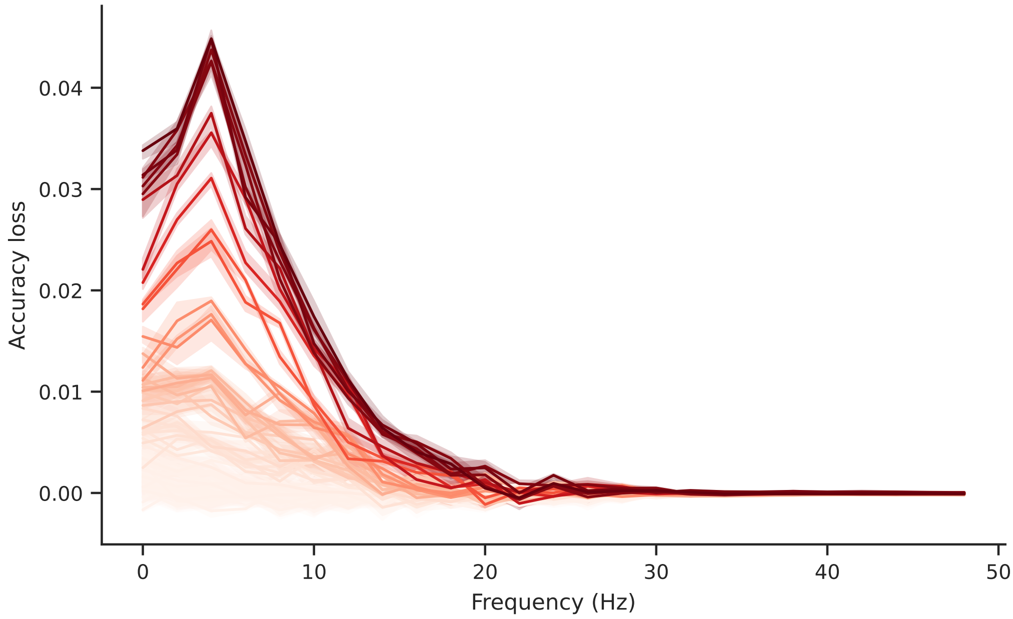


Figure 3.14: Spatiospectral PFI of multiclass full-epoch LDA-NN on the 118-image dataset, averaged over subjects. Blocks of 4-channel neighbourhoods are shuffled in each frequency to obtain the per-channel frequency profile. Each line corresponds to a sensor. The color map of the upper plot is based on the overall spatial PFI of each sensor, i.e. sensors with high spatial PFI accuracy loss are shown as darker red. The shading is across the permutations used for PFI and indicates the 95% confidence interval.

3.4 Discussion

We made the following contributions in this chapter. We showed empirically that full-epoch models achieve higher accuracy than sliding window decoding models. We showed how temporal, spatial, and spectral brain activity patterns related to stimulus discrimination can be extracted for any black-box full-epoch model. We have shown that learning the dimensionality reduction of input features in a supervised way, within a neural network optimised for the classification task, improves performance substantially. Next, we discuss each result in more detail.

We have found that training a single full-epoch model for multiclass decoding is effective in improving decoding performance. We have shown how this can be used while still providing neuroscientific insights by using PFI to learn which features are contributing to the decoding accuracy. Our results show that a full-epoch model generally performs better than individual sliding window models for visual decoding tasks, and the magnitude of this effect increases with the size of the dataset. The time-efficiency benefits of using a full-epoch model are significant, as training sliding window models takes roughly 10 times longer than a single full-epoch model for a 100 ms time window with a 100 Hz sampling rate.

Our analysis of different window sizes showed that while larger window sizes may improve performance, they are not effective in accurately capturing the temporal profile of information content. It has also been suggested that using equal-length time windows for all trials does not account for trial-by-trial variability, and Vidaurre et al. (2018c) proposed time-resolved decoding using a Hidden Markov Model to segment trials along the time dimension. This approach still involves training multiple models on multiple time windows. We, therefore, recommend using full-epoch models, as they only need to be trained once and contain information from all potentially useful time windows. After training any desired window size can be selected for temporal or spatial investigations through PFI, providing good decoding performance and dynamic spatiotemporal resolution without the need for retraining.

Both dataset size and trial length are important considerations in whether full-epoch decoding is beneficial over sliding-window decoding. Due to having to deal with more input features in full-epoch decoding overfitting becomes an issue with small datasets. Similarly if the experimental setup involves trials of e.g., multiple seconds this may lead to overfitting due to more input features. Thus, full-epoch decoding has considerable limitations in the case of small datasets and long trial windows. The length of the full-epoch window used for decoding should depend on the task at hand (i.e. what is the length of the time period which contains information) and the amount of data available.

There are additional considerations when adapting our methods to BCI settings. In real-time decoding it may be unacceptable to wait for 500ms before making a prediction to the user. In this case it is advisable to use smaller windows for good temporal resolution at the loss of classification performance. In asynchronous BCI paradigms it is possible to use larger windows (full epochs).

We also found that incorporating a supervised dimensionality reduction layer is essential for good decoding performance when using linear neural networks and LDA models. This can be used as a drop-in replacement over standard unsupervised dimensionality reduction typically done with PCA. One limitation of our supervised dimensionality reduction is the increased computation time of training a neural network versus computing a simple PCA. However, the neural network approach provides both the dimensionality reduction and the final decoding model end-to-end. If optimised well, running times are comparable to doing a PCA and then training an LDA model.

To further solidify our results we compared our approach with a Riemannian classifier on the 118-image dataset. We used the pyriemann library (Barachant et al., 2022), specifically *XdawnCovariances* for covariance computation followed by a tangent space projection. The LDA model was then applied to the features in the tangent space. The average validation accuracy over participants was 0.16, which is lower than the LDA-PCA approach. Both LDA-NN and the neural network are statistically significantly ($p < 1e-3$) better than this. We note that because of the high number of classes and channels, the dimensionality of the tangent space features was much higher than the features obtained from our neural network approach. We tried using the best possible settings for the Riemannian classifier, i.e.

specifying the number of filters (n_{filter}) to be 1, to reduce the number of features generated by *XdawnCovariances*. However, the main issue is the number of classes, since the number of output features is equal to $2 \cdot n_{classes} \cdot min(n_{channels}, n_{filter})$. In standard EEG-BCI applications, the number of classes is much lower than ours (118), which is why Riemannian classifiers may be better suited for EEG-BCI. To fully leverage the Riemannian classifier for this kind of MEG data, an additional feature reduction or selection step may be needed.

We compared PFI results from a full-epoch model with those from individual models trained on either separate time or spatial windows. This demonstrated that PFI can effectively extract both temporal and spatial information, and can also be used to investigate the interaction between these two dimensions. We also introduced a new technique whereby PFI can be used to extract spectral discriminatory information content and confirmed that this matches previous work training individual models on separate frequency band features. PFI is a particularly flexible technique, as it can be applied to nonlinear models and temporal or spatial resolution can be chosen post-hoc without the need for retraining. The performance of full-epoch nonlinear decoding and corresponding PFI analysis is explored in the next chapter. PFI can also be applied to individual conditions or single trials by rerunning with different permutations, enabling the investigation of various neuroscientific questions. Other methods for obtaining temporal and spatial information from trained models, such as the Haufe transform, are limited to linear models and do not provide trial-level patterns (Haufe et al., 2014). As opposed to the statistical nature of PFI, the Haufe transform directly maps the weights of a linear decoding model to input patterns, thus showing which parts of the input are the most important for the decoding objective. One limitatin of PFI compared to the Haufe transform is that the absence of influence on the output does not necessarily mean that those parts of the input (channels or time windows) do not contain information about the target.

To conclude, we recommend using a full-epoch multiclass model equipped with a supervised dimensionality reduction in order to achieve the best possible decoding performance while also allowing for flexibility in conducting neuroscientific investigations post-hoc such as MVPA or RSA. Our methods and recommendations scale well with data size and can be readily applied to deep learning models as well, thus bringing the applications of decoding to brain-computer interfaces and representational brain dynamics under a joint approach.

4 | Group-level decoding

In the previous chapter we have seen how we can improve decoding performance at the individual subject level, while providing useful neuroscientific insights. In this chapter we move on from within-subject variability and explore how nonlinear deep learning methods may be used to deal with between-subject variability. As we will see, while linear methods work well at the subject level, tackling group data requires more complex models.

Decoding is typically subject-specific and does not generalise well over subjects, due to high amounts of between subject variability. Techniques that overcome this will not only provide richer neuroscientific insights but also make it possible for group-level models to outperform subject-specific models. In this chapter, we propose a method that uses subject embedding, analogous to word embedding in Natural Language Processing, to learn and exploit the structure in between-subject variability as part of a decoding model, our adaptation of the WaveNet architecture for classification. We apply this to magnetoencephalography data, where 15 subjects viewed 118 different images, with 30 examples per image; to classify images using the entire 1s window following image presentation.

We show that the combination of deep learning and subject embedding is crucial to closing the performance gap between subject- and group-level decoding models. Importantly, group models outperform subject models on low-accuracy subjects (although slightly impair high-accuracy subjects) and can be helpful for initialising subject models. While we have not generally found group-level models to perform better than subject-level models, the performance of group modelling is expected to be even higher with bigger datasets. In order to provide physiological interpretation at the group level, we make use of permutation feature importance (PFI). This provides insights into the spatiotemporal and spectral information encoded in the models. We show that PFI works similarly well with nonlinear full-epoch models as with the linear models in the previous chapter.

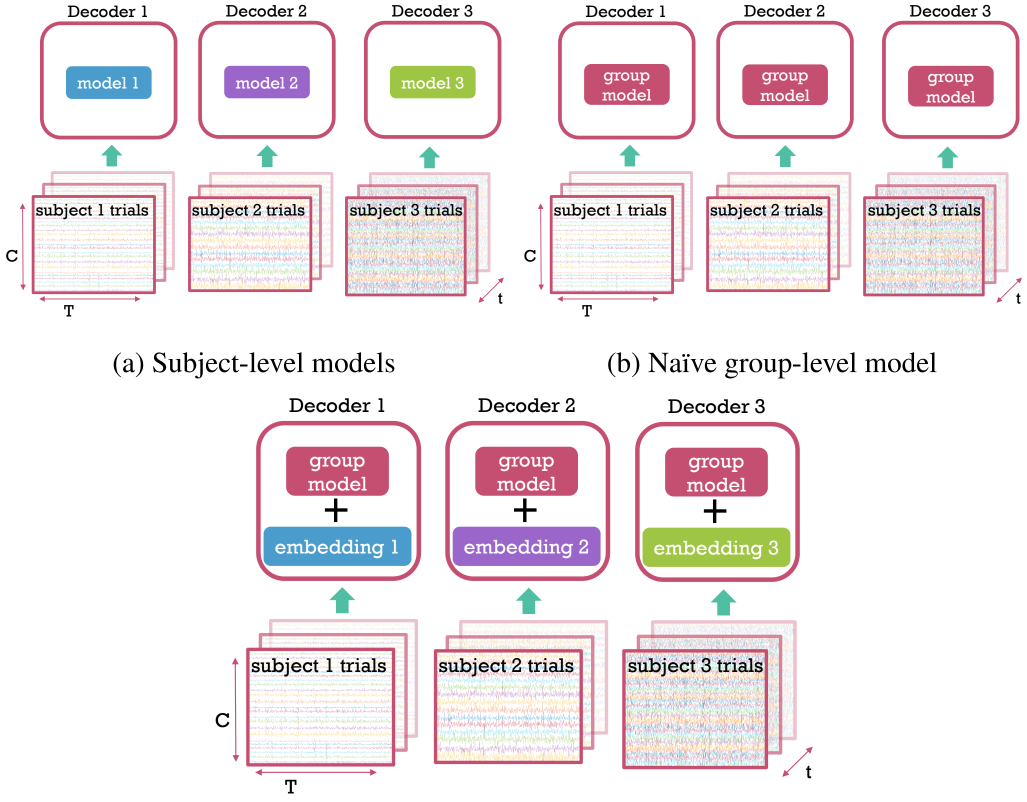
Note: Most of this chapter is part of a published paper (Csaky et al., 2023b). All of the work has been carried out by the thesis author. Most experiments in this chapter can be reproduced using the associated GitHub repository[1].

4.1 Introduction

Applications of decoding to brain recordings typically fit separate (often linear) models per dataset, per subject (Guggenmos et al., 2018; Dash et al., 2020a; Csaky et al., 2023a). This has the benefit that the decoding is tuned to the dataset/subject, but has the drawback that it is unable to leverage knowledge that could be transferred across datasets/subjects. This is especially desirable for the field of neuroimaging because gathering more data is expensive and often impossible (e.g. in clinical populations). More practical drawbacks of subject-specific (subject-level) models include increased computational load, a higher chance of overfitting, and the inability to adapt to new subjects. We aim to leverage data from multiple subjects and train a shared model that can generalise across subjects (group-level). A conceptual visualisation of subject-level and group-level models is given in Figure 4.1.

Subject-specific modelling is often preferred due to the high levels of between-subject variability in neuroimaging data. An alternative approach would be to train and use the same decoding model across multiple subjects (Olivetti et al., 2014; Li et al., 2021). We will refer to an approach that does this, while not explicitly modelling any of the between-subject variability, as “naïve group modelling”.

[1] https://github.com/ricsinaruto/MEG-group-decode



(a) Subject-level models

(b) Naïve group-level model

(c) Proposed group-level model

Figure 4.1: Comparison of subject-level (a), naive group-level (b), the proposed group-level (c) modelling. (a) A separate model is trained on the trials (examples) of each subject. (b) A single, shared model is trained on the trials of all subjects without capturing between-subject variability. (c) A single, shared model is trained on the trials of all subjects with an additional embedding component that is subject-specific. Each trial is $\mathbb{R}^{C \times T}$. Each of the S subjects has T trials.

Such naïve approaches, effectively pretend that all data comes from the same subject (see Figure 4.1b), but due to high amounts of between-subject variability typically perform very badly (Saha and Baumert, 2020; Olivetti et al., 2014; Li et al., 2021). The work in this paper is motivated by a need to improve on these methods. If group modelling could be advanced to account for the high amounts of between-subject variability, then this would allow relevant information to be pooled across subjects, resulting in two key benefits. First, we would be able to obtain neuroscientific insights from the decoding models directly at the group level instead of pooling over subjects. Second, with appropriately large multi-subject datasets, group models would be able to outperform subject-level models.

In this chapter we propose a general architecture capable of jointly decoding multiple subjects with the help of subject embeddings (Figure 4.1c and Figure 4.2). Our main aim is to improve subject-level models by using a single group decoding model that generalises across (and within) subjects. We refer to this as *across-subject* decoding, in which models are trained on part of the data from all subjects and then tested on left-out data from all subjects. This is motivated by the fact that group-level models that perform well in this manner can be useful for gaining neuroscientific insights that are relevant at the group level, as we will show in Sections 4.3.4 and 4.3.5. An alternative approach, *Leave-one-subject-out (LOSO) analysis* is also presented in Section 4.3.3. In LOSO analyses, group-level models are trained on data from multiple subjects and tested on a new, unseen subject (Zubarev et al., 2019), which can be especially useful in zero-shot BCI applications.

Recently, different transfer learning approaches have been proposed to deal with the problem of variability between subjects. Kostas and Rudzicz (2020) have proposed two distinct methods. First, there is Euclidean alignment, which is very similar to a spatial whitening of the data. We tried this in conjunction with our group model, and found it to lower performance, and thus opted for a simpler channel-wise standardisation. Second, there is mixup regularisation, which is entirely complementary to our approach and can be used in conjunction with it. It is a general regularisation/data augmentation technique and does not specifically deal with inter-subject variability.

Most transfer learning frameworks consist of applying a model trained on one subject to a different (target) subject (Elango et al., 2017; Dash et al., 2019; Cooney et al., 2019a; Olivetti et al., 2014; Halme and Parkkonen, 2018; Li et al., 2021). Some approaches use learnable affine transformations between subjects (Elango et al., 2017), while others finetune the whole model on target subjects (Cooney et al., 2019a; Dash et al., 2019).

Hyperalignment has been successful for fMRI data to align different subjects to a common cortical space, and some applications have been explored in MEG data as well. For example, Benz (2020) used hyperalignment on MEG data via procrustes matrix transformation to a common sensor space and showed improvement in evoked fields. Similar methods have been explored in recent studies aiming to deal with between-subject variability (Ravishankar et al., 2021; Michalke et al., 2023; Zhou et al., 2020). However, to our knowledge, no prior work has applied it successfully to MEG decoding.

One key consideration is that hyperalignment is a linear method, constraining the transformation between subjects. While this is a sensible assumption, we think that in order to fully leverage data from multiple subjects a nonlinear method is required. Our subject embedding method is fully data-driven without any constraints on the nature of variability between subjects. It may be that this flexibility becomes truly useful when dealing with a large number of subjects, and for a few subjects, the linear assumptions of hyperalignment could work better. Our method also directly optimises the subject embeddings for the decoding objective. It is not clear, whether an unsupervised method, such as hyperalignment would result in better decoding accuracy. We leave it for future work to provide a full comparison between hyperalignment and subject embedding for MEG decoding.

Transfer learning is also popular in the wider machine learning field. Parallels can be drawn with domain adaptation (Long et al., 2015), or transferring knowledge from large to small datasets within the same domain (Wang et al., 2019; Zhuang et al., 2020). Natural language processing (NLP) datasets often contain data from widely different sources (Radford et al., 2022), but due to the sheer size of the dataset and model complexity, training on joint data achieves good results (Brown et al., 2020a; Devlin et al., 2019b). Modern approaches to NLP often use Transformer-based (Vaswani et al., 2017) architectures. We believe convolutional architectures are an attractive approach for multi-channel timeseries data, and explore Transformers in the next chapter. There are several issues to overcome when applying Transformers to multi-channel timeseries, similar but perhaps even more challenging than their application to computer vision (Parmar et al., 2018; Dosovitskiy et al., 2020). In this work we wanted to keep the model relatively simple, as our dataset size is also limited, and analyse the effect of the subject embedding.

As discussed before, a naive concatenation of subjects does not work well on small neuroimaging datasets. Perhaps the most relevant parallels can be drawn with dialogue and speech modelling work, where inter-speaker differences are modelled using speaker embeddings (Li et al., 2016; Zhang et al., 2018; Saito et al., 2019; Mridha et al., 2021). Chehab et al. (2022) have similarly found that subject embeddings provide a small but significant improvement in encoding MEG data from a language task. They used a combination of recurrent and convolutional neural networks for encoding MEG data. However, limited information is provided on how subject embedding helps, and their results cannot be directly generalised to MEG *decoding*. Our results expand this work to the task of decoding images from MEG data and provide additional insight into how deep learning and subject embeddings help group-level decoding models. In concurrent work, Défossez et al. (2022) have also shown the effectiveness of subject embeddings in group-level speech decoding. They have also compared it to subject-specific layers as a way of dealing with between-subject performance and found this latter approach slightly better. The advantage of subject embeddings is that they use less parameters to deal with the between-subject variability and the structure in the learned representations can be readily interpretable.

We make the following contributions using a 15-subject MEG dataset with a visual task (Cichy et al., 2016). First, we introduce a group-level model with subject embeddings, substantially improving over naive group modelling and showing the potential improvements in decoding that can be provided over subject-specific decoding models. Second, we provide insight into how non-linearity and subject embedding helps group modelling. Third, we show that we can gain neuroscientific insights from the deep learning-based decoding model, using permutation feature importance (Altmann et al., 2010) to reveal how meaningful spatiotemporal and spectral information is encoded.

4.2 Methods

4.2.1 Data

In this work, a task-MEG dataset is used where 15 subjects view 118 different images, with each image viewed 30 times (Cichy et al., 2016). The raw epoched data is publicly available[2], however, we obtained the continuous raw MEG data directly from the authors to be able to run our preprocessing pipeline using MNE-Python (Gramfort et al., 2013). This is the same dataset as the 118-image dataset used in the previous chapter.

Raw data is bandpass filtered between 0.1 and 125 Hz and line noise is removed with notch filters. Whitening is used to remove covariance between channels for

[2] http://userpage.fu-berlin.de/rmcichy/fusion_project_page/main.html

subject-level models. Removal of cross-channel covariance (whitening), or in other words multivariate noise normalisation has been previously found to improve the performance of linear decoding models (Guggenmos et al., 2018). The whitening is simply done by performing a PCA projection over the channels keeping all components. For group-level models no whitening is performed, instead, each channel is individually standardised by removing the mean and dividing by the variance. The reason for not using whitening in the case of group-level models is that it would destroy the alignment of the channels between subjects, as each PCA decomposition projects into a different space. Alternatively, we can run PCA at the group-level on the data concatenated over subjects, however, we did not see an improvement in performance when doing this.

After whitening, we downsample to 250 Hz and 1.024-second epochs are extracted, starting 100 ms before stimulus presentation. This resulted in $\mathbb{R}^{C \times T}$ dimensional trials with $C = 306$ and $T = 256$. We do multiclass decoding, predicting a separate probability for each of the 118 classes (images). For a summary of the epoched data see Table 4.1.

Number of subjects	**Number of classes**	**Number of samples per class**	**Dimensions of one sample**
15	118	30	306 channels × 256 timesteps

Table 4.1: Dimensons of the epoched dataset.

4.2.2 Models

Our choice of core decoding model was based on a desire to assess the extent to which group decoding models might allow for the use of more complex, nonlinear networks when compared to subject-specific decoding. In addition, we did not aim to design a new kind of architecture for decoding MEG data, but rather build our model based on CNN-based architectures that have already been proven to be effective on time series data. As such, we used a decoding model based on WaveNet for classification, which has been used successfully in the audio domain (van den Oord et al., 2016; Zhang et al., 2020), and which we refer to as the Wavenet Classifier. An important aspect of adapting this model is the high dimensionality of MEG data compared to audio. Whereas audio consists of a single channel, MEG has over 300. We simply set the input channels of the first convolutional layer to this number and thus the model processes the channel dimension in a fully-connected way.

The dilated convolutions in WaveNet are effective for modelling time series data, as successive layers extract complementary frequency content of the input (Borovykh et al., 2018). While CNN-based architectures have been used successfully on M/EEG data (Lawhern et al., 2018), there is no prior work specifically applying Wavenet to neural decoding. To be clear when we refer to *model* in this section we mean the general (untrained) architecture and not a trained model on some dataset. For all training instances in this paper, we used a randomly initialised model instead of using the pretrained weights from the audio-WaveNet.

Our Wavenet Classifier model consists of 2 parts: the (temporal) convolutional block, intended to act as a feature extractor; and the fully-connected block, which is designed for classification (Figure 4.2). The convolutional block uses a stack of 1D dilated convolutional layers, which include dropout and the inverse hyperbolic sine activation function (asinh). One layer is defined as

$$\mathbf{H}^{(l+1)} = \operatorname{asinh}\left(\operatorname{Dropout}\left(\operatorname{DilatedConv}^{(l)}\left(\mathbf{H}^{(l)}\right)\right)\right) \tag{4.1}$$

where $\mathbf{H}^{(l)} \in \mathbb{R}^{M^{(l)} \times T^{(l)}}$ is the input representation at layer l with $M^{(l)}$ channels and $T^{(l)}$ time points, and $\mathbf{H}^{(l+1)}$ is the output representation. $\mathrm{DilatedConv}^{(l)}$ is the dilated convolutional operation at layer l. For a single channel it is defined as:

$$y[n] = \sum_{k=0}^{K-1} x[n - d \cdot k] \cdot w[k] \tag{4.2}$$

where $x[n]$ and $y[n]$ is the input and output at time n, $w[k]$ is the kernel weight at index k, and d is the dilation factor. The dilation factor is doubled in successive layers, i.e., $d \in 1, 2, 4, \ldots$. In our case the dilated convolution kernel always has 2 learnable values ($K = 2$). This, together with the increasing dilations allows for rapid increase of the receptive field with few parameters and layers. In terms of number of channels we have an initial convolutional layer with a kernel size of 1 which simply projects the input channels C to the channel dimensionality used throughout the rest of the convolutional layers: $M^{(l)} = 2 * C, \forall l \in L$.

For subject-level modelling, we use 3 convolutional layers. For group-level modelling, we use 6 convolutional layers. We arrived at these numbers empirically by training both 3-layer and 6-layer subject-level and group-level models and selecting the best model version in each case, thus providing a fair comparison between subject- and group-level settings. Given there is no pooling and a convolution stride of 1, the output of each layer preserves the temporal dimensionality, except the amount that gets chopped off because of the kernel size itself. When using a convolutional layer with a kernel size K, the output representation's temporal dimension is

$$T^{(l+1)} = T^{(l)} - K - d + 2 \tag{4.3}$$

Since the dilation factor is doubled in successive layers, the receptive field of the

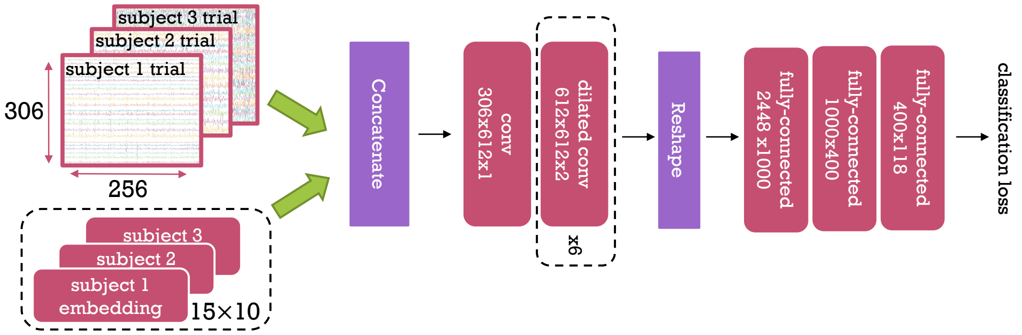


Figure 4.2: Group-level WaveNet Classifier with subject embeddings. Dashed boxes represent parts of the model which differ between subject-level and group-level versions of our architecture. Red boxes represent learnable parameters. For convolutional layers, the numbers represent *input channels* x *output channels* × *kernel size*. For fully-connected layers, the numbers represent *input neurons* × *output neurons*. The embedding layer dimensionality is given as $S \times E$, where $S = 15$ is the number of subjects, and $E = 10$ is the embedding size. Embeddings are concatenated with input trials to provide information about which trial is coming from which subject. The classification loss is cross-entropy.

convolutional block is 2^L where L is the number of layers. At the end of the convolutional block, we downsample temporally by the size of the receptive field. For example, the a convolutional block with 6 layers has a receptive field of 64 and thus its output representation has a length of $T - 64 + 1 = 193$, where $T = 256$ is the input trial's length. This is downsampled by a factor of 64, resulting in 4 values per channel.

Next, this downsampled output is flattened and fed into a fully-connected block. The final output is a logit vector corresponding to the 118 classes. The model is trained with the cross-entropy loss for classification, which includes a softmax function that maps the logit vector to a probability distribution over classes.

We assess two versions of each model, one with a Wavenet Classifier that is linear and one that is nonlinear. This allows us to see how nonlinearities (a bedrock of deep learning) interact with group modelling. The linear versions simply correspond to Wavenet Classifier where the activation function is set to be the identity function.

Finally, we divide the group-level modelling into two approaches. First, we have a naive group model, which is our standard 6-layer Wavenet Classifier. Second, we have our proposed group model, which improves on the naive group model through the inclusion of subject embeddings. A high-level mathematical description of subject-level (Equation 4.4), naïve group-level (Equation 4.5), and the embedding-aided group-level (Equation 4.6) models is given below (corresponding to the 3 panels in Figure 4.2).

$$\forall s \in S : \mathbf{y}_s = f_s(\mathbf{X}_s) \tag{4.4}$$

$$\forall s \in S : \mathbf{y}_s = f_g(\mathbf{X}_s) \tag{4.5}$$

$$\forall s \in S : \mathbf{y}_s = f_g(\mathbf{X}_s, \mathbf{e}_s) \tag{4.6}$$

Where s denotes a single subject and S is the set of all subjects. $\mathbf{y}_s$ and $\mathbf{X}_s$ are the target variables and input trials of subject s, f_s is the subject-specific model, and f_g is the shared group-level model across subjects. $\mathbf{e}_s$ is the subject-specific learned embedding.

Subject embeddings are introduced as a way of dealing with between-subject variability, similarly to Chehab et al. (2022). Like word embeddings in NLP, each subject has a corresponding dense vector (Mikolov et al., 2013b). This same vector is concatenated with the channel dimension of the input trial across all time points (in each trial). This operation is given in programming notation below.

$$\mathbf{H}_s = concatenate((\mathbf{X}_s, \mathbf{E}_s), dim = 0) \tag{4.7}$$

Where $\mathbf{X}_s \in R^{C \times T}$ is the input trial consisting of C channels and T time points, $\mathbf{E}_s \in R^{E \times T}$ is the subject embedding of size E repeated across the T timepoints. $\mathbf{H}_s \in R^{(C+E) \times T}$ is the input that gets fed into the model f_g. Embedding size was set to 10 a priori, and the effect of different values is explored in Section 4.3.2. Subject embeddings are learnt together with other model weights using backpropagation. We reasoned that an embedding-aided model can learn general features across subjects, with the capability of adapting its internal representations for each subject.

4.2.3 Model analysis

In this section, we describe several approaches to uncovering the information encoded in the WaveNet Classifier. In the previous chapter we have validated the use of PFI in MEG decoding against more traditional methods, such as sliding-window decoding. We leverage PFI here due to its flexibility in obtaining spatiotemporal information from trained models, and its applicability to nonlinear models. The application of temporal and spatial PFI to the Wavenet Classifier follows the same methods described in the previous chapter.

We also extended the PFI method to individual kernels of the Wavenet model. In this case, the feature importance measure is the absolute difference between the kernel output using the original and permuted inputs. We reason that a more important feature will cause a higher output deviation. The model receives the same permuted inputs as in model-level PFI, the difference is that we look at the output of individual kernels instead of the whole model. Specifically, for a kernel in layer l, applied to input channel i, contributing to output channel o, the feature importance Δp_j is

$$\Delta p_j = \mathbb{E}_{\mathbf{X}} \left[f(\mathbf{X}; \theta)_{l,i,o} - f(\mathbf{X}_{\perp j}; \theta)_{l,i,o} \right] \tag{4.8}$$

where f is the trained model with parameters θ. $\mathbf{X}_{\perp j}$ denotes shuffling of feature j in $\mathbf{X}$. This feature can be either temporal, spatial, spectral, or joint across multiple dimensions as described in the previous chapter. For group models we add the learned subject embedding to $\mathbf{X}$ as usual.

4.2.4 Experimental details

Our main evaluation metric is the classification accuracy of the across-subject decoding across the 118 classes. Recall that in across-subject decoding, each subject has a train and test split, and the aim is to see if a single group decoding model generalises across (and within) subjects. Train and validation splits with a 4:1 ratio were constructed for each subject and class. This means that classes are balanced (i.e., contain the same number of examples) across subjects and splits. Subject-level and group-level models are trained and evaluated on the same splits. Note that for each model, an extra training is conducted wherein the (linear) identity function is used as an activation function to assess the influence of nonlinearity. Linear and non-linear models are trained for 500 and 2000 epochs (full passes of the training data), respectively, with the Adam optimiser (Kingma and Ba, 2015). Table 4.2 lists all of the model and training combinations that are presented in Figure 4.3. In this section when we refer to *model* we mean specific trained models on the respective datasets given in Table 4.2.

Dropout was set to 0.4 and 0.7, and a batch size of 590 and 59 was used for group-level and subject-level models, respectively. The learning rate was set to 0.0001 for group-level, and 0.00005 for subject-level models. Training of a single subject-level and group-level model took 5-15 minutes and 4 hours on an NVIDIA A100 GPU, respectively. For linear models, validation losses (cross-entropy) and accuracies were negatively correlated, i.e. loss decreases while accuracy increases, and eventually both suggested overfitting. Since non-linear models

Method name	**Linear/ non-linear**	**No. of conv layers**	**Subject embed-dings**	**Trained on *subject* or *group* data**	**Finetuned on *subject* data**
Linear subject	linear	3	no	subject	no
Nonlinear subject	nonlinear	3	no	subject	no
Linear group	linear	6	no	group	no
Nonlinear group	nonlinear	6	no	group	no
Linear group-emb	linear	6	yes	group	no
Nonlinear group-emb	nonlinear	6	yes	group	no
Nonlinear group finetuned	nonlinear	6	no	group	yes
Nonlinear group-emb finetuned	nonlinear	6	yes	group	yes

Table 4.2: Model and training combinations and their corresponding naming

are more expressive, they overfitted sooner according to the loss, but accuracy kept improving until it reached a plateau, never overfitting. Analysing the loss distribution across validation examples (for non-linear models) shows that even during overfitting most examples' loss keeps decreasing with a few high-loss outliers disproportionately influencing the mean. Since accuracy is binary, outliers are diminished, explaining the apparent difference in learning behaviour. For linear models, this unintuitive behaviour was not observed probably due to inherent model simplicity.

We compute Wilcoxon signed-rank tests for comparisons of interest over trained models, where the pairing is within-subject, and samples are the subject-level mean accuracies over validation trials. We used PyTorch for training (Paszke et al., 2019) and several other packages for analysis and visualisation (Pedregosa et al., 2011; Virtanen et al., 2020; Harris et al., 2020; Wes McKinney, 2010; Waskom, 2021; Hunter, 2007).

4.3 Results

4.3.1 Subject embedding aided group models

Validation accuracies for all trained models are shown in Figure 3. Interestingly, at the subject level, linear models performed slightly better than non-linear (4% increase, $p = 5.7e - 4$). We think that both the limit in data size and noise levels in the data contribute to the subpar performance of non-linear models when trained/validated within-subject. The large between-subject variability common to MEG datasets is apparent, with individual subjects' accuracy ranging from 5% to 88%. As expected, training naive group models, i.e. a naive application of either the linear or non-linear WaveNet Classifier to the group modelling problem (orange violin plots), results in much worse performance than training subject-level models, i.e. 30% decrease compared to `linear subject`. Inferring such high variability implicitly between so few subjects is not trivial.

Adding subject embeddings to the non-linear model (`non-linear group-emb`) improves performance by 24% ($p = 1.9e - 6$), with no increase for the linear model (`linear group-emb`). This shows that leveraging subject embeddings in conjunction with non-linear activations can narrow the gap with subject-level models (6% difference with `linear subject`, $p = 1.3e - 2$). Limiting the non-linearity to the first layer resulted in a subpar performance, similar to that of a linear model. This indicates that non-linearity is needed within multiple layers to benefit from subject embeddings. The impact of subject embeddings is further investigated in Section 4.3.2.

We also finetuned the embedding-aided group-level model on the training data of each subject separately (`non-linear group-emb finetuned`) for 500 epochs. We effectively use the group-level model as an initialisation for subject-

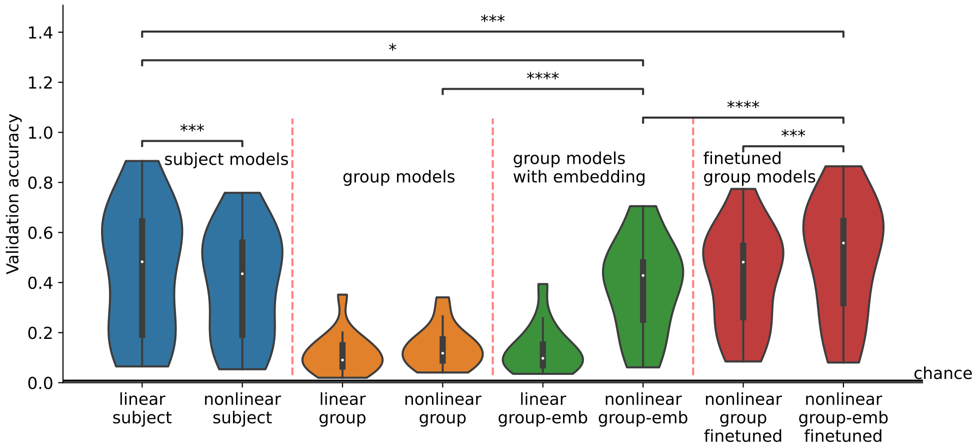
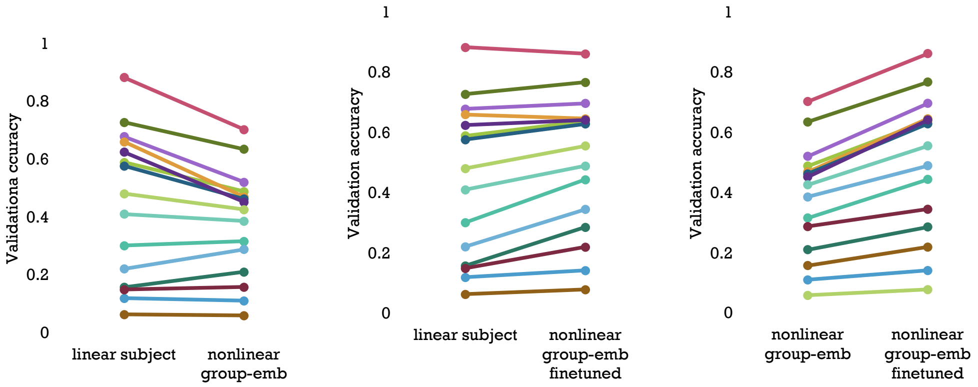


Figure 4.3: Trained subject-level and group-level models evaluated on the validation set of each subject. Wilcoxon signed-rank tests are shown for comparisons of interest ($* = p < 5e-2, ** = p < 1e-2, *** = p < 1e-3, **** = p < 1e-4$). The `non-linear group-emb finetuned` model is finetuned separately on each subject, initialized with the `non-linear group-emb` model. Chance level is $1/118$.

level models, improving over subject-level models trained from scratch (`linear subject`), achieving 50% accuracy (5% increase, $p = 1e-3$). This shows that representations learned at the group level are useful for subject-level modelling. In contrast, finetuning a naive group model (`non-linear group finetuned`) only achieved 42% accuracy showing that the best performance is reached when finetuning is combined with the best group-model. Thus, in addition to closing the gap between subject-level and group-level modelling, finetuning our embedding-aided model provides the best overall accuracy for subject-level modelling. The variance of `non-linear group-emb` (19%) and `non-linear group-emb finetuned` (24%) is lower than `linear subject` (26%). Generally, group models reduce between-subject variability.

In neural decoding, group models are widely understood to perform worse than individual models (Guggenmos et al., 2018; Dash et al., 2020a) But why is this?



(a) Subject- to group-level (b) Subject to group finetuned (c) Group to group finetuned

Figure 4.4: Accuracy changes across all 15 subjects (individual colours), when comparing trained `linear subject`, `non-linear group-emb`, and `non-linear group-emb finetuned` models. Both `non-linear group-emb` and the finetuned version clearly reduce the variability of accuracies across subjects and are especially helpful for low-accuracy subjects. When finetuning `non-linear group-emb` on individual subjects (c), we can see that accuracy increases for all subjects, and especially for high-accuracy subjects. This is unsurprising because these subjects have good enough data on their own for subject-level models to be able to learn well. As seen in (a) and (b) these high-accuracy subjects are usually impaired by group-level models, exactly for the aforementioned reason.

By plotting per-subject performance in both kinds of models (Figure 4.4), we see something revealing. In the case of `non-linear group-emb`, 4 subjects with generally low accuracies (15-30%) had higher accuracies than `linear subject` (even though the mean across subjects is lower). This suggests that group models could be successfully used for some subjects if those subjects could be identified. Compared to a subject-level model that might overfit to the noisy data of a bad subject, the group model might possess inherent regularisation due to having been trained on multiple subjects, and thus improve performance on bad subjects. As shown this is achieved at the expense of decreasing the accuracy of good subjects.

Strong negative correlations of -0.88 and -0.54 are obtained between `linear subject` subject-level accuracies and the change in accuracy achieved by the `non-linear group-emb` and `non-linear group-emb finetuned` models, respectively. Comparing finetuning to from-scratch subject-level models (`linear subject`), only 2 high-accuracy subjects are slightly worse, and generally low/mid-accuracy subjects show more improvement than high-accuracy subjects (Figure 4.4).

We analysed our main findings on another publicly available visual MEG dataset with 92 different images (15 subjects, and 30 trials per image) (Cichy et al., 2014). This dataset is the same as the 92-image dataset used in the previous chapter. Linear subject-level models achieved 35% accuracy, whereas a linear group model without embeddings had 12%, and a nonlinear group model with embeddings had 28%. Thus we can see that our approach behaves similarly on this dataset, improving a lot over the naive group baseline, but not quite achieving the performance of the linear subject-level models. Finetuning the group model separately on individual subjects achieved 38% accuracy surpassing from-scratch subject-level models.

In summary, these results suggest the following recommendations for decoding MEG task data.

1. Subject embeddings and non-linearity should be used for achieving good group models.
2. Group-level models can be used to improve over subject-level models on low-performance subjects.
3. For the best subject-level performance, the finetuning approach should be used, benefitting low-performance subjects the most.

4.3.2 Insights into the embedding-aided group model

For the embedding-aided group-level setup (`non-linear group-emb`), 4 further models were trained for 5-fold cross-validation. The training and validation sets still contained 80% and 20% of trials respectively, and the splitting was done so that all of the data appears exactly once in the validation set across the 5 folds, for each subject and class. Average accuracy was 37.4% (as opposed to the 38% reported in Figure 4.3), with a 95% confidence interval of 0.8%. Thus, the proposed group-level model is robust to different random seeds and dataset partitions. More extensive robustness analysis is omitted due to computational constraints.

In non-linear subject-level models (`non-linear subject`), accuracy improves as we use fewer convolutional layers, whereas for non-linear group-level models (`non-linear group-emb`) using more layers improved accuracy (see Table 4.3). Thus, subject-level models seem to rely more on the fully-connected block as they are unable to extract good features, and group-level models rely more on the convolutional block to learn shared features across subjects. To have a fair comparison between the two approaches we selected the best number of layers for both individual and group-level models. To be clear, because of how we perform the temporal downsampling after the convolutional layers (described in Section 4.3.1), using fewer convolutional layers increases the overall parameter count because the fully-connected block has to be enlarged. Thus, the group model (with 6 conv layers), is about 2.5x smaller than the subject-level models (with 3 conv layers). However, `non-linear group-emb finetuned` models achieve higher accuracy than from-scratch subject-level models `linear subject`. This shows that, when they are initialised well (with a group model trained on multiple subjects), even subject-level models can benefit from non-linearity and more convolutional layers.

	linear subject	nonlinear subject	nonlinear group-emb
3 conv layers	0.45	0.39	0.22
6 conv layers	0.41	0.25	0.38

Table 4.3: Effect of number of convolutional layers on the validation accuracy of two subject-level and one group-level model.

We tried different approaches to understand how subject embeddings help the `nonlinear group-emb` model. A clustering or 2D projection of the embedding space such as PCA or t-SNE (Van der Maaten and Hinton, 2008) did not show any clusters (see Figure 4.5). This is likely to be a consequence of only having 15 subjects since cases where such visualisations work well (Liu et al., 2017) typically have thousands of dimensions (e.g. words in word-embeddings). To assess whether the embeddings simply encode which subjects are good, we transformed the embeddings with PCA and correlated all components with the accuracies across subjects. We found no significant correlations; therefore, embeddings do not appear to encode information about subject-level accuracy. To assess how much embeddings contribute to a trained model, we tried both setting the embeddings to zero and shuffling them. The validation accuracy of `nonlinear group-emb` decreased to 10% for both approaches. This is a 28% reduction from the original accuracy. Thus, embeddings encode crucial information to aid decoding, but the `nonlinear group-emb` is still better than chance without them.

To gain further insight into the learned subject embeddings we computed accuracy on each subject's validation data using other subjects' embeddings. In the resulting subject-by-subject confusion matrix the value in the i-th row and j-th column shows how well the embedding of subject i can be replaced with the embedding of subject j (Figure 4.6). After division with the original accuracies, the metric shows how much accuracy can be retained when swapping subject embeddings.

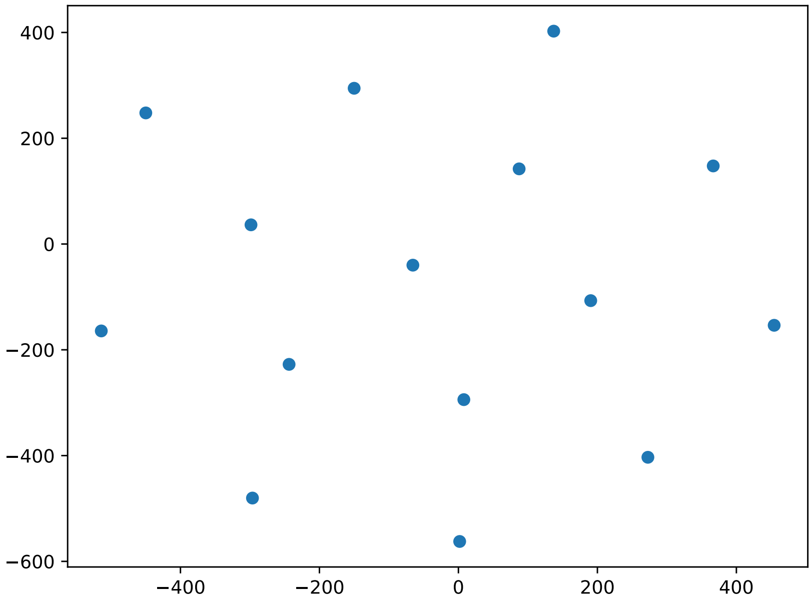
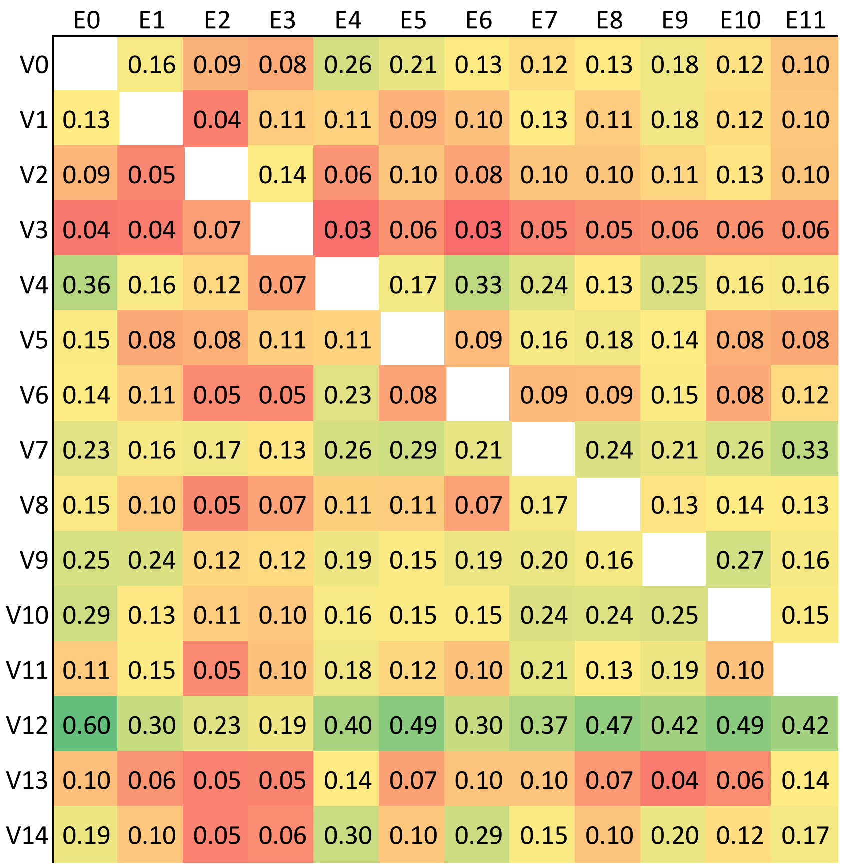


Figure 4.5: 2D t-SNE projection of the subject embeddings in the trained `non-linear group-emb` model.

Some subjects' embedding cannot be replaced by others (e.g. subject 3), and some subjects' embedding can be more easily replaced (e.g. subject 12). Conversely, some subjects' embeddings are more general as they can replace many others (e.g. subject 14), and some are less general (e.g. subject 2). We tried clustering this matrix, and looked at correlation with both embedding distance and subject accuracy, however no meaningful results were found.

Training with an embedding dimensionality of 3 and 14, resulted in 20% and 38% accuracy, respectively. We tried these two settings to see how embedding size in the lower and upper limits influences performance. As an embedding dimensionality of 14 performs the same as 10, we could draw the conclusion that 10 is not a limiting factor. From the much worse result with an embedding dimensionality of 3, we could draw the conclusion that compressing the embedding representations too much is not possible. As with the clustering analysis, this is likely to be due to



	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11
V0		0.16	0.09	0.08	0.26	0.21	0.13	0.12	0.13	0.18	0.12	0.10
V1	0.13		0.04	0.11	0.11	0.09	0.10	0.13	0.11	0.18	0.12	0.10
V2	0.09	0.05		0.14	0.06	0.10	0.08	0.10	0.10	0.11	0.13	0.10
V3	0.04	0.04	0.07		0.03	0.06	0.03	0.05	0.05	0.06	0.06	0.06
V4	0.36	0.16	0.12	0.07		0.17	0.33	0.24	0.13	0.25	0.16	0.16
V5	0.15	0.08	0.08	0.11	0.11		0.09	0.16	0.18	0.14	0.08	0.08
V6	0.14	0.11	0.05	0.05	0.23	0.08		0.09	0.09	0.15	0.08	0.12
V7	0.23	0.16	0.17	0.13	0.26	0.29	0.21		0.24	0.21	0.26	0.33
V8	0.15	0.10	0.05	0.07	0.11	0.11	0.07	0.17		0.13	0.14	0.13
V9	0.25	0.24	0.12	0.12	0.19	0.15	0.19	0.20	0.16		0.27	0.16
V10	0.29	0.13	0.11	0.10	0.16	0.15	0.15	0.24	0.24	0.25		0.15
V11	0.11	0.15	0.05	0.10	0.18	0.12	0.10	0.21	0.13	0.19	0.10	
V12	0.60	0.30	0.23	0.19	0.40	0.49	0.30	0.37	0.47	0.42	0.49	0.42
V13	0.10	0.06	0.05	0.05	0.14	0.07	0.10	0.10	0.07	0.04	0.06	0.14
V14	0.19	0.10	0.05	0.06	0.30	0.10	0.29	0.15	0.10	0.20	0.12	0.17

Figure 4.6: Subject embedding confusion matrix from the trained `non-linear group-emb` model. Columns (E0-E14) refer to subject embedding indices and rows (V0-V14) refer to subject validation sets. Greener shading (higher values) shows subjects with higher retained accuracy when their embeddings are swapped.

having few subjects.

4.3.3 Leave-one-subject-out evaluation

To this point, we have reported results for across-subject decoding, in which we use a single group decoding model that generalises across (and within) subjects; an approach that is, for example, relevant when one wants to gain neuroscientific insights that generalise to the group level. In this section, we report leave-one-subject-out (LOSO) cross-validation results; which is relevant, for example, when one wants to develop BCI methods that work on previously unseen subjects. Movement classification is one such application where it would be beneficial to be able to use a decoder trained on other subjects in a zero-shot setting. We also analyse how performance improves when we allow models to use increasing amounts of data (finetuning) from the left-out subject. We compare the LOSO and finetuning performance of `nonlinear group`, `nonlinear group-emb`, and `linear subject`. The `linear subject` approach serves as a baseline and it is only trained on the left-out subject. Thus, in the LOSO (zero-shot) setting, this model has chance level, since no training is performed on the left-out subject.

When training `nonlinear group-emb` the left-out subject's embedding was initialised randomly. In the LOSO (zero-shot) evaluation, both group models achieve 5% accuracy (Figure 4.7). Up to the case when 70% of the training data of the left-out subject is used, both group models are much better than `linear subject` ($p < 0.05$, corrected for multiple comparisons). This is expected and the benefit of group-level models in LOSO analysis has been previously established (Elango et al., 2017). Thus, to achieve the same level of performance as linear subject much less data is needed when finetuning a group model. Unsurprisingly, the `nonlinear group-emb` model does not improve over the naive model (`nonlinear group`), but is, importantly, not worse. As opposed to the finetuning setup in Figure 4.3, when adapting to new subjects, better group performance does not translate to better finetuning performance. We think this is because when adapting to a new subject, that subject's embedding was randomly initialised, and thus it has to be learned during the finetuning. This is a limitation of our approach.

To see what effect increasing the number of subjects has on group model performance we trained 15 embedding-aided sub-group models with increasing number

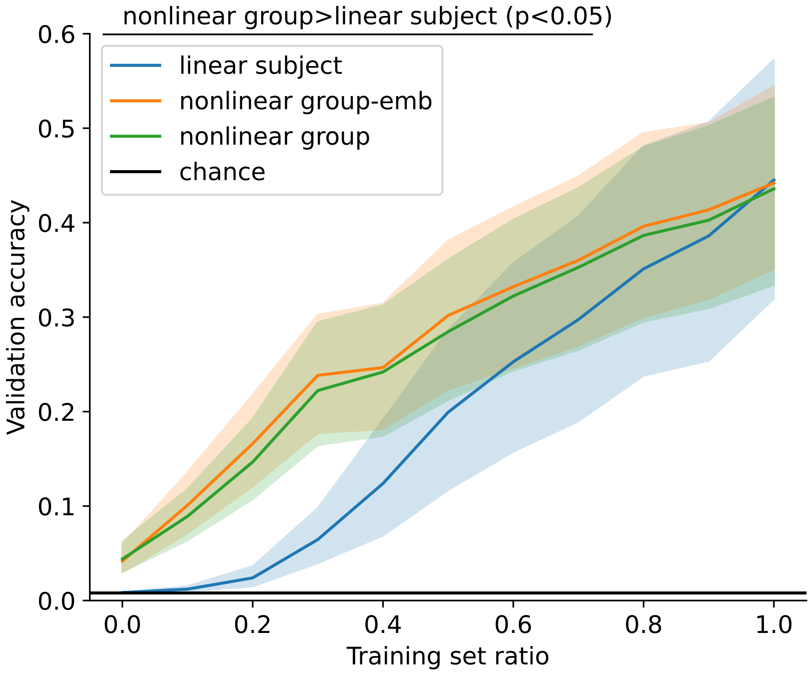


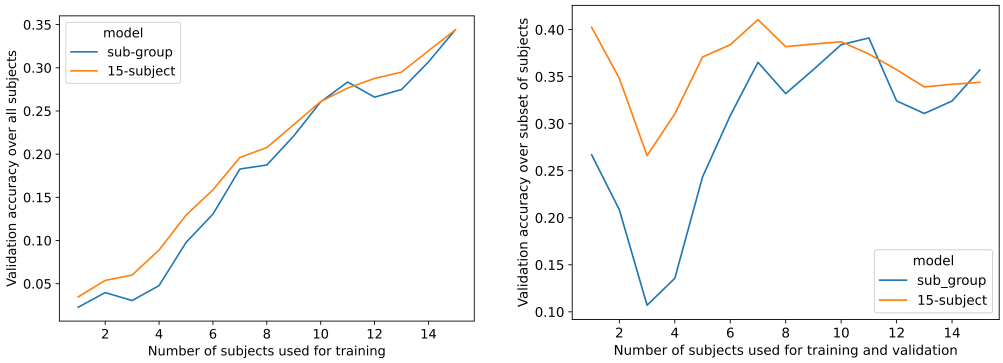
Figure 4.7: Generalisation and finetuning on left-out subjects. The horizontal axis shows the amount of training data used from the left-out subject; a training set ratio of 0 corresponds to a zero-shot approach. `Linear subject` is trained from scratch, while `nonlinear group-emb` and `nonlinear group` are initialised with the trained non-linear group-level model with and without embeddings, respectively. The 95% confidence interval of the accuracy across left-out subjects is shown with shading.

of subjects, i.e. 1 subject, 2 subjects, . . . , 15 subjects. We used the same hyperparameters as for our original `non-linear group-emb`. We then evaluated each sub-group model on the validation set of the subjects it was trained on (Figure 4.8b). The resulting validation accuracy is shown in the plot as a function of the number of subjects in the sub-group. One downside of this approach is that the order in which we add the subjects to the sub-group models matters a lot because of the high between-subject variability. However, to test multiple orderings we would have to run hundreds of trainings which is not possible under our computational constraints. Nevertheless, we compared our increasing subject-number sub-group models with the theoretical best performance achieved by the group model trained on all subjects (15-subject). We can see that the gap between the full group model and the restricted sub-group models generally tightens as we increase the number of subjects used for training. It is difficult to draw strong conclusions without repeating this analysis with different permutations of subjects.

An alternative visualisation for the previous analysis is to keep the validation set fixed, i.e. always compute validation performance on the validation set of all subjects (Figure 4.8a). To provide the theoretical maximum from the 15-subject group model we took its performance on the respective subjects (e.g. in the case of 2 subjects, the first 2 subjects), and replaced the other subjects with a 1/118 (chance) accuracy value. This again shows a slight tightening between the full group model and the restricted sub-group models as we increase the number of subjects. Notably, there is a dip in performance when we add subject 12 to the group model as this subject had a particularly bad performance.

4.3.4 Model-level PFI

An established critique of deep learning models applied to neuroimaging data is the lack of interpretable, neuroscientific insight they provide about the underlying neural processes that drive the decoding (Murdoch et al., 2019). To gain such insights, it is useful to assess the time- and space-resolved information/discriminability within trials. Figure 4.9 shows the temporal and spatial PFI of the trained `nonlinear group-emb` model. To make the results robust and smooth, the shuffling for temporal PFI was applied to 100 ms windows, and magnetometers and gradiometers in the same location were shuffled together for spatial PFI. Time windows or channels with higher accuracy loss than others are interpreted as containing more information about the neural discriminability of the visual images. This indicates when and where information processing related to the presented images



(a) Evaluation over all subjects

(b) Evaluation over subsets of subjects

Figure 4.8: (a) Validation accuracy over all subjects with respect to increasing the subset of subjects used for training the sub-group model (blue line) on the horizontal axis. The 15-subject model (orange line) is our standard `non-linear group-emb` model trained on all subjects. (b) Validation accuracy over the subset of subjects used for training the sub-group model (blue line). The 15-subject model (orange line) is our standard `non-linear group-emb` model trained on all subjects. The 15-subject model is evaluated on the same increasing sets of subjects as used for the sub-group models.

is happening in the brain.

Temporal PFI shows a large peak around 150 ms which is in line with previous subject-level PFI results on this dataset (see Chapter 3. After this, the information content rapidly decreases, with a second, smaller peak around 650 ms, which could correspond to a brain response following the end of image presentation at 500 ms. Spatial PFI shows that the most important channels are in the back of the head in the sensors in visual areas, as expected for a visual task.

We found good agreement between the PFI analysis and the alternative approach of a gradient-based analysis often used in deep learning models (Figure 4.10). In this analysis a salience map is obtained by backpropagating to randomly initialised inputs. We smoothed the temporal profile with the same window size as for the PFI analysis. Temporally we can see that the agreement between the two methods

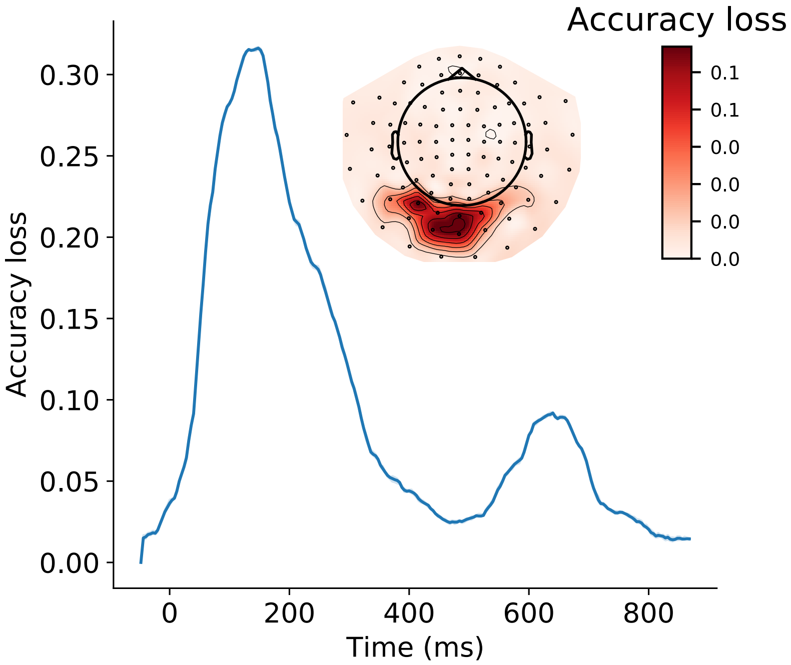
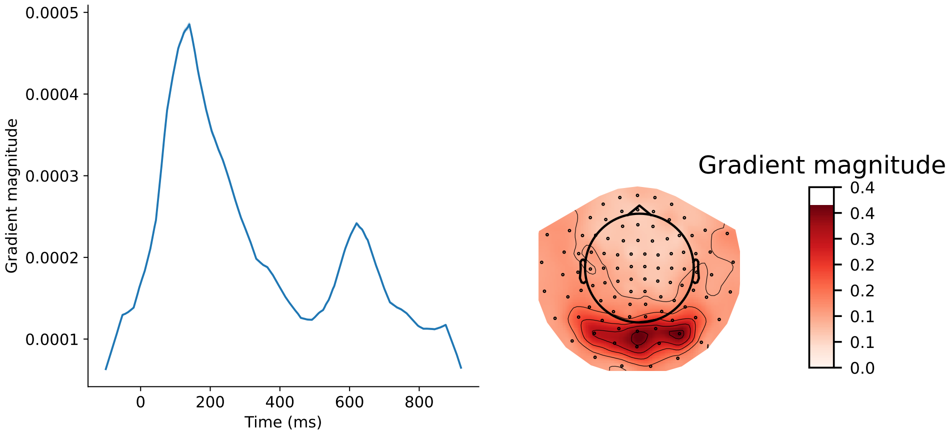


Figure 4.9: Temporal (line) and spatial (sensor space map) PFI for the trained `non-linear group-emb` model. For temporal PFI accuracy loss (vertical axis) is plotted with respect to time since visual image presentation (horizontal axis). Shading shows the 95% confidence interval which is not visible due to low variability. For spatial PFI, darker red shading is equivalent to higher accuracy loss.

is high, with peaks aligning very well (less than 10ms difference). Spatially the two methods do show some differences, but overall gradient analysis still points to the most important information being in the visual cortex. For a full explanation of this method please see Section 2.5.5.

4.3.5 Kernel analysis

To provide further insight into our trained `non-linear group-emb` model, we next show that interpretable spatial, temporal, and spectral information can be obtained by analysing the learnt weights. This analysis becomes possible because we use a multi-layered neural network, and there is no equivalent analysis that we



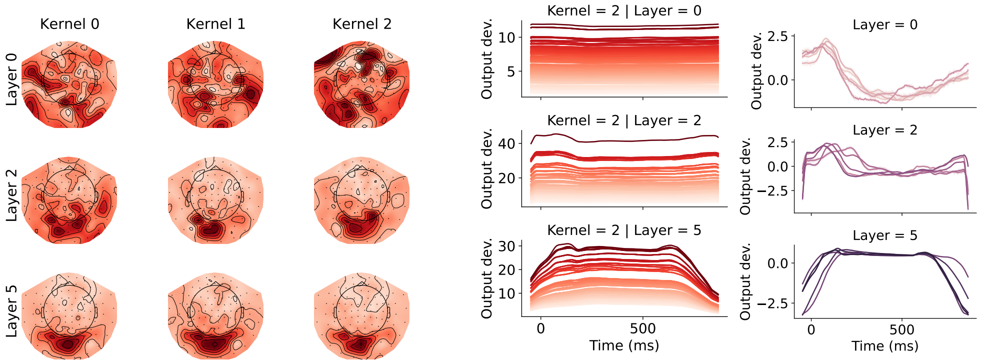
(a) Temporal gradient analysis

(b) Spatial gradient analysis

Figure 4.10: Using gradient analysis by backpropagating the loss to randomly initialised inputs with the trained `non-linear group-emb` model. In (a) we can see the temporal profile of the gradients averaged over channels. In (b) we can see the spatial profile of the gradients averaged over time.

could do in a classical linear model. When using deep learning it is important to ask how the trained model arrives at the information presented in Section 4.3.4. We can leverage the structure of the model, i.e. the successive layers, and the filters in the convolutional layers can be regarded as individual computational units. The aim here is to understand the model itself and how it represents and processes the data internally. This is in line with previous efforts showing how successive layers in a deep convolutional model align with the visual system of the brain (Kriegeskorte, 2015).

Figure 4.11 shows results for just 3 of the 6 convolutional layers, with all 6 layers shown in Figure B.1 and Figure B.2. Kernels within a layer tend to have similar temporal sensitivity, and hence we only show 5 out of over 1e5 total kernels (Figure 4.11c). Output deviations are standardised to compare temporal PFI across kernels with different output magnitudes. In the early layers, sensitivity peaks around 100 ms (as in Figure 4.9), then rapidly decreases, eventually climbing again



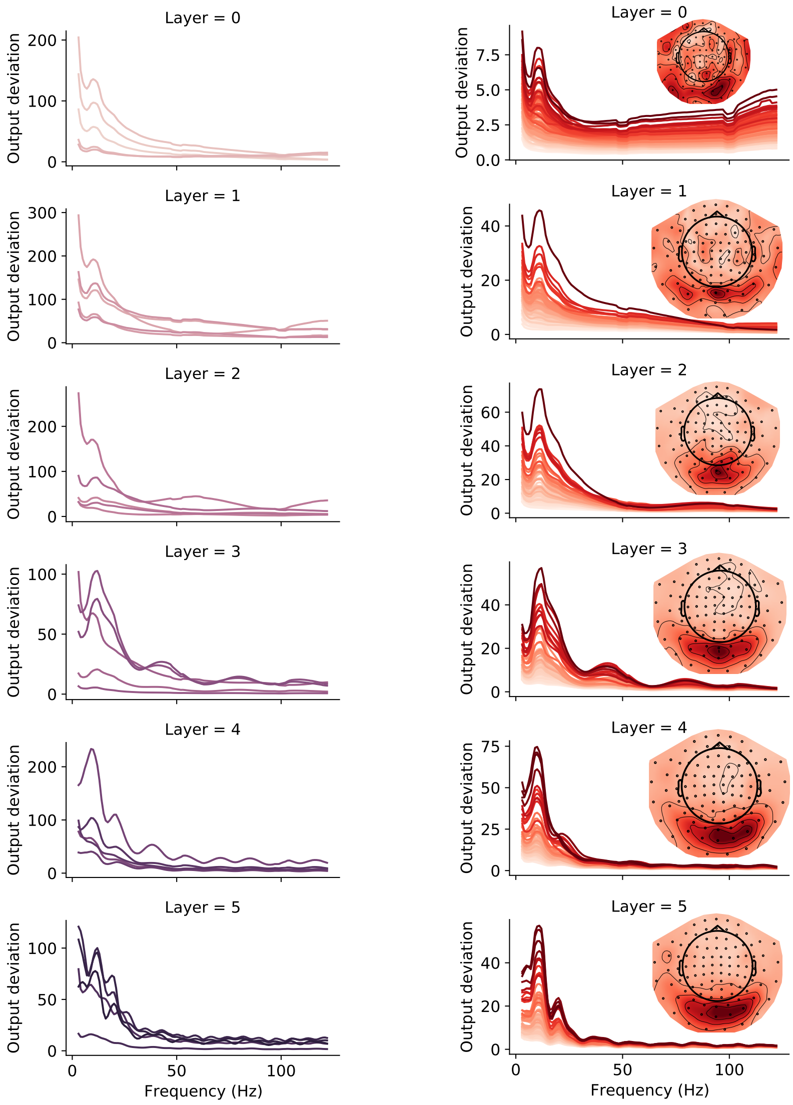
(a) Spatial PFI (b) Spatiotemporal PFI (c) Temporal PFI

Figure 4.11: Spatio-temporal insights can be obtained using PFI. Spatial (a), channel-wise temporal (b), and temporal (c) PFI across `non-linear group-emb` kernels within 3 layers (rows). For spatial PFI, kernels are plotted separately; whereas for temporal PFI, 5 kernels (lines) are plotted together. Channel-wise temporal PFI shows the temporal PFI of each channel for Kernel 2. Channel colouring is matched to the corresponding spatial PFI map, and darker reds mean higher output deviation. For temporal PFI, output deviation is normalised. The horizontal axis shows the time elapsed since the image presentation, for both temporal PFI types. 95% confidence intervals are shown with shading.

slowly. Kernels in early layers have somewhat random spatial sensitivity (Figure 4.11a), but this gets narrowed down to channels over the visual cortex in deeper layers, with some differences between individual kernels. This sensitivity is similar to the spatial features that were shown to be most informative for classification performance (see Figure 4.9).

Figure 4.11b shows the temporal profile of the spatial PFI. This is achieved by limiting the shuffling to 100 ms time windows and 4-channel neighbourhoods (3 closest channels for each channel) at a time, which is then repeated across all time points and channels. This shows spatial sensitivity does not seem to change with time; i.e. the most important channels are always the same, also observed in previous spatiotemporal PFI analyses of this dataset presented in Chapter 3.

In neurophysiology, we are often interested in the oscillatory content of the signal,



(a) Spectral PFI (b) Channel-wise Spectral PFI

Figure 4.12: Frequency sensitivity of kernels via spectral PFI (a), channel-wise spectral PFI (b) of the trained `non-linear group-emb` model in 6 layers (rows). Kernels are plotted together (lines) for spectral PFI. Each channel-wise spectral PFI plot is for 1 kernel, where lines show the spectral PFI of corresponding channels in the topomap. 95% confidence interval is shown with shading for spectral PFI. Due to small variability across permutations, this is barely visible.

and what/how specific frequencies are associated with certain tasks, here, decoding of visual stimuli. To this end, we use PFI in the spectral domain, where it is used to measure the change in kernel output to perturbations in specific frequency bands (Figure 4.12a). Across all layers and kernels, the profile has a $1/f$ (frequency) shape with a clear peak at 10 Hz. These are common features of the MEG signal (Demanuele et al., 2007; Drewes et al., 2022), indicating that the spectral sensitivity of the kernels coincides with the power spectra of the data. Our previous model-level spectral PFI analysis in Chapter 3 did not show the 10Hz peak so clearly because of lower sampling rates. In Figure 4.12b, we also looked at the spectral PFI of 4-channel neighbourhoods and found that kernels are sensitive to the same channels (in the visual area) across all frequencies, with these channels having larger 10 Hz peaks. Further spectral investigations are presented in Appendix B.

In summary, the analyses presented in this section show that the kernels are sensitive to interpretable temporal, spatial, and spectral features of the MEG data. Specifically, we have shown that kernels are sensitive to channels in the visual area, with this sensitivity getting more focused in deeper layers. Kernels are also sensitive to the 10Hz peak of the MEG data, and the temporal sensitivity shows a peak at 100-150ms in early layers.

4.4 Discussion

In this chapter, we focused on across-subject decoding, motivated by the fact that group-level models that perform well in this manner can be useful for gaining neuroscientific insights that are relevant at the group level. In this setting, our proposed deep learning-based group-level model outperforms naïve group models and achieves similar performance to subject-level models, but with three key benefits. First, it provides potentially richer insights at the group level. As nonlinear models are required for the group-level decoding to work, we had to use PFI, and showed that it is effective in the case of nonlinear group-level models. Second, there is potential for the group-level model to outperform subject-level models when larger population datasets are available. Third, a group-level model can be used to initialise subject-level models surpassing the performance of subject-level models initialised randomly. We have shown how subject embeddings and non-linearity are crucial for this. These are important insights towards the goal of using group models in decoding neuroimaging data, which would allow for better use of this inherently limited resource.

Interestingly, we found that at the subject level, linear models perform better than their nonlinear counterparts. Although some studies have found deep learning to improve over simpler linear models, this improvement is often marginal (Cooney et al., 2019b; Schirrmeister et al., 2017a). Such results are difficult to generalise across different MEG datasets, due to variability in tasks, the number of subjects, and the amount and quality of data (Schirrmeister et al., 2017a).

Other than being useful for fine-tuning, our embedding-aided group model can be useful in the case of much larger datasets, where we cannot afford to have a separate model for each subject. As we have shown, even in this limited dataset with 15 subjects, the group model can provide improvement in a few subjects. Our results suggest follow-up studies to understand why some subjects performed better or worse. A current limitation of our approach is that it is still worse than subject-level models (on average).

We have demonstrated the use of PFI on group models to obtain insight into which time points and channels contributed to the decoding and to obtain meaningful information encoded in convolutional kernels. PFI can also provide group-level temporal, spatial and spectral information by averaging over linear subject-level models. Here our aim was to show that PFI works similarly well in the case of non-linear group-level models. Using this and other methods, such as representational similarity analysis, neuroscientific investigations can be performed at the group level using a single model, instead of averaging over individual subject models. We note that one downside of PFI is that the absence of influence on the output does not necessarily mean that a specific channel or time window does not carry information about the target variable. When applying PFI to kernel outputs it is unclear how to summarise and visualise this information across millions of kernels. This is one downside of using deep learning models with such a large parameter space. As future work it may be useful to assess the usage of subject embeddings with PFI.

While the across-subject decoding we focus on in this work is most relevant to situations where we want to obtain insights at the group-level, other applications, such as BCIs that need to work well on previously unseen subjects, may be more appropriately evaluated using leave-one-subject-out (LOSO) evaluation. In this context, we found that using subject embeddings did not improve performance. Exploiting subject embeddings in a pure LOSO framework is not trivial, as some additional approach is needed to initialise/learn the embedding of the left-out subject in an unbiased manner. We have not tried to only optimise the subject embedding while freezing the rest of the model. While computationally less expensive, this is not expected to be as good as optimising the whole model (and subject embedding) on the new subject, which we have presented in Section 4.3.3. In larger datasets with more subjects, between-subject similarities in the embeddings could be exploited and different heuristics explored, e.g. initialising the embedding with the average of all learned subject embeddings. However, research aimed at improving performance in new subjects often leverages transfer learning in some way, where a limited amount of data from the new subject can be used (Zubarev et al., 2019). In this scenario, we think our across-subject group model could be helpful, by, for example, using the limited data from the new subject or by learning a useful embedding for the new subject in an unsupervised manner. As we have shown in Section 4.3.1 this could be especially useful for subjects with low performance.

As opposed to a naive continuation of the trends in Figure 4.7, we expect that with more trials, the gap between group initialisation and training from scratch would continue, up to some limit. We believe that the reason why the gap closes at 100% training data is due to the ratio of training and validation sets and the low number of examples. The small validation set (6 examples per class) is probably not representative of the full data distribution.

We expect the subject embedding and group modelling to generalise to different task and recording modalities (EEG, fMRI, etc.) because they face similar decoding challenges. The specific Wavenet-based model is readily generalisable to other electrophysiological data such as EEG and electrocorticography (ECoG), because of the same temporal dynamics they capture. Further research is needed into deep learning models capable of implicitly learning inter-subject variability. An important question is whether scaling up models on large datasets would achieve this goal.

5 | Forecasting MEG signals

In previous chapters, we have presented methods for dealing with within-subject and between-subject variability in MEG decoding. When addressing such variability, we assumed that all data came from the same experiment and scanner, which is a serious limitation of these approaches. Single experiments usually investigate only a few research questions with limited dataset sizes. By utilising multiple datasets collected by different researchers, we could potentially achieve the scale required for deep learning to be truly applicable. Training a single model on multiple datasets allows us to apply it to various encoding and decoding paradigms, which could be especially useful for experiments with small datasets.

However, there are two major challenges to overcome. First, the decoding models used so far are not well suited for generalisation across datasets, as not all data is recorded with decoding in mind, such as rest data. It would also be difficult to include the vastly different experimental stimuli from every dataset into a single decoding model. Second, variability in electrophysiology becomes even more problematic, as we now must address variability not only between subjects but also between different experimental setups and scanners.

For the first issue, the title of this chapter (forecasting) provides a natural solution. As discussed in Chapter 2, forecasting of multivariate time series is a general task that can be formulated for any M/EEG dataset. Thus, we need not be concerned with specific experimental paradigms or recording modalities. Moreover, by using an unsupervised modelling framework, we can potentially learn shared representations across datasets, which could then prove useful for specific tasks such as encoding or decoding. For the second issue, a solution that often works well in deep learning is scale. That is, by using more and more data and sufficiently expressive models, variability can be implicitly learned and modelled. Unsupervised modelling (e.g., forecasting) also enables larger scale, by utilising all timepoints for training, rather than using only a subset as in encoding or decoding

In this chapter, we explore how to design deep learning forecasting models that can reproduce spatiotemporal dynamics and various other properties of MEG data. We present both Wavenet-based (van den Oord et al., 2016) and Transformer-based (Vaswani et al., 2017) models, comparing them with standard linear autoregressive (AR) modelling on MEG data.

We show that Transformer-based models provide better modelling capabilities than Wavenet and linear AR models by reproducing the HMM statistics of real data and evoked activity in task data. Through a series of ablations, we demonstrate which aspects of the Transformer-based models enable these improvements. In the case of the Transformer, our design includes a novel application of tokenisation methods, allowing such a model developed for the discrete domain of language to be applied to continuous multichannel time series data.

We also extend the forecasting framework to work with condition labels as inputs, enabling better modelling (encoding) of task data. Finally, we present a method for transforming a forecasting model into a generative decoder through the use of Bayes' theorem.

To be clear, we do not apply our methods to multiple datasets, as our aim was to develop proof-of-concept models and analyse them on a reasonably sized dataset. Testing on multiple datasets at scale is left for future work.

5.1 Introduction

Unsupervised learning provides a dataset-agnostic method for learning shared representations. Within unsupervised learning, we can further differentiate between methods aiming to learn interpretable representations and purely data-driven approaches. The goal of interpretable models is to provide neuroscientific insights into electrophysiology data in an unsupervised manner. This is especially useful for rest data, where there is no external stimulus or behaviour linked to the brain activity. Models designed without focusing on interpretability can still be analysed using the techniques mentioned in Section 2.5.5. However, such models are primarily used to generalise over multiple heterogeneous datasets and provide a pre-trained foundation model. By leveraging large amounts of data, the hope is that the model will be capable of generalising to new data types and provide improvement over a model trained on a single, small dataset. This is especially useful for BCI settings.

The concept of using vast amounts of data to boost performance in downstream tasks originates from deep learning. Perhaps the most successful recent example is that of large language models, trained on diverse data sources and demonstrating enhanced capabilities over task-specific models in a multitude of language-related tasks (e.g., translation, summarisation) (Brown et al., 2020b). This can also be viewed as a form of transfer learning. Zero-shot performance is obtained when no fine-tuning is done for the downstream task.

Several factors enabled the success of large language models, including data scale, model size, fast GPUs, and effective neural network architectures (Kaplan et al., 2020; Fedus et al., 2022; Sutton, 2019). To adopt this paradigm for electrophysiology data, the primary obstacles are the model architecture and data size. In this chapter, we focus on the former. Unfortunately, the number of brain-recording datasets is limited due to the high cost of data collection. Recordings are often not publicly released because of privacy concerns. To achieve data scale comparable to language modelling, lowering the financial barrier to collecting brain data and advocating for public release is needed. While language data is freely available online, brain data is far more difficult to find via automated scraping, has much higher dimensionality requiring more storage and download bandwidth, and is far more heterogeneous due to differences in scanners and experiments.

Our focus in this chapter is designing general models well-suited to multichannel timeseries that can scale effectively. We also focus on using forecasting models, which are causal and can generate data recursively, as they achieve a good balance between interpretability and scalability. While some unsupervised models aimed at neuroscientific investigations have been proposed (Gohil et al., 2022) here we focus on reviewing more data-driven self-supervised approaches.

5.1.1 Self-supervised learning

Self-supervised learning (SSL) has emerged as a promising approach for learning useful representations from unlabelled electrophysiological data. SSL reformulates an unsupervised learning problem as a supervised one by exploiting inherent structure in the data to generate "pseudo-labels". In the context of electrophysiology, recent works have proposed SSL tasks tailored to the temporal and multivariate nature of neural time series data (Banville et al., 2021; Kostas et al., 2021; Wang et al., 2023).

Banville et al. (2021) investigate three SSL tasks for learning from unlabelled EEG recordings. Each task is trained via a contrastive loss function, where the model learns to pull positive pair examples closer in a representation space while pushing negative pairs apart. They demonstrate that the representations learned via SSL on unlabelled EEG data transfer well to supervised downstream tasks, consistently improving over limited label training and matching full supervision performance.

Building on this Kostas et al. (2021) propose combining self-supervised contrastive learning with Transformer networks to enable pre-training on large amounts of unlabelled EEG data. Their approach, BErt-inspired Neural Data Representations (BENDR), uses a Transformer encoder architecture applied to learned representations of raw EEG segments. A technical description of Transformer models is provided in Section 5.2.3. First, a temporal convolutional network extracts initial representations of the EEG time series, referred to as BENDR features. Next, a Transformer encoder module takes the BENDR features as input. Contiguous segments of the BENDR representations are randomly masked, and the model is trained via a contrastive loss to predict the original features. Fine-tuning the pretrained model significantly improves performance on supervised EEG analysis tasks compared to training just on the downstream datasets.

5.2 Methods

In our quest for designing expressive forecasting models of MEG data, we can look to artificial intelligence domains with similar characteristics, such as audio or natural language processing. These domains share some similarities with MEG data, like the sequential nature of the modality. However, while audio data is also a continuous timeseries, it only contains a single channel and comes at a much higher sampling rate compared to M/EEG data. Language data is perhaps even more different as its timeseries are comprised of distinct units (words) from a finite vocabulary set. As such, starting with models developed for these domains and adapting them to handle the nuances of M/EEG data is a promising approach.

Indeed, in this chapter we adapt Wavenet, originally developed for forecasting audio data (van den Oord et al., 2016), and GPT-2, originally developed for forecasting language (Radford et al., 2019).

5.2.1 Wavenet

Here we describe the Wavenet architecture (Figure 5.1) used in the original paper (van den Oord et al., 2016), and how we adapted it for electrophysiological data. Wavenet models the conditional probability of each time sample given all preceding samples autoregressively:

$$p(\mathbf{X}) = \prod_{t=1}^{T} p(\mathbf{x}_t | \mathbf{x}_1, ..., \mathbf{x}_{t-1}) \tag{5.1}$$

where $\mathbf{x}_t$ is the sample at time t and T is the total sequence length. Unlike our simplified Wavenet used in Chapter 4 which outputs point estimates of the continuous value of the data at the next timepoint, the full network predicts a categorical distribution over tokenised samples using a softmax output layer. Throughout this chapter we use tokenisation and quantisation interchangeably. Both have the aim of discretising a continuous quantity into a finite set of distinct bins/levels/tokens.

In the original paper, the audio waveform is tokenised using a quantisation to 8 bits following a μ-law companding transform (Lewis and MTSA, 1997):

$$f(\mathbf{x}_t) = \operatorname{sign}(\mathbf{x}_t) \frac{\ln(1 + \mu|\mathbf{x}_t|)}{\ln(1 + \mu)} \tag{5.2}$$

where μ controls the number of quantisation levels, set to 255 as in the original Wavenet. $f(.)$ is applied to each value of $\mathbf{x}_t$ independently. This nonlinear transformation improves reconstruction versus uniform quantisation of the raw input, as it

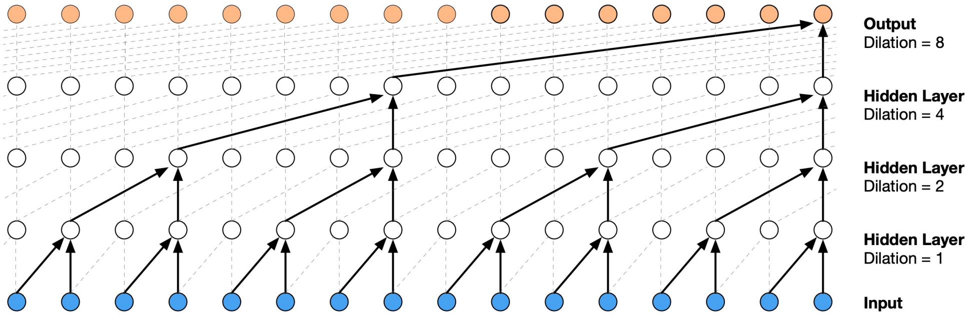


Figure 5.1: A stack of dilated convolutions, the core architecture of Wavenet. The dilation factor is doubled in successive layers. Figure from van den Oord et al. (2016).

skews the distribution such that more levels are allocated to smaller magnitudes. For MEG data, we observe similar benefits when applying this transform prior to quantisation. Note that the input must be scaled to $(-1, 1)$ first, and clipping outliers above some threshold helps ensure a more uniform mapping.

Critically, tokenisation, in this case through quantisation, enables modelling of probability distributions over data and sampling, instead of just point estimates from MSE-based training. Cross-entropy loss also avoids the mean-prediction bias induced by MSE (Banville et al., 2021).

Wavenet uses nonlinear activation functions and skip connections between layers (Figure 5.2). The computations for layer l are:

$$\mathbf{Z}^{(l)} = \tanh\left(\mathbf{W}_f^{(l)} * \mathbf{H}^{(l)}\right) \odot \sigma\left(\mathbf{W}_g^{(l)} * \mathbf{H}^{(l)}\right) \tag{5.3}$$

$$\mathbf{S}^{(l)} = \mathbf{W}_s^{(l)} * \mathbf{Z}^{(l)} \tag{5.4}$$

$$\mathbf{H}^{(l+1)} = \mathbf{W}_r^{(l)} * \mathbf{Z}^{(l)} + \mathbf{H}^{(l)} \tag{5.5}$$

where $*$ is convolution, $\odot$ is element-wise multiplication, σ is the sigmoid, $\tanh$ is

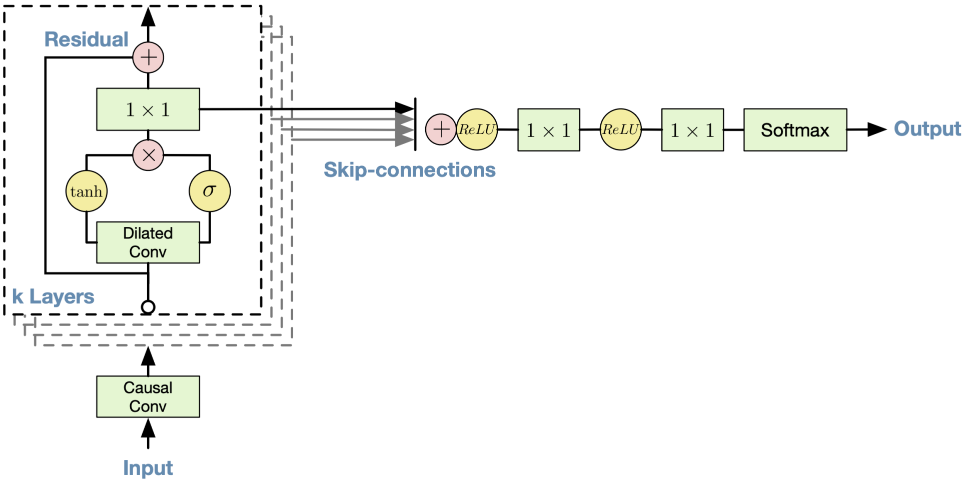


Figure 5.2: Overview of the full Wavenet architecture with gated dilated convolutions and residual connections. Figure from van den Oord et al. (2016).

the hyperbolic tangent function. $\mathbf{W}_f^{(l)}$ and $\mathbf{W}_g^{(l)}$ are the filter and gate convolutions. $\mathbf{S}^{(l)}$ is the skip connection output, and $\mathbf{H}^{(l+1)}$ is the residual layer output. Importantly, the residual and skip convolutions ($\mathbf{W}_r^{(l)}$ and $\mathbf{W}_s^{(l)}$) use 1x1 kernels, while the initial two are dilated.

The skip outputs are summed across layers and passed through further 1x1 convolutions:

$$\mathbf{S} = \sum_{l=1}^{L} \mathbf{S}^{(l)} \tag{5.6}$$

$$\mathbf{Y} = \mathrm{Conv1x1}\left(\mathrm{ReLU}\left(\mathrm{Conv1x1}\left(\mathrm{ReLU}\left(\mathbf{S}\right)\right)\right)\right) \tag{5.7}$$

$$\hat{\mathbf{X}}_{T+1} = \mathrm{Softmax}(\mathbf{Y}) \tag{5.8}$$

By setting the channels of the last convolution to the number of tokens, i.e. the vocabulary size which in this case is the number of quantisation bins ($Q = 256$),

$\mathbf{Y}$ represents logits over tokens and $\hat{\mathbf{X}}_{T+1} \in \mathbb{R}^{C \times Q}$ gives the predicted distribution at $T+1$. Cross-entropy loss can then train the model to accurately forecast future timesteps.

5.2.2 Multi-channel Wavenet

When adapting Wavenet to M/EEG, a key challenge is the multi-channel nature of the data. We devise two versions: `WavenetFullChannel` as univariate, and `WavenetFullChannelMix` as multivariate. In both, each channel is transformed and tokenised independently to form the input to the models.

In `WavenetFullChannel`, we first apply an embedding layer to the tokenised data, learned separately per channel. The embedding layer represents each discrete bin as a high-dimensional continuous vector, enabling powerful representations in the convolutional layers whose input channels match the embedding size. To be clear in this univariate approach the same model is applied to each channel. However, a different embedding layer is learned for each channel, meaning that for example the quantised value of 0.42 in channel x will have a different vector representation than in channel y. This helps the model differentiate between channels.

The embedding operation is given below:

$$\forall c \in 1, 2, \ldots, C : \mathbf{X}_e^{(c)} = \mathbf{W}^{(c)}\mathbf{X}^{(c)} \tag{5.9}$$

$$\mathbf{H}_0 = \text{Concatenate}(\mathbf{X}_e^{(1)}, \mathbf{X}_e^{(2)}, \ldots, \mathbf{X}_e^{(C)}) \tag{5.10}$$

Here, $\mathbf{X}^{(c)} \in \mathbb{R}^{Q \times T}$ is the tokenised one-hot input and $\mathbf{W}^{(c)} \in \mathbb{R}^{E \times Q}$ is the embedding layer of channel c mapping tokens Q to embeddings of size E. Concatenate

concatenates along the channel dimension.

$\mathbf{H}_0 \in \mathbb{R}^{C \times E \times T}$ is the resulting input to Wavenet with C as the batch dimension. Thus, the same model is applied independently to each channel in parallel. At output, a distribution is predicted simultaneously for each channel at $T + 1$. The model is optimised to accurately predict all channels.

`WavenetFullChannelMix` includes an extra linear layer after summing the skip representations to mix information across the channel dimension:

$$\mathbf{S} = \sum_{l=1}^{L} \mathbf{S}^{(l)} \tag{5.11}$$

$$\mathbf{S} = \mathbf{S}.\text{permute}(1, 2, 0) \tag{5.12}$$

$$\mathbf{S}_{out} = \mathbf{S}\mathbf{W}_m \tag{5.13}$$

where $\mathbf{W}_m \in \mathbb{R}^{C \times C}$ is the mixing weight matrix. The permutation is needed to apply the projection to the appropriate channel dimension. After this $\mathbf{S}_{out}$ is permuted back to the original dimension order and the rest proceeds identically to `WavenetFullChannel`.

In the original Wavenet, audio generation can be conditioned on additional inputs through embedding-based global conditioning or time-aligned local conditioning. For some experiments, we augment the model with local features of task stimuli or subject labels, first embedded into continuous vectors:

$$\mathbf{H}_y = \mathbf{Y}\mathbf{W}_y \tag{5.14}$$

$$\mathbf{H}_o = \mathbf{O}\mathbf{W}_o \tag{5.15}$$

$$\mathbf{H}_c = \text{Concatenate}(\mathbf{H}_y, \mathbf{H}_o) \tag{5.16}$$

where $\mathbf{Y} \in \mathbb{R}^{T \times N}$ contains the condition index $n \in (1, \ldots, N)$ at each time point, and $\mathbf{O} \in \mathbb{R}^{T \times S}$ contains the subject index $s \in (1, \ldots, S)$ at each time point $t \in (1, \ldots, T)$. $\mathbf{W}_y \in \mathbb{R}^{N \times E_n}$ and $\mathbf{W}_o \in \mathbb{R}^{S \times E_s}$ are embedding matrices mapping the labels to learned continuous vectors of size E_n and E_s, respectively. The subject index is the same across time points of the recording from the same subject. The condition index is set to the (visual) stimuli presented (e.g., one of the 118 images in Cichy et al. (2016)), for exactly those time points when the stimulus is on. At any other time, the task condition embedding $\mathbf{H}_y$ is set to 0.

$\mathbf{H}_c$ is the conditioning vector fed into Wavenet at each layer, modifying Equation 5.3 to:

$$\mathbf{Z}^{(l)} = \tanh\left(\mathbf{W}_f^{(l)} * \mathbf{H}^{(l)} + \mathbf{W}_c^{(l)} * \mathbf{H}_c\right) \odot \sigma\left(\mathbf{W}_g^{(l)} * \mathbf{H}^{(l)} + \mathbf{W}_c^{(l)} * \mathbf{H}_c\right) \tag{5.17}$$

where $\mathbf{W}_c^{(l)}$ (1x1 convolution) projects $\mathbf{H}_c$ before adding it to the input representation. This conditions the prediction on both past brain activity and stimuli:

$$p(\mathbf{X}|\mathbf{Y}, \mathbf{O}) = \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{x}_1, ..., \mathbf{x}_{t-1}, \mathbf{y}_1, ..., \mathbf{y}_{t-1}, \mathbf{o}_1, ..., \mathbf{o}_{t-1}) \tag{5.18}$$

In single-subject models we only use the task labels $\mathbf{Y}$.

The full Wavenet architecture can either be interpreted as forecasting with extra conditioning or as a generative encoder augmented with past brain activity. In addition, the probabilistic formulation allows converting the model into a decoder using Bayes' rule, enabling both forecasting and decoding within the same framework:

$$p(Y|X) = \frac{p(X = x|Y)p(Y)}{p(X = x)} \tag{5.19}$$

where X is the random variable representing the data, Y is the random variable representing the task labels, and x is a particular sample of X. $p(Y)$ is the task label prior distribution which in the 118-image dataset is uniform. $p(X = x|Y)$ is the likelihood of the data given the label which we get from the above formulation of Wavenet. The only tricky part is $p(X = x)$ as this requires marginalisation over Y. In the case of the 118-image dataset this means that we have to run the trained model with all of the possible task labels to obtain $p(X = x)$:

$$p(X = x) = \sum_{i=1}^{N} p\left(X = x|Y = i\right) p\left(Y = i\right) \tag{5.20}$$

Thus, in a single self-supervised deep learning model we have flexibly encapsulated forecasting, encoding, and decoding, all three of the main modelling methods of M/EEG data. This unification of modelling approaches was inspired by a GitHub repository applying similar ideas to images[1]. The inverted decoder formulation also allows for iterative estimation of $p(Y|X)$ at each timestep. The author of the GitHub repository has applied this method to estimating the probability of image

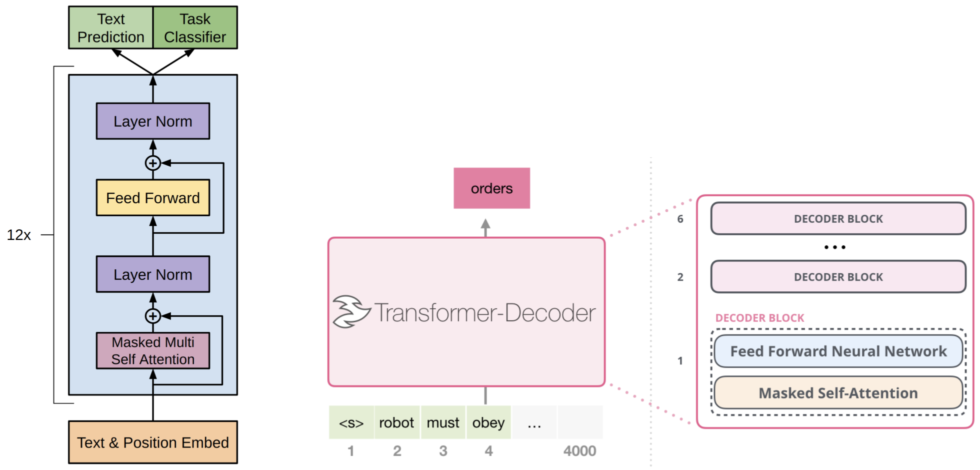
[1] https://github.com/cheind/autoregressive

labels (digits 0 to 9) from pixel images, as more and more of the image was fed into the model.

5.2.3 GPT2

While Wavenet is an effective model for forecasting time series, it may be that other types of architectures are better suited for multichannel data. The dilated convolutional architecture, while fast and parameter-efficient, might limit the model's expressivity, particularly when scaling up on multiple datasets. Indeed, in recent years there has been a second deep learning revolution driven by the Transformer architecture (Vaswani et al., 2017). Unlike the weight sharing and autocorrelation inductive biases (priors) of convolutional models, Transformers have two key architectural priors. First, they are sequential models operating on a discrete set of input tokens (e.g. words), mapped to continuous embeddings. Second, their primary mode of processing these representations is the attention mechanism. This allows Transformers to model complex dependencies across long sequences without regard to their distance in the input or output. This provides a more flexible inductive bias well-suited to language modelling and other tasks involving highly structured sequential data (Devlin et al., 2019a; Brown et al., 2020a).

Since their introduction, Transformers have become the dominant model architecture for natural language processing (NLP). Models like BERT (Devlin et al., 2019a), GPT-2 (Radford et al., 2019), and GPT-3 (Brown et al., 2020b) have achieved state-of-the-art results on a wide range of NLP benchmarks. The success of Transformers for language has led researchers to apply them to other sequential modelling tasks. For time series forecasting, Transformer-based models offer several potential advantages over RNNs and temporal convolutional networks. The self-attention mechanism provides direct connectivity between any two time



(a) GPT2 architecture (b) GPT2 architecture (alternative visualisation)

Figure 5.3: Two visualisations of the core GPT2 architecture for language modelling. Figures from Radford et al. (2018) (left) and Alammar (2019) (right).

steps, capturing long-range dependencies. Pre-trained representations like BERT can inject useful inductive biases from language modelling. The parallelisable architecture allows more efficient computation compared to recurrent models.

Early explorations of Transformers for time series have shown promising results. Zhou et al. (2021) adapted the self-attention mechanism for long-range forecasting and demonstrated state-of-the-art performance on multiple public datasets. As with NLP, we expect Transformer models to become a leading approach for time series modelling (Wen et al., 2022).

Thus, we set out to design a Transformer model suited for M/EEG data, while keeping the key elements that made it successful in language modelling. Specifically, we use GPT-2, a popular autoregressive Transformer variant. When adapting GPT-2 to continuous multivariate time series, the main challenges are at the input and output layers interfacing the model with the data. We describe GPT-2 for language modelling first, then present our modifications. A particularly detailed visual description of GPT2 is given in Alammar (2019).

GPT2 utilises a multi-layer Transformer decoder[2] architecture (Figure 5.3). Each layer contains two sublayers: a multi-head self-attention mechanism and a position-wise feedforward network. Residual connections and layer normalisation are employed around each sublayer. The self-attention mechanism allows the model to attend over previous positions when generating the next token. It operates on query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$ projections of the layer input $\mathbf{H}^{(l)} \in \mathbb{R}^{T \times E}$:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax} \frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{E}} \mathbf{V} \tag{5.21}$$

where E is the dimension of the feature space inside the model, also called hidden dimension, often set to the embedding size. $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are computed by:

$$\mathbf{Q}^{(l)} = \mathbf{H}^{(l)} \mathbf{W}_Q^{(l)} \tag{5.22}$$

$$\mathbf{K}^{(l)} = \mathbf{H}^{(l)} \mathbf{W}_K^{(l)} \tag{5.23}$$

$$\mathbf{V}^{(l)} = \mathbf{H}^{(l)} \mathbf{W}_V^{(l)} \tag{5.24}$$

where $\mathbf{W}_Q^{(l)}, \mathbf{W}_K^{(l)}, \mathbf{W}_V^{(l)} \in \mathbb{R}^{E \times E}$ are learned projections. A crucial function of the attention mechanism is comparing different vector representations of elements in the input sequence through the dot-product similarity measure. Most Transformer variants use multi-head attention computing attention separately over multiple distinct feature partitions, concatenating the outputs:

[2] Here the term decoder is used as an architecture type in deep learning, i.e., autoregressive forecasting, as opposed to a decoder of brain data.

$$\forall i \in N : \mathbf{H}_i^{(l)} = \mathbf{H}^{(l)}[:, i : i + d] \tag{5.25}$$

$$\mathbf{Z}_i^{(l)} = \text{Attention}(\mathbf{H}_i^{(l)}\mathbf{W}_{Q_i}, \mathbf{H}_i^{(l)}\mathbf{W}_{K_i}, \mathbf{H}_i^{(l)}\mathbf{W}_{V_i}) \tag{5.26}$$

$$\text{MultiHeadAttention}(\mathbf{H}^{(l)}) = \text{Concatenate}(\mathbf{Z}_1^{(l)}, \ldots, \mathbf{Z}_N^{(l)})\mathbf{W}_O^{(l)} \tag{5.27}$$

where N is the number of attention heads, and $d = \frac{E}{N}$ is the feature dimension of a single head. Note that the feature dimension depends on the number of heads, as the dimensionality of all heads has to sum up to E. $\mathbf{W}_{Q_i}, \mathbf{W}_{K_i}, \mathbf{W}_{V_i}$ are the projections for head i, and $\mathbf{W}_O^{(l)}$ is an output projection.

The feedforward layer applies two affine transforms with ReLU activation:

$$\text{FFN}(\mathbf{Z}) = \text{ReLU}(\mathbf{Z}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \tag{5.28}$$

This allows learning nonlinear representations of the input at each position. Altogether a GPT2 layer is the combination of the self-attention and feedforward layers:

$$\mathbf{H}^{(0)} = \mathbf{X}\mathbf{W}_e + \mathbf{W}_p \tag{5.29}$$

$$\mathbf{Z}^{(l)} = \text{LN}(\mathbf{H}^{(l)} + \text{MultiHeadAttention}(\mathbf{H}^{(l)})) \tag{5.30}$$

$$\mathbf{H}^{(l+1)} = \text{Dropout}(\text{LN}(\mathbf{Z}^{(l)} + \text{FFN}(\mathbf{Z}^{(l)}))) \tag{5.31}$$

$$\hat{\mathbf{Y}} = \text{softmax}(\mathbf{H}^{(L)}\mathbf{W}_e^T) \tag{5.32}$$

where $\mathbf{W}_e \in \mathbb{R}^{Q \times E}$ embeds the discrete tokens $\mathbf{X} \in \mathbb{R}^{T \times Q}$ into E dimensions. LN is Layer Normalisation, a regularisation technique which normalises all activations within a layer to zero mean and unit variance. $\mathbf{W}_p \in \mathbb{R}^{T \times E}$ contains positional encodings, providing the model with sequential order information. This is needed as GPT2 lacks recurrent or convolutional elements. Each vector in $\mathbf{W}_p$ indexed by $t \in (1, \ldots, T)$ contains a distinct E-dimensional representation of position t. The output $\mathbf{H}^{(L)}$ is projected back to the vocabulary via the transpose embedding matrix (weight tying). Alternatively, a separate output projection can be learned. The softmax output gives a token probability distribution.

GPT-2 is trained via supervised learning to predict the next token given previous context, minimising cross-entropy loss between model outputs $\hat{\mathbf{Y}}$ and ground truth targets $\mathbf{Y}$. To enable autoregressive training, $\mathbf{Y}$ is set to $\mathbf{X}$ shifted one timestep ahead. Crucially, to prevent information leakage from future timesteps $t+1, \ldots, T$, causal masking is applied in each self-attention layer, setting outputs that would reveal future information at position t to zero.

The mask $\mathbf{M} \in \mathbb{R}^{T \times T}$ is a lower triangular matrix:

$$\mathbf{M}_{ij} = \begin{cases} 0 & \text{if } i < j \\ -\infty & \text{if } i \geq j \end{cases} \tag{5.33}$$

5.2.4 Channel-independent GPT2

To apply GPT2 to our continuous multichannel time series data, we take a similar approach as with Wavenet by tokenising each channel independently using the same method as before. This serves as our equivalent of the discrete set of tokens in language modelling. The same GPT2 model is applied to each channel in parallel by setting the channel dimension as the batch dimension. We call this `ChannelGPT2`.

The input to the model includes the position embeding as well as subject and task-stimulus embeddings. We also add a label/embedding telling GPT2 which channel the current time series is coming from:

$$\mathbf{H}^{(0)} = \mathbf{X}\mathbf{W}_e + \mathbf{W}_p + \mathbf{Y}\mathbf{W}_y + \mathbf{O}\mathbf{W}_o + \mathbf{W}_c \tag{5.34}$$

where $+$ denotes element-wise addition, $\mathbf{X} \in \mathbb{R}^{C \times T \times Q}$ is the tokenisd input, $\mathbf{W}_c \in \mathbb{R}^{C \times T \times E}$ are the learned channel embeddings of size E, which are distinct for each channel $c \in 1, \ldots, C$ but constant across time t. $\mathbf{Y}$ and $\mathbf{O}$ are the task and subject index matrices, mapped to their respective embeddings. As with the positional encoding $\mathbf{W}_p$, we simply add all embeddings (task, subject, channel) into a single representation. Note that instead of having channel-specific embeddings of the tokenised input $\mathbf{X}$ we learn the same mapping $\mathbf{W}_e \in \mathbb{R}^{Q \times E}$ across channels. Channel information is provided to the model through the channel embeddings.

A serious limitation of this channel-independent GPT2 model is that when predicting a single channel, it does not receive information from other channels. This is analogous to a univariate autoregressive model and ignores crucial cross-channel dependencies in the data. To be clear we often use the term univariate AR modelling in the sense that a separate AR model is trained on each channel. In the case of channel-independent Wavenet and GPT2 models, we train one and the same model on all channels.

5.2.5 Flat GPT2

In the image domain, tokenisation is often abandoned, and a linear projection directly maps image patches to continuous vector representations (Dosovitskiy et al., 2020). Similarly, Nie et al. (2022) have designed a channel-independent Transformer architecture applied to overlapping patches of continuous time series for forecasting. While this facilitates the input, without tokens categorical outputs cannot be generated. As discussed, maintaining operations over tokens and categorical outputs are desirable GPT2 features for M/EEG data. This is because we would like to output probability distributions and train using the cross-entropy loss.

The tokenisation can happen either before or after mixing information across channels. The latter matches GPT2's original design. One example of this is vector quantisation, which is used to tokenise multiple channels in Jukebox, a successful autoregressive Transformer model used on audio data (Dhariwal et al., 2020). Before training the Transformer, a hierarchical VQ-VAE (vector quantized variational autoencoder (Van Den Oord et al., 2017)) learns discrete codes (tokens) from raw audio. Once trained, VQ-VAE can map a continuous time series to a discrete token sequence $\mathbf{z}$. In the second step of Jukebox, the VQ-VAE is kept fixed, and the discrete tokens are used to learn an autoregressive Transformer.

Importantly, VQ-VAE is applied to single-channel audio to compress the temporal dimension into discrete codes. For our application we would primarily want to apply vector quantisation to the channel dimension, to have a discrete token at each timestep, or perhaps across a few timesteps. While an adaptation of this could work on MEG data, we opted for a simpler non-deep learning method.

In `FlatGPT2` we apply vector tokenisation on small groups of channels using the Residual Quantiser algorithm (Babenko and Lempitsky, 2014) from the faiss library[3]. By using 30 channel groups (buckets) we obtain 30 tokens per timestep, which is already a 10-fold reduction of the original dimension space. However, to have a single token per timestep (as in language modelling) we flatten the feature

[3] https://github.com/facebookresearch/faiss/wiki/Additive-quantizers

dimension (buckets) when feeding tokens to GPT2, hence the name `FlatGPT2`. Our total sequence length then becomes $B{\cdot}T$, where B is the number of buckets and T is the number of timesteps. This approach is also motivated by the observation that language models include extra information such as context within the sequence, instead of the feature space. Thus, when predicting the token of bucket b, we treat the previous timesteps of the other buckets as contextual information. For brevity and because `FlatGPT2` showed mostly negative results we omit the full mathematical description which can be found in Appendix C.1.2. Our main results and discussion focuses on the channel-independent GPT2 approach.

5.2.6 Model interpretation

To evaluate whether Wavenet and GPT2 models accurately capture brain dynamics beyond just predictive performance, we develop several analysis techniques to interrogate what these models learn.

Data generation As mentioned in Section 2.5.2, generating new data from a trained model can reveal its capabilities. Different models have distinct generation procedures. Linear AR models take Gaussian noise as input and generate one timestep at a time. Gaussian noise is added to the output, which is appended to the input sequence. This recursive process is described by:

$$\mathbf{x}_t = \boldsymbol{\epsilon}_t + f(\mathbf{X}_{t-K:t-1}) \tag{5.35}$$

This intuitively treats the model f as a black-box infinite impulse response (IIR) filter, where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, 1)$, and K is the receptive field size. These models can also be analysed as finite impulse response (FIR) filters by removing recursion and using only noise inputs at each timestep: $\mathbf{x}_t = f(\boldsymbol{\epsilon}_{t-K}, \ldots, \boldsymbol{\epsilon}_{t-1})$.

For tokenised models (Wavenet and GPT2), we generate data by sampling from the predicted probability distribution and recursively feeding the sample back as input. Sampling can be done via argmax, top-p, top-k, or full distribution sampling. Argmax selects the bin/token with the highest probability, while top-k orders outputs by probability and samples from the top k (Holtzman et al., 2020). Top-p samples from the ordered outputs whose cumulative probability mass exceeds $p\%$ (Holtzman et al., 2020). Full distribution sampling treats the distribution as categorical and samples directly. While this makes sense intuitively, top-p and top-k sampling can often work better in practice by avoiding generation of low-probability tokens, and thus reducing noise.

We compare generated timeseries to real/simulated data using power spectral density (PSD), covariance, and Hidden Markov Model (HMM) statistics. For task-conditioned models, we assess reconstruction of task-dependent dynamics by feeding in task labels during generation and examining evoked responses. To evaluate how well models capture task activity, we apply standard decoding models (e.g., linear classification) to generated trials and compare performance to real data. We also evaluate the generalisability of decoders trained on generated data to real data. Strong similarity in these metrics would indicate accurate modelling of task responses.

By removing certain model components and evaluating performance, ablation studies assess the contribution of different architectural factors. We perform ablations on linearity, conditioning embeddings, input length, univariate/multivariate modelling, and sampling strategies.

5.3 Results

As our dataset of choice, we used the continuous 118-image data from Cichy et al. (2016). For each subject, the data was bandpass filtered between 1 and 50 Hz, and a notch filter was applied to remove line noise. Subsequently, independent component analysis (ICA) artifact rejection was performed with a dimensionality of 64. Components were visually inspected for each subject, and those that exhibited clear artefactual features (e.g. eye or cardiac signals) were removed. The data was then downsampled to 100 Hz. The continuous data was split into non-overlapping validation, test, and training sets. The validation and test sets included 4 trials of each of the 118 conditions, while the training set contained the remaining 22 trials. This non-overlapping uniform splitting of the continuous data was possible due to the experimental setup during data recording.

For each model other than `FlatGPT2`, the data from each channel was tokenised independently to 256 bins using a quantisation via the mu-law algorithm discussed in Section 5.2.1. To achieve uniform quantisation, we first standardised each continuous-data channel, clipped values higher than 4 or lower than -4, applied per-channel maximum absolute scaling to map the data to the range (-1, 1), and finally applied the mu-law transform and 8-bit quantisation.

Our aim was to evaluate several models and methods on this dataset. Due to computational constraints and limited iteration speed over experiments and methods, all experiments in the following sections were performed on a single representative subject, except in Section 5.3.4 where we explore our models on all 15 subjects.

We compared the performance of linear AR models, Wavenet-based models, and GPT2-based models. We trained univariate AR(255) models on each channel. Note that we did also assess multivariate AR models (results not shown), but this did not improve performance compared to the univariate AR. We trained `WavenetFullChannel` with a matched receptive field of 255, two stacks of dilation blocks (7 layers per block, doubling dilation factors), 256 hidden channels, 1024 skip channels, no dropout, and a 20-dimensional task embedding. `WavenetFullChannelMix` had the same architecture but 128 hidden channels and 512 skip channels. We used early stopping on the validation set. This means that we ran training until overfitting was observed, and then analysed the model version with the lowest validation loss. All our analyses were performed on the distinct test set.

Our Channel-independent GPT2 (`ChannelGPT2`) had a variable receptive field between 128 and 256. This means that during training the model encountered examples that had a sequence length between 128 and 256, rather than all examples having the same length. GPT2 is normally trained to output all timesteps in a sequence of length T, given previous timesteps. However, this means that for the second timestep, the receptive field is only 1. Ideally, we wanted to match the training setup of our Wavenet models, where the receptive field is always 256. However, this would significantly slow down training as the whole forward and backward pass must be recomputed at each timestep. We opted for a trade-off, where we set the minimum receptive field to 128, ensuring efficient training and that the model is not trained to predict shorter sequence lengths. Hyperparameters for `FlatGPT2` are given in Appendix C.2.3.

The embedding size of all inputs (token vocabulary, position, task, channel) was set to 96, and we used 12 GPT2 layers, with 12 attention heads. We used Huggingface's implementation[4], so the rest of the parameters were the same as in their configuration. Dropout was set to 0 and we used early stopping on the validation set.

[4] https://github.com/huggingface

On average both the mu-law, and the residual tokenisation achieved low reconstruction error. We tested the reconstructed data by performing evoked analysis, and classification of the task responses, and achieved comparable performance to the raw data (results not shown). Thus, both types of tokenisation add negligible quality loss to the data.

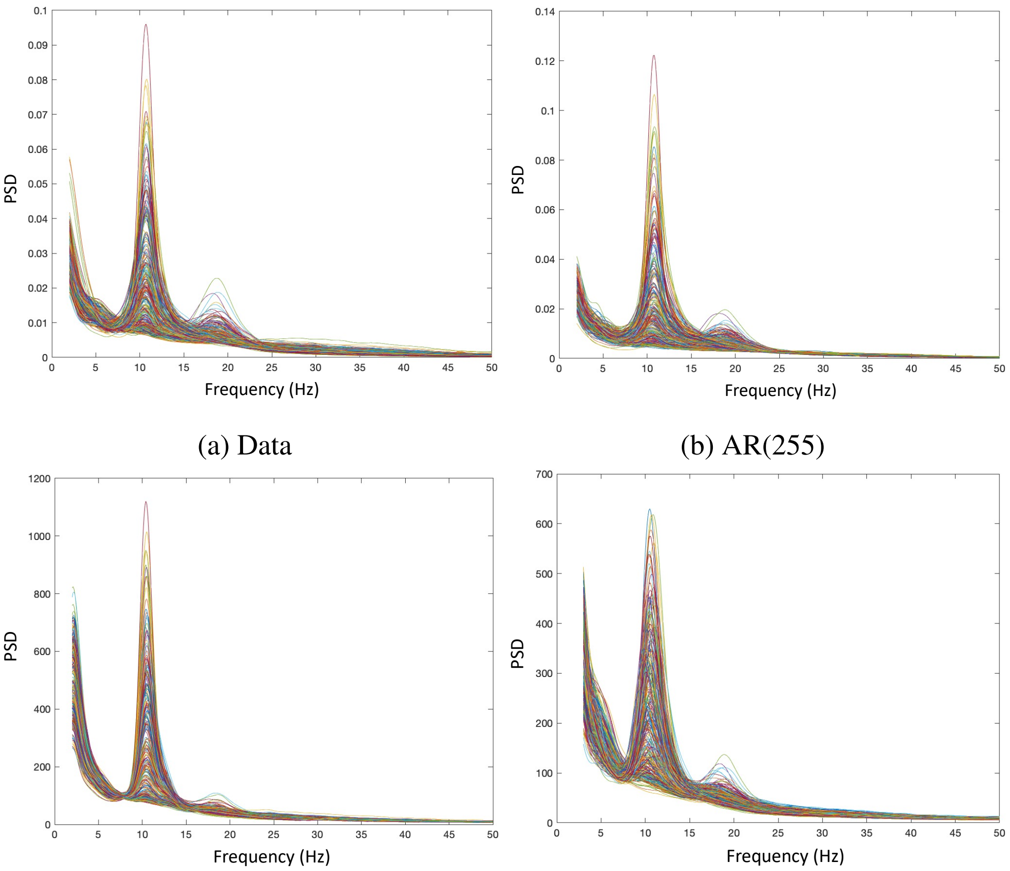
Forecasting performance in terms of token accuracy and mean-squared error is given in Appendix C.2.3. We found that these metrics are not very useful for comparing different models. What we are really interested in is how well they can generate the underlying spatiotemporal dynamics, which we investigate next.

5.3.1 Generating MEG data

For deep learning models we used top-p sampling with $p = 80\%$ (unless otherwise noted in the figure caption) to recursively generate data. We generated 3600 seconds with all models. For models that have task-conditioning (all except AR(255)) we use the task label timeseries from the training set.

Generated token sequences are first de-tokenised and then the power spectral density (PSD) is computed on the continuous data. Figure 5.4 compares the PSD of the generated data across our channel-independent models. AR(255) clearly reproduces the MEG data PSD, and `WavenetFullChannel` and `ChannelGPT2` also do a good job with slight differences. All models capture the characteristic $1/f$ shape, and peaks at 10 and 19 Hz, likely related to alpha and beta band activity. Notably, `WavenetFullChannel` has reduced power at the 19 Hz peak which could indicate issues in capturing higher frequency dynamics.

Looking at channel-mixing models in Figure 5.5, the results are more mixed. We explored two settings of the top-p parameter, and this has a large effect on the quality of the PSD of the generated data. Even slight modifications (e.g., 0.72 vs.



(a) Data

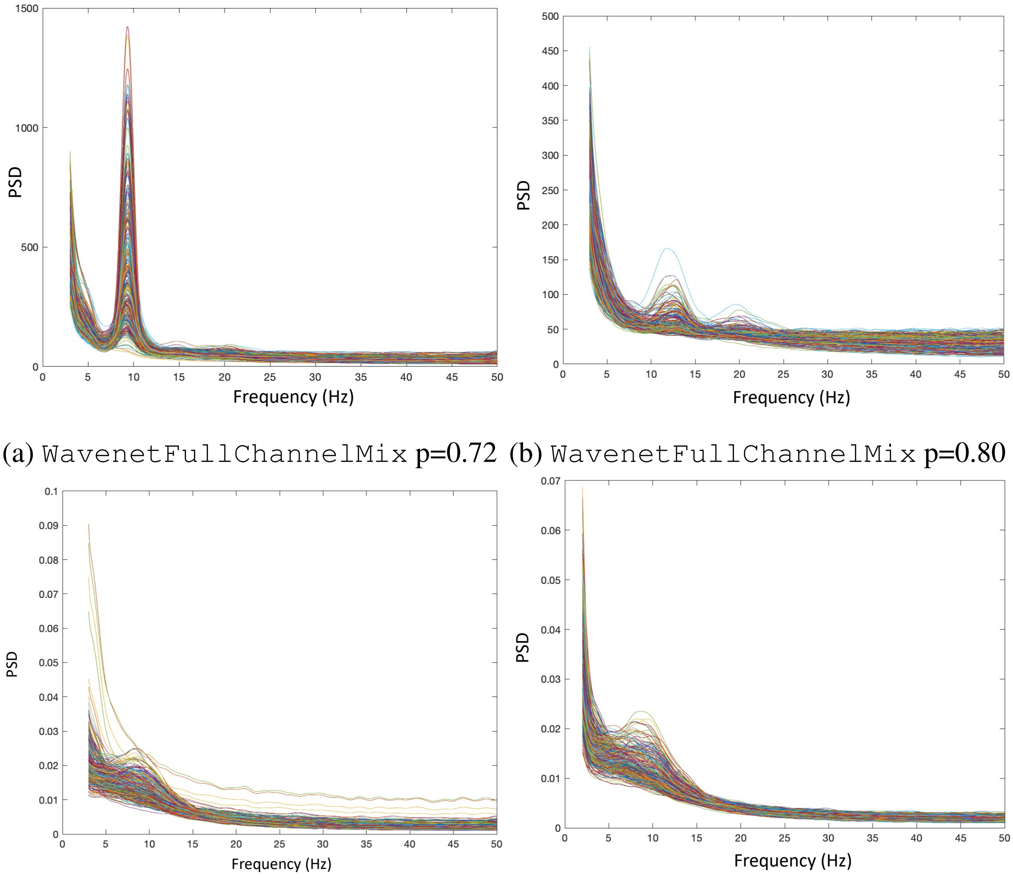
(b) AR(255)

(c) `WavenetFullChannel`

(d) `ChannelGPT2`

Figure 5.4: Comparison of generated data power spectral density (PSD) across channel-independent models. Each line represents a different MEG channel.

0.8 for `WavenetFullChannelMix`) result in large differences in the frequency of the two peaks, and also the width of the peaks. This highlights the sensitivity of these models to sampling hyperparameters. Ultimately both top-p values provide subpar PSD's compared to channel-independent models, likely due to overfitting as channel-mixing models lack the implicit regularisation of modelling each channel separately. For `FlatGPT2` the situation is even worse as the PSD looks much noisier and the frequency of the peaks does not match the true data. As shown previously, the PSD can already be well captured by a linear (univariate) AR model.



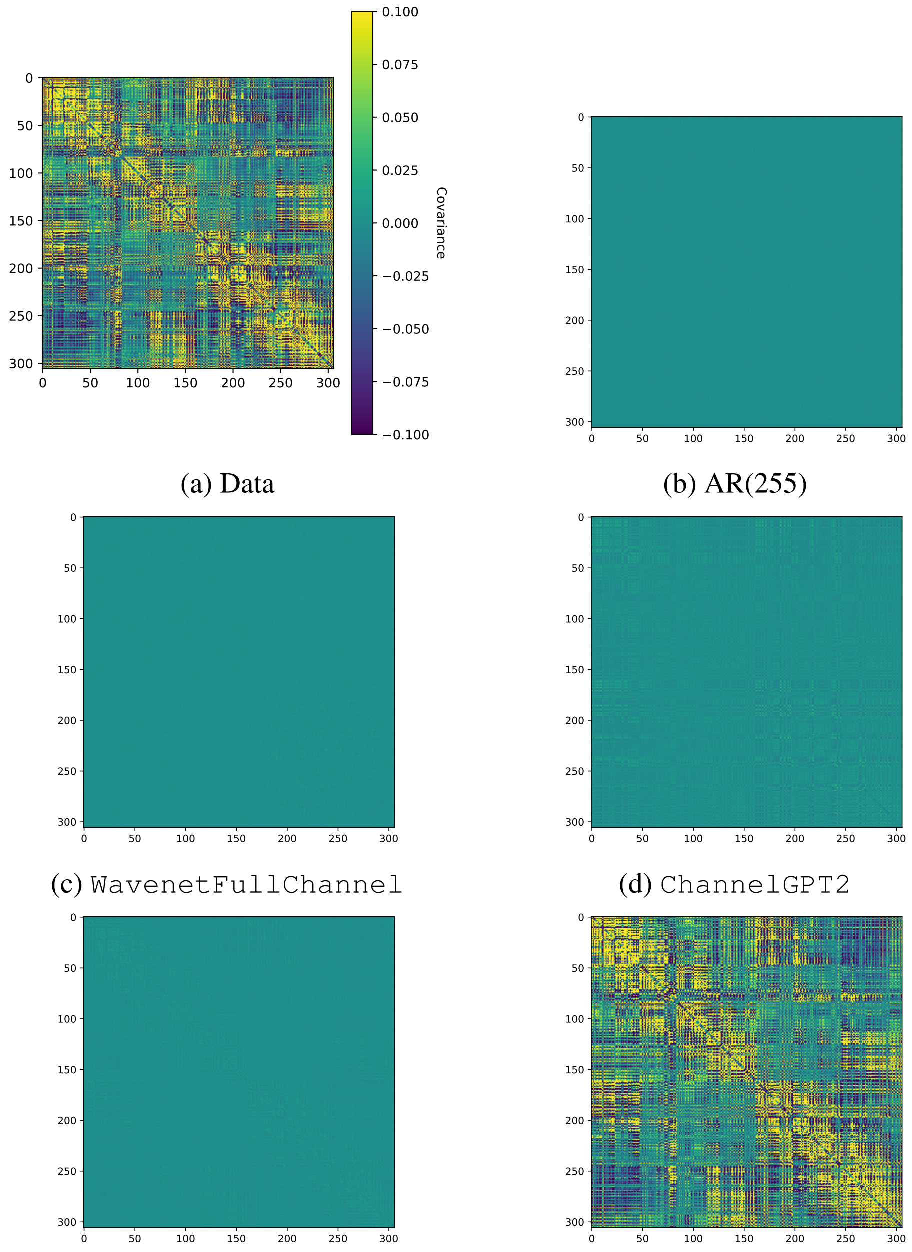
(a) `WavenetFullChannelMix` p=0.72 (b) `WavenetFullChannelMix` p=0.80

(c) `FlatGPT2` p=0.8 (d) `FlatGPT2` p=0.95

Figure 5.5: Comparison of generated data PSD across two channel-mixing models with varying top-p values. Each line represents a different MEG channel.

The added complexity of channel-mixing may introduce suboptimal loss minima where the PSD is not captured as well. Thus it is important to compare models on alternative measures where AR(255) might perform worse given its simplicity.

As the PSD is a channel-independent measure, we next looked at generated data covariance which captures the interactions between different channels (Figure 5.6). This reveals that the only model capable of closely matching the data covariance is `FlatGPT2`. All other models produce data with covariances much closer to 0. This is perhaps expected for channel-independent models



(a) Data

(b) AR(255)

(c) `WavenetFullChannel`

(d) `ChannelGPT2`

(e) `WavenetFullChannelMix`

(f) `FlatGPT2`

Figure 5.6: Covariance of generated data between channels (vertical and horizontal axes). All plots have the same scaling as (a).

which generate data independently for each channel, but somewhat surprising for `WavenetFullChannelMix`. Even though `FlatGPT2` may not produce accurate spectral data, by having information about other channels in the input it does an excellent job at capturing covariance. This highlights the trade-offs between different model architectures.

5.3.2 HMM statistics of generated data

Next, we looked at how well the generated data matches real data in terms of HMM statistics. HMMs are useful for unsupervised discovery of discrete states underlying timeseries data (Rabiner, 1989b; Vidaurre et al., 2018b). We fit a separate 12-state time-domain embedding HMM (TDE-HMM) to each multivariate generated timeseries (Vidaurre et al., 2018c). We used the osl-dynamics package (Gohil et al., 2023), and set the number of embeddings to 15, the PCA projection dimensionality of the channels to 80 and the sequence length to 2000. We trained the HMMs for 20 epochs with an initial learning rate of 0.02, and extract four different summary statistics from the inferred state timecourse. The distributions of these summary statistics over the 12 states across models are shown in Figure 5.7. Note that since a separate HMM is trained for each model, the states are not matched between models. Thus, we look at the distribution over states, rather than individual states.

Across the four summary statistics we can see that the real data has high variance in the distribution over states. AR(255) and `WavenetFullChannelMix` fail to produce data with variable state statistics, and even the mean over states is not captured well. `WavenetFullChannel` does a great job at capturing the mean of the state distributions, but still produces data with relatively invariant states. `ChannelGPT2` seems to best capture the distributions across all four statistics, especially for the mean interval and switching rate. This shows that Transformer-based models can generate data that better matches the HMM-inferred dynamics of real MEG data. Example state timecourses generated from all models are plotted

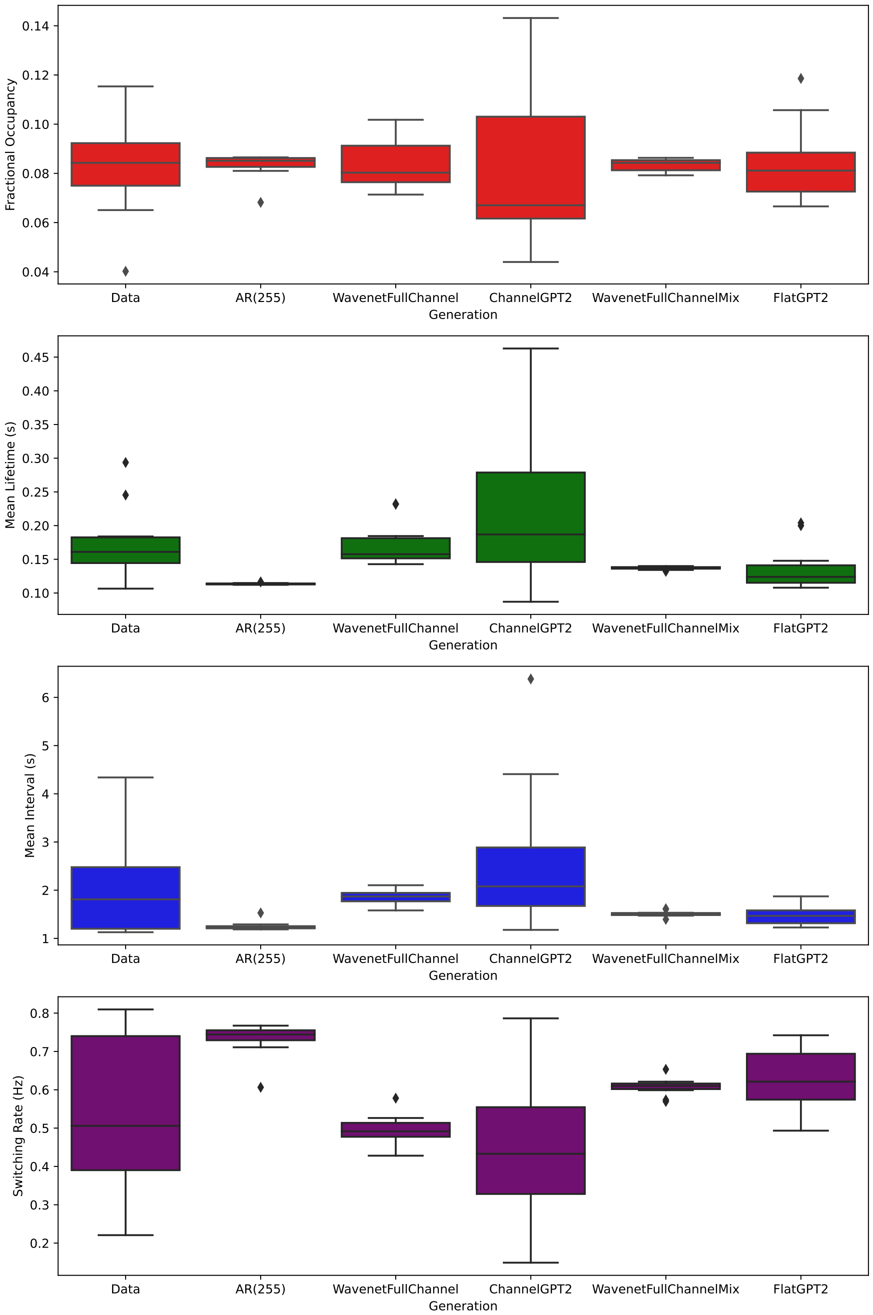


Figure 5.7: Distribution across states of 4 HMM statistics (rows) for each model and data (columns). The distribution mean and variance is computed across the individual values of the 12 states. Lower variance means states with more homogeneous statistics.

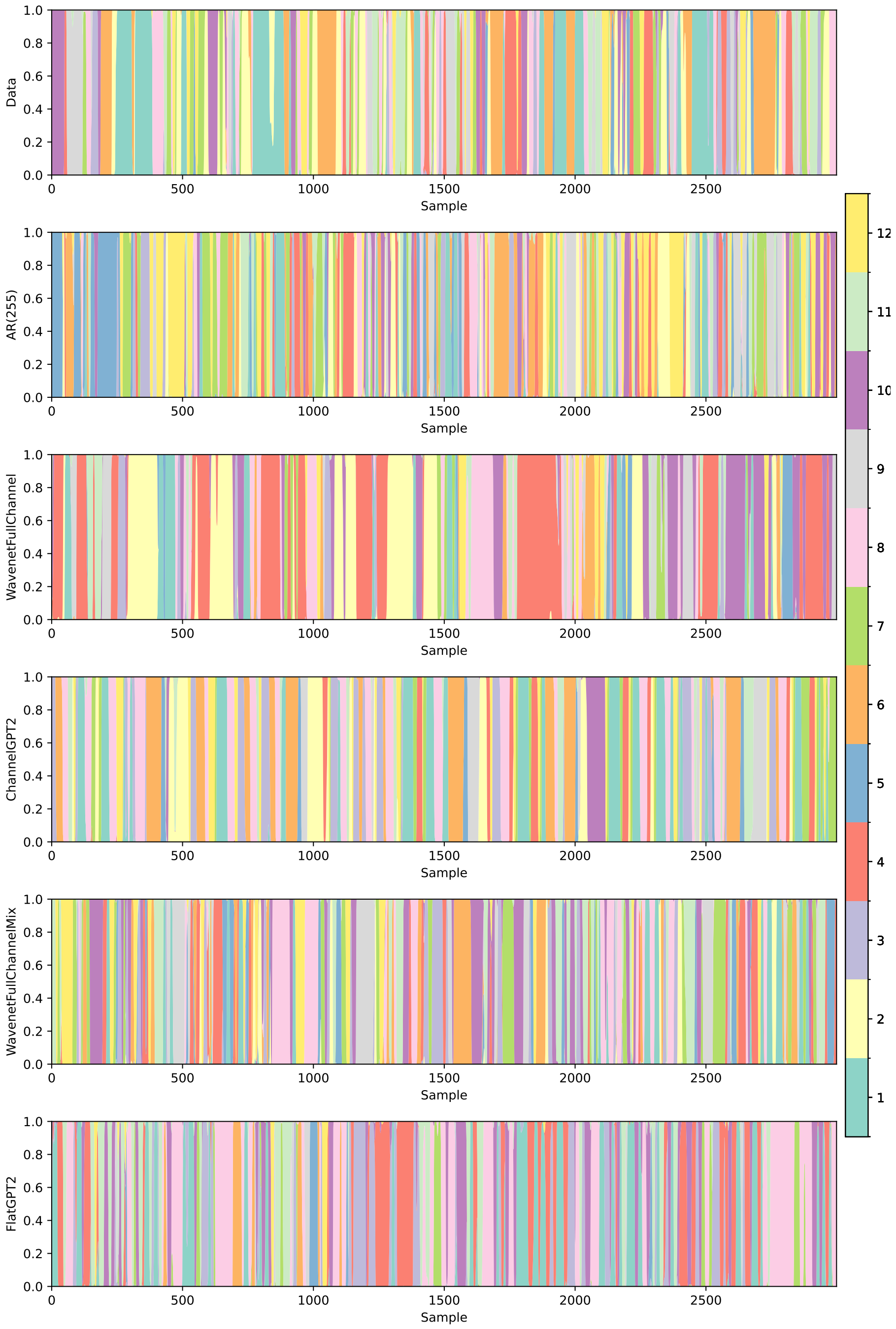


Figure 5.8: Example state timecourses from the HMMs trained on each model's generated data (rows). Each state is represented by a different colour. Note that state indices and timecourses are not matched across models.

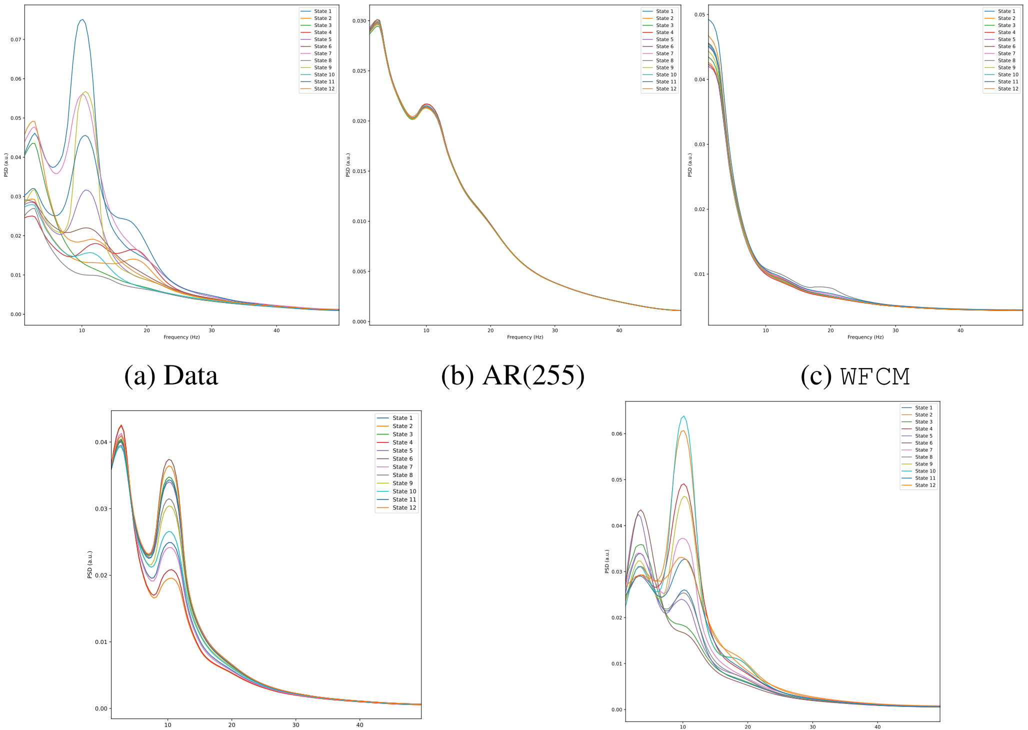
in Figure 5.8 to qualitatively illustrate the differences in the generated dynamics.

In addition to state statistics, we can also compute the power spectra of each state across the timeseries. In MEG data different states might capture oscillatory activity with specific frequencies. We plot the extracted power spectra from the inferred state time courses in Figure 5.9. We can see that the HMM trained on the MEG data contains many states that capture the 10 Hz peak, with fewer states having a 20 Hz peak. It is also clear that the states of the HMM fitted to the `WavenetFullChannelMix` generated timeseries do not contain these spectral peaks. While the AR(255) does contain states with a 10 Hz peak, the shape does not match the data well, and also states do not show the same variability as in real data.

In contrast `ChannelGPT2`, matches the state PSDs of the real data very well, further demonstrating the superiority of Transformer models in capturing complex neural dynamics. While `WavenetFullChannel` also improves substantially over the AR(255) power spectra, it falls short in capturing the 20 Hz peak and the heterogeneity between states observed in the real data and the generated data of `ChannelGPT2`. This and previous analyses show that the combination of channel-independence and a Transformer-based architecture are critical for matching the dynamics of real data.

5.3.3 Evoked analysis of generated data

The analyses in the previous section considered metrics for assessing the quality of an arbitrary generated timeseries, applicable to any M/EEG dataset. We can also leverage the experimental aspect of the Cichy et al. (2016) data and provide further focused insights on the task-related brain activity. As mentioned before, we used the task label timeseries from the training data when generating data with our



(a) Data

(b) AR(255)

(c) `WFCM`

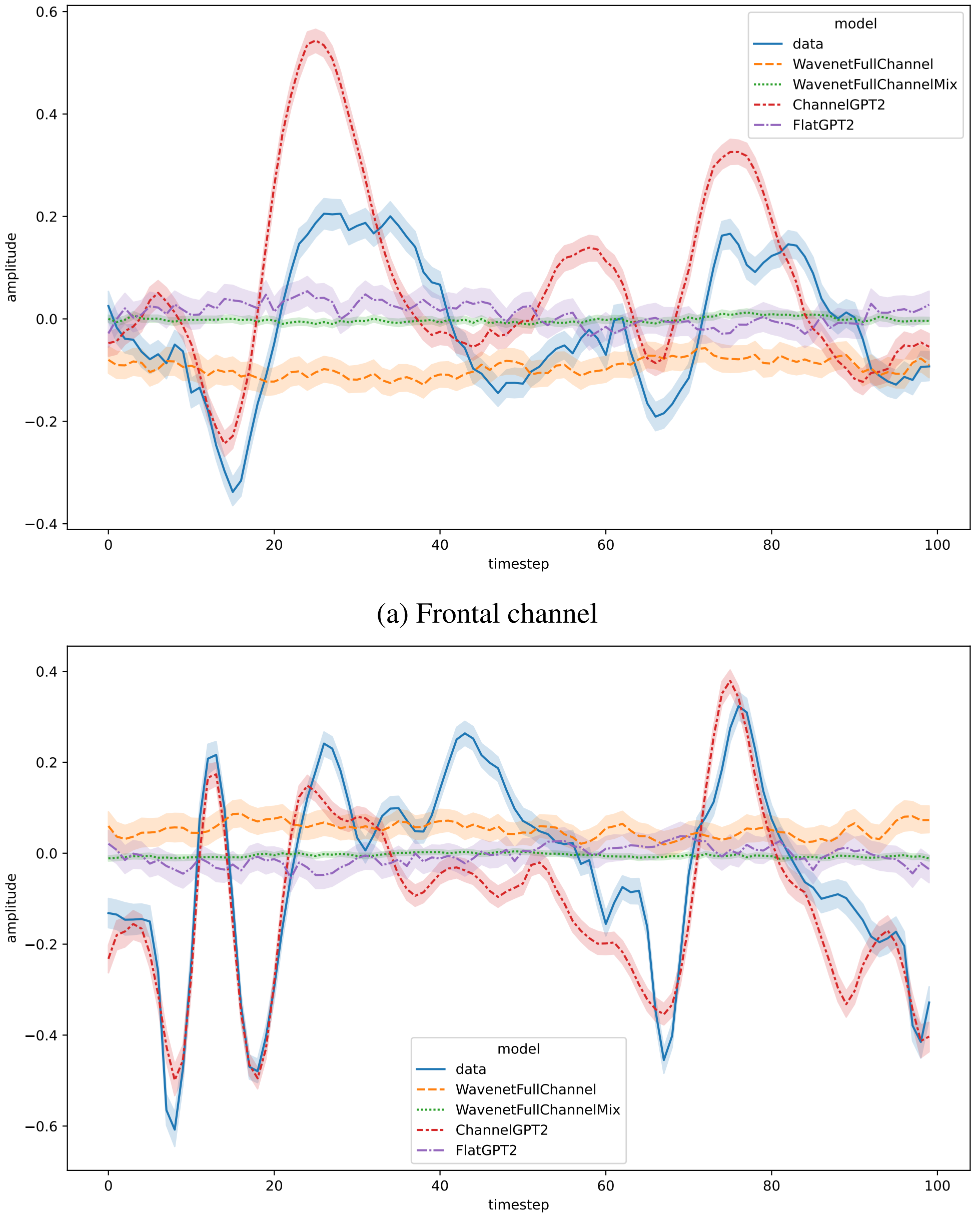
(d) `WavenetFullChannel`

(e) `ChannelGPT2`

Figure 5.9: Power spectral density of HMM states inferred on the generated data of each model. `WFCM` refers to `WavenetFullChannelMix`. Each line is the PSD of a different state. Note that states are not matched across models. Horizontal axis represents frequency in Hz. `FlatGPT2` is omitted due to failing to generate data with PSD matching real data.

models. If the models properly incorporate this conditioning, the generated data should reflect aligned task-related activity similar to real data.

By simple epoching of the generated timeseries based on the known task labels, we can compute evoked responses generated by our models. We do this for all models except AR(255) as it did not include task labels in its model. To compare the shape of average evoked responses, we average over all epochs in both real data and the generated timeseries. This results in data of shape $\bar{\mathbf{X}} \in \mathbb{R}^{C \times T}$ where $C = 306$ is the number of channels and $T = 1000$ ms is the trial/epoch length.



(a) Frontal channel

(b) Visual channel

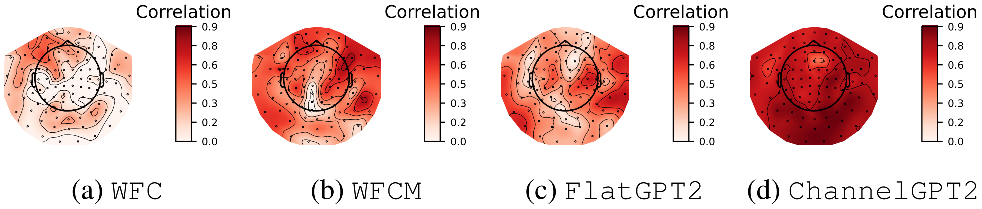
Figure 5.10: Comparison of evoked timecourses of 2 channels across our task-conditioned models. The whole x-axis encompasses 1 second. Timestep 0 is when stimulus presentation starts, and timestep 50 (500 ms) is when it stops. The peak occurring after 50 timesteps indicates a visual response to the stopping of the stimulus (removal of the image). Shading indicates variability across trials.

We visualise evoked responses across our models and the real data in a frontal and a visual channel in Figure 5.10. While both Wavenet models and `FlatGPT2` completely fail to capture the evoked timecourse, `ChannelGPT2` does a remarkably good job, especially in the visual channel. This is not surprising as the dataset is collected from a visual experiment, so most activity is visual. `ChannelGPT2` closely matches both the amplitude and the timing of the evoked response peaks across the whole 1-second epoch. Variability across trials is also well matched.

To quantify the similarity between real and generated evoked activity, we compute the correlation of the mean and variance (across individual epochs) of the evoked response for each channel separately. We plot the correlation values between the data and each model as a sensor space map, allowing insights into the spatial pattern of similarity. For the plots we average over magnetometers and gradiometers at the same location.

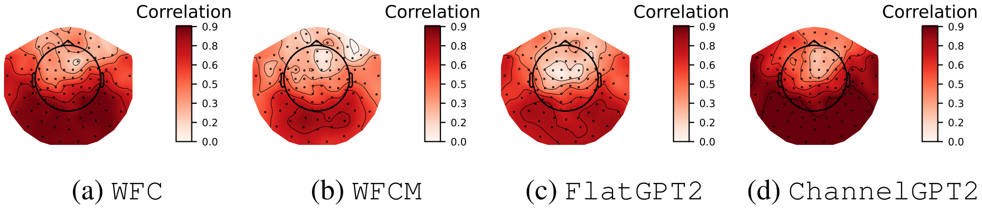
Figure 5.11 shows the correlation between the timecourses of the mean (across trials) evoked responses obtained from the actual data and the mean evoked responses obtained from data generated by each model. By computing the correlation between the timecourses of each channel we can plot these correlation values as a sensor space map. As expected, `ChannelGPT2` generates data with evoked responses that have much higher correlation with evoked responses from real data, and slightly higher correlation in visual areas compared to other channels, matching the known topography of visual evoked responses. In other models the correlation is low, and spatially better in frontal areas, likely because the evoked responses here are noisier providing an easier fit.

Figure 5.12 shows the correlation between the variance (over individual epochs) timecourses of the mean evoked response obtained from the actual data and the evoked responses obtained from data generated by each model. Again,



(a) `WFC` (b) `WFCM` (c) `FlatGPT2` (d) `ChannelGPT2`

Figure 5.11: Correlation between the timecourses of the mean (over individual epochs) evoked responses from the real data and mean evoked responses generated by each model. The correlation values are visualised across sensors. `WFC` refers to `WavenetFullChannel` and `WFCM` refers to `WavenetFullChannelMix`. Darker reds indicate higher correlation.



(a) `WFC` (b) `WFCM` (c) `FlatGPT2` (d) `ChannelGPT2`

Figure 5.12: Correlation between the timecourses of the variance (over individual epochs) of the mean evoked responses from the real data and the variance of the mean evoked responses generated by each model. The correlation values are visualised across sensors. `WFC` refers to `WavenetFullChannel` and `WFCM` refers to `WavenetFullChannelMix`. Darker reds indicate higher correlation.

`ChannelGPT2` generates data that has the highest correlations with the real data, with higher values in channels in the back of the head, appropriately capturing the topography of response variability. Other models have similar spatial distribution, and notably `WavenetFullChannel` also produces evoked responses with variance partially matching the real data.

Finally, a different way to assess task-related activity is to examine the evoked state timecourses from the HMMs fitted on the real and generated timeseries. Rather than looking at individual channels, this provides an overall view of which state gets activated when during individual trials. This is computed by simply epoching the state timecourse, and averaging over all trials. We plot these for the real data and

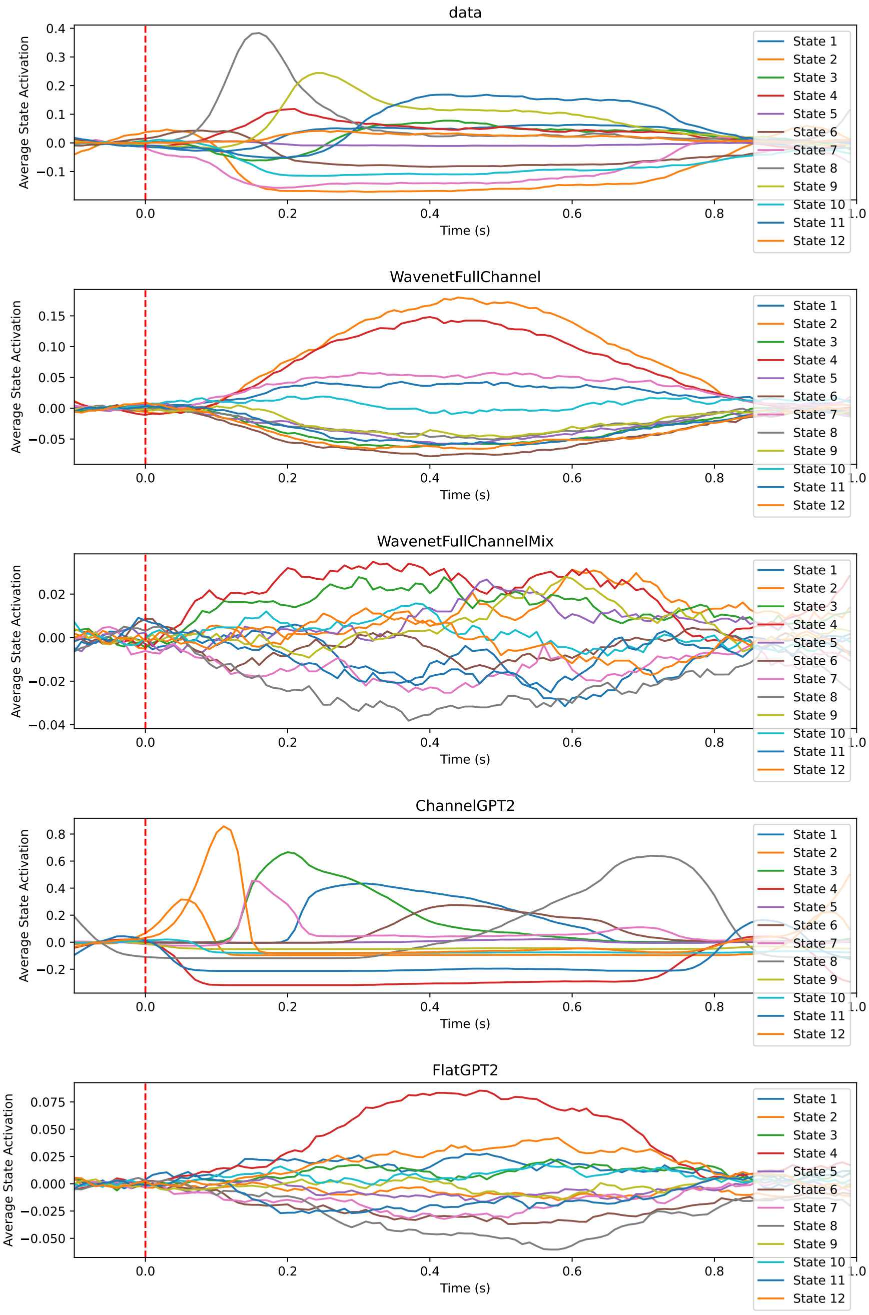


Figure 5.13: Evoked response state timecourses of HMMs trained on the MEG data and generated data from our task-conditioned models. Note that states are not matched between models. Image presentation starts at 0 seconds and ends at 0.5 seconds.

each generated timeseries in Figure 5.13. As expected, the HMM trained on models other than `ChannelGPT2` shows poor evoked state timecourses. `ChannelGPT2` generated data produces states with similar evoked dynamics and variability as the real data.

In summary, by leveraging the experimental nature of the MEG dataset, we evaluated how well different models generated task-evoked responses and dynamics. Across standard evoked analysis and discovered brain states, the Transformer-based `ChannelGPT2` model produced accurate task-related activity closely matching real MEG recordings. This further demonstrates its ability to generate physiologically grounded and experimentally relevant MEG timeseries. While we have not tested it directly for encoding by comparing individual trials with real data, our generation results show promise for encoding applications.

5.3.4 Group modelling

Up to this point, all trainings and analyses were done on MEG data from a single subject. We now look at whether adding more data improves modelling and generation. This is in line with the overall goal of training such foundational forecasting models on multiple large datasets. Here we take a first step in exploring this by scaling `ChanelGPT2` and `FlatGPT2` to the 15 subjects in the Cichy et al. (2016) data, and calling these `ChanelGPT2-group` and `FlatGPT2-group`, respectively. For adapting to multiple subjects and to capture variability over subjects, we used subject embeddings as described in the Methods.

We used the same hyperparameters as for the single-subject trainings, except for the following modifications. For `ChanelGPT2-group` we increased the embedding size to 240. For `FlatGPT2-group` we increased it to 480 and increased the number of layers and attention heads to 12. Dropout within the GPT2 model for `FlatGPT2-group` was set to 0.1. Both `ChanelGPT2-group` and `FlatGPT2-group` proved difficult to overfit, meaning that using more data acted as a regulariser, and we stopped training when validation losses did not improve for 5 consecutive epochs.

We were interested in whether evoked responses improve even further when using more data. To compare with the single-subject training we generated data using the subject embedding of that subject assuming that the model learned to condition its predictions on the subject labels. We compare the evoked response of single-subject and group models for one 1 visual channel in Figure 5.14. `FlatGPT2-group` failed to produce sensible evoked responses similar to the single-subject `FlatGPT2`. We found that generally `ChannelGPT2-group` produces evoked responses that are more smoothed than the single-subject model. We hypothesise this is partly because the model learns to generate data that is closer to the average statistics over subjects, and while it can adapt its generation based on the subject label, it is not perfect.

To test our hypothesis regarding `ChannelGPT2-group` generating more of an average across subjects, we generated data for all subjects (using appropriate subject embeddings) and compared the grand average evoked responses with those extracted from the MEG data of all subjects. Two channels are plotted in Figure 5.15. The evoked response averaged over all subjects is much noisier because of the high between-subject variability. However, we can see that indeed `ChannelGPT2-group` can generate this well, perhaps slightly smoother than the real data. Comparing these plots with Figure 5.14, it is also clear that it adapts its generation well to a specific subject compared to the group average.

A further way to test alignment between group-level evoked responses is to fit an HMM on the data of all subjects, and then infer state timecourses with this model

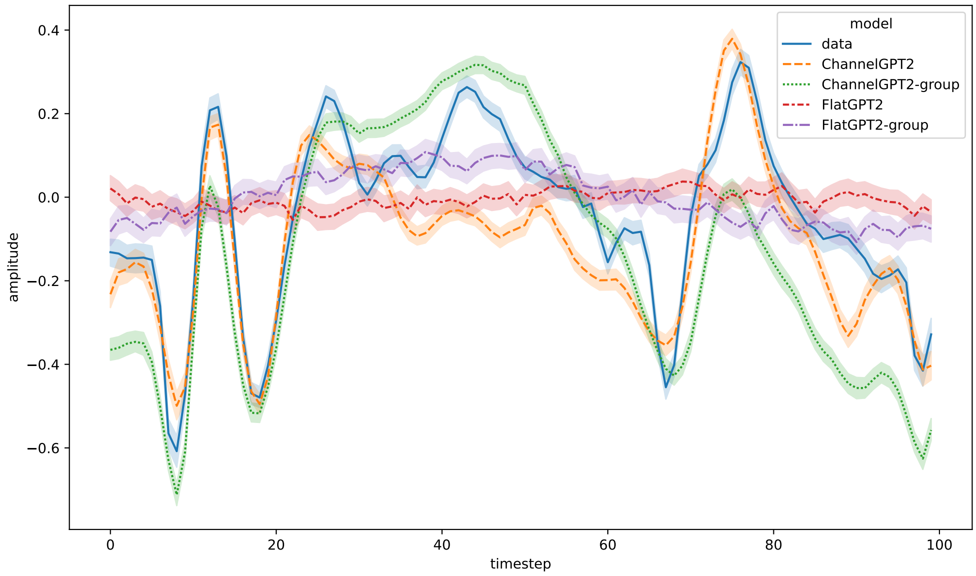
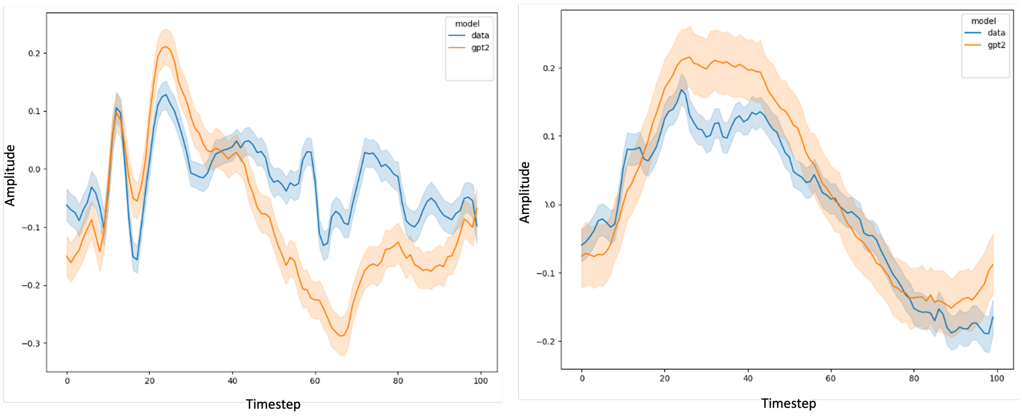


Figure 5.14: Comparison of evoked responses in a visual channel across single-subject and group models. The horizontal axis encompasses 1 second, where timestep 0 is the stimulus onset and timestep 50 is stimulus offset. Shading indicates 95% confidence interval across trials.



(a) Visual channel

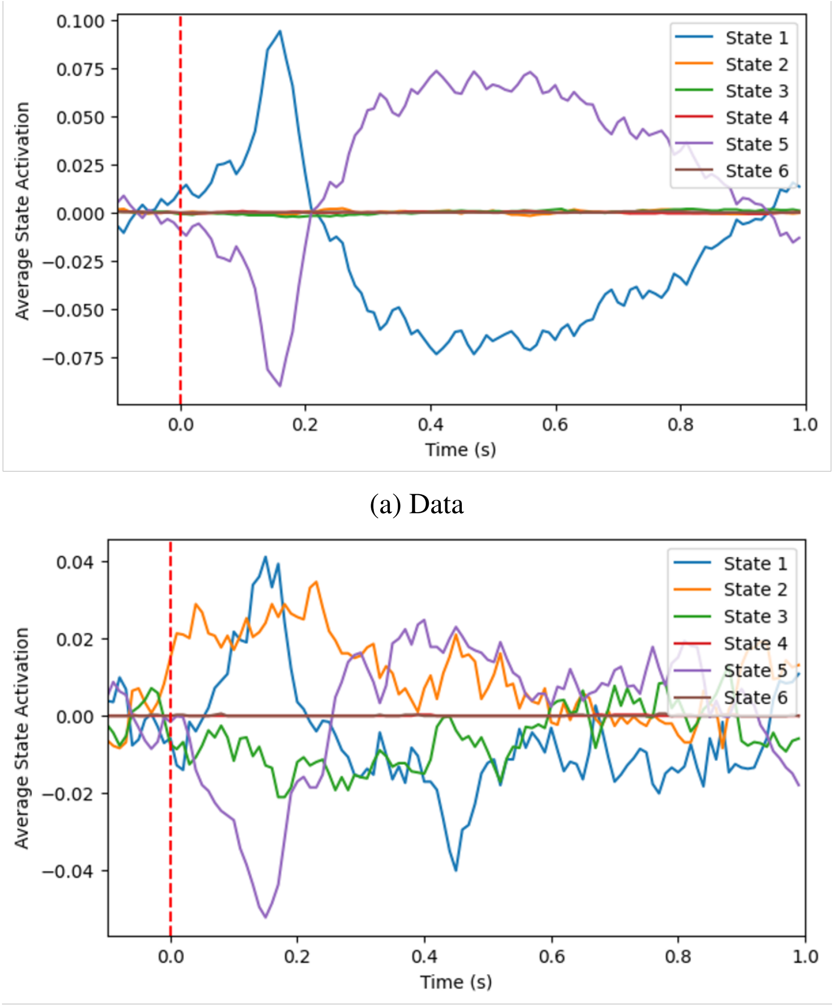
(b) Frontal channel

Figure 5.15: Comparison of evoked responses averaged across all subjects in the data (blue line) and the generated data from `ChannelGPT2-group` (orange line). The horizontal axis encompasses 1 second, where timestep 0 is the stimulus onset and timestep 50 is the stimulus offset. Shading indicates 95% confidence interval across trials.

on the generated data of all subjects from ChannelGPT2-group. By taking this approach we can directly match the evoked state timecourses between the real and generated timeseries. We trained an amplitude-envelope HMM (AE-HMM) with 6 states (Quinn et al., 2019) and show results in Figure 5.16. Two states that show strong activation during real task data show similar temporal signatures and amplitude changes in the generated data, albeit slightly noisier. In the generated data there are two additional states which seem to get activated during the trial. This indicates that while ChannelGPT2-group can capture some of the state-level dynamics, there is room for improvement.

Finally, we examine the variability in state time courses over individual trials. For this we trained an 8-state HMM on the real data of a single subject, and inferred the state timecourses on both the single-subject ChannelGPT2 and ChannelGPT2-group generated data, obtaining matched states. We hypothesised that even if the average evoked responses are similar to the real data, GPT2 may not able to generate trials with variability in the temporal activation of states. Figure 5.17 shows that this is indeed true for the single-subject ChannelGPT2 generated data. ChannelGPT2-group responses seem to include much higher temporal variability in state activations, though still falling short of the real data. This indicates that the model can capture some trial-to-trial variability through its exposure to multiple subjects, but has difficulty fully matching the complexity of real neural data. More data may be needed to improve this aspect of generation.

In summary, in the last few sections we showed that deep learning models, and in particular a channel-independent Transformer-based model can reproduce spatial, temporal, and spectral signatures of real data both at single-subject and group-levels. We were next interested whether such a model can aid in specific tasks, for example decoding of experimental conditions in the Cichy et al. (2016) data.



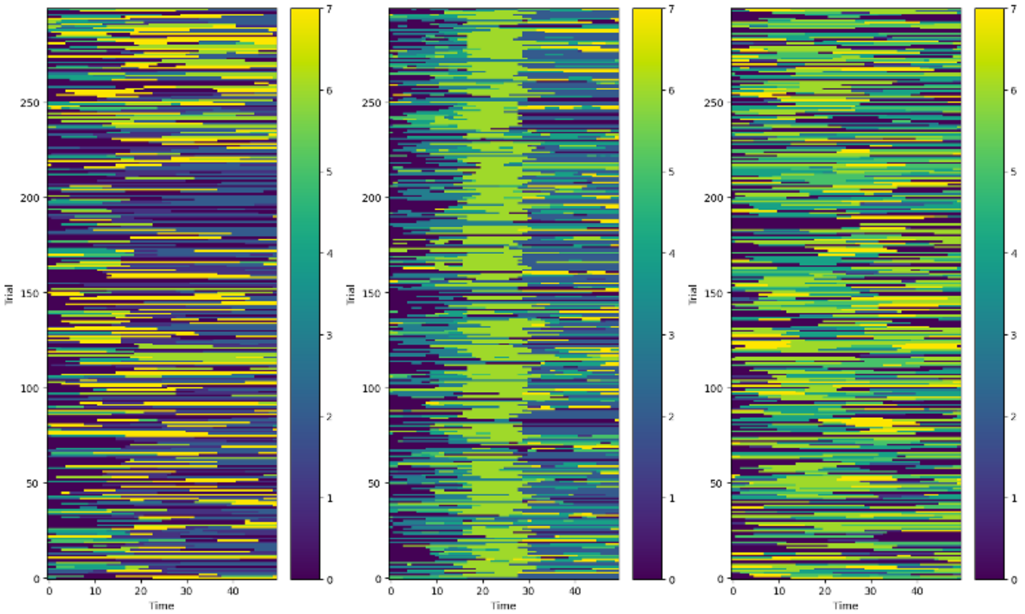
(a) Data

(b) `ChannelGPT2-group`

Figure 5.16: Comparison of evoked state timecourses inferred from the data of all subjects and from the generated data of `ChannelGPT2-group` for all subjects. State indices are matched between the two plots, as the same fitted HMM model was used.

5.3.5 Classification of generated data

While there are multiple ways a forecasting model could be used to aid decoding of task labels, here we opted for two approaches, leaving more complicated methods to future work. We first investigated whether the task responses produced by



(a) Data (b) `ChannelGPT2` (c) `ChannelGPT2-group`

Figure 5.17: Comparison of trial-level variability in the evoked state time-courses of an HMM trained on real data and applied to the generated data of `ChannelGPT2` and `ChannelGPT2-group`. Different colours represent different states (matched across models). Individual trials however are not matched and we cannot compare the plots at the trial-level, only as an aggregate visualisation of variability across trials.

`ChannelGPT2` can be classified with performance comparable to trials of real data. This also further tests how well the model captures spatiotemporal task-related activity and information. The benefit of this approach is that if similar performance is obtained, then `ChannelGPT2` could simulate an arbitrarily large number of trials to potentially improve decoding of real data through pretraining on this simulated data. This is a form of transfer learning where the decoding model, not the forecasting model, is transferred.

We generated 20 trials for all 118 conditions for 1 subject with both `ChannelGPT2` and `ChannelGPT2-group`. We trained separate linear neural network models described in Chapter 3 on the real data (20 trials/condition) and the generated datasets, with an appropriate 4:1 train and validation set ratio. This achieved 17.6%, 1.9%, and 7.2% validation accuracy for the real data, `ChannelGPT2`, and `ChannelGPT2-group`, respectively. Thus the group model generates more classifiable subject-specific task-responses, but still does not reach the classifiability of real data. This and previous analyses indicate the group model successfully leverages larger datasets to produce more accurate task-related activity.

5.3.6 Transfer learning

A key advantage of generated data is the ability to generate virtually infinite amounts. We generated additional datasets with 40 and 60 trials/condition using `ChannelGPT2-group`. Training a decoder on these achieved 21.7% and 44.2% accuracy, respectively, exhibiting linear scaling of classification performance with simulated data amount. Critically, we assessed whether this simulated data can pretrain classifiers for transfer learning. We first pre-trained the neural network decoder on the 20-, 40-, and 60-trial generated datasets, then finetuned it (trained it further) on the MEG data (20 trials/condition). As the simulated data used for pre-training increased, accuracy of the finetuned model improved rapidly. Zeroshot (no finetuning) performance on real data was above chance with 2%, 3%, and 4% accuracy, for increasing pretraining data quantities. Final accuracies after finetuning were 19.5%, 21.5%, and 23%, respectively. Thus, each additional 20 simulated trials/condition improved final decoding by ~2%. These results are summarised in Table 5.1.

Finally, we also tried obtaining a decoding model directly from the `ChannelGPT2-group` forecasting model using Bayes' theorem, as described in Section 5.2.1. We found limited 5% accuracy over 1 subject's validation set (versus 40-50% with a discriminative decoder). This generative decoding approach may require larger datasets or

Trained on (no. trials)	**Tested on MEG (20)**	**Tested on GPT2 (same no. trial data)**
MEG (20)	17.6	-
GPT2 (20)	2	7.2
GPT2 (40)	3	21.7
GPT2 (60)	4	44.2
GPT2 (20) + MEG (20)	19.5	-
GPT2 (40) + MEG (20)	21.5	-
GPT2 (60) + MEG (20)	23	-

Table 5.1: Summary of transfer learning results. The first column shows the data used for training the decoder, with the number of trials per condition shown inside the parenthesis. GPT2 refers to the `ChannelGPT2-group` generated data, while GPT (.) + MEG (20) is the fine-tuned decoder on the MEG data. The other two columns represent the validation data on which the decoder performance is shown. Accuracy values are provided in percentages. Chance level is $100/118$.

more sophisticated architectures.

In summary, while generated data did not match real data in decodability when the number of trials was matched, the group model produced classifiable responses capturing key features, improving substantially over a single-subject model. Further, its simulated responses could improve decoding of real data through pretraining, demonstrating the utility of forecasting models for transfer learning. There is clear promise in scaling up simulated datasets to improve MEG decoding.

5.3.7 Ablation experiments

Ablation studies are a common approach in machine learning to understand model behaviour by selectively removing or altering components of the model (Meyes et al., 2019). We performed ablation experiments with `ChannelGPT2` to investigate how well it can generate task-related brain activity under varied conditions without further training.

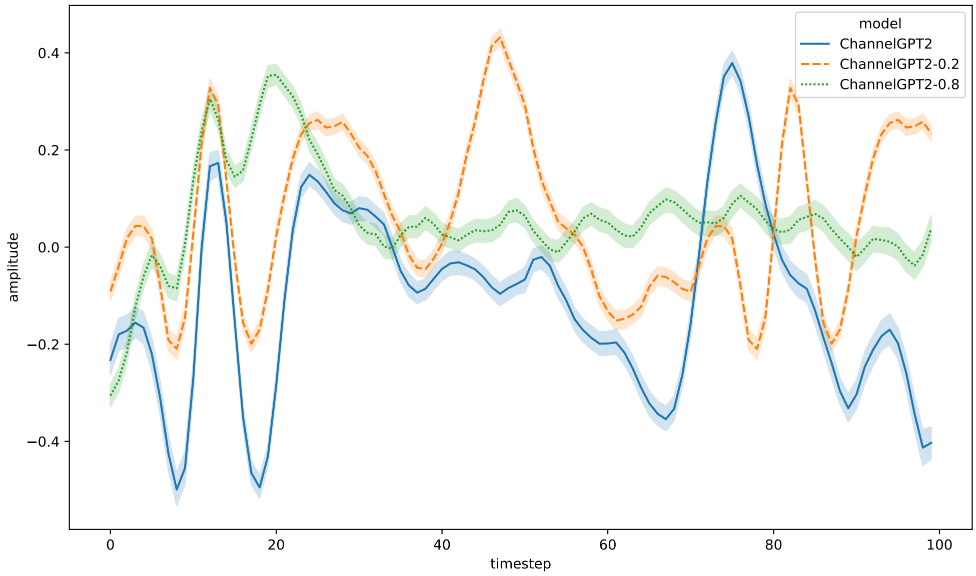
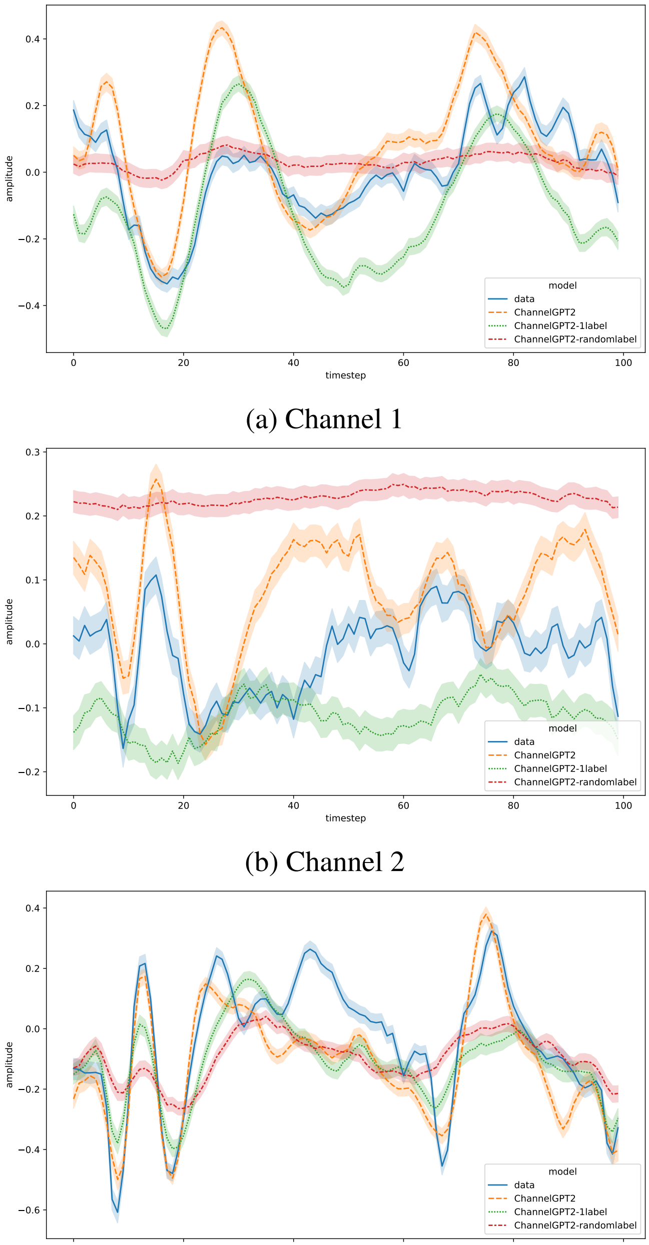


Figure 5.18: Evoked responses generated by `ChannelGPT2` for trials of 0.2 s (orange), 0.5 s (blue), and 0.8 s (green). The model was trained only on data containing trials of 0.5 s but adapts appropriately to the different durations.

First, we evaluated the model's ability to adapt to different trial durations. The original `ChannelGPT2` was trained on trials lasting 0.5 seconds. We generated data using the same model but with trial durations of 0.2 s and 0.8 s. As shown in Figure 5.18, `ChannelGPT2` accurately adapted to the shorter and longer trials. The evoked responses matched the expected timecourses, with appropriate truncation or lack of second peaks due to stimulus offset. This demonstrates the model's ability to generalise to varied trial durations despite being trained on a fixed duration.

Next, we performed two experiments to determine whether `ChannelGPT2` relies solely on timing information or also utilises the semantic content of the condition labels. First, we trained a model (`ChannelGPT2-randomlabel`) where the condition labels were shuffled randomly during training, breaking the semantic alignment between labels and evoked responses. Second, we trained a model



(a) Channel 1

(b) Channel 2

(c) Channel 3

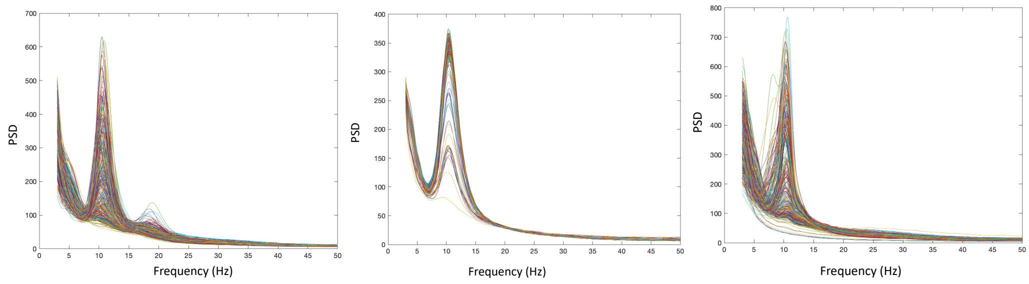
Figure 5.19: Evoked responses for models trained with shuffled or single condition labels, indicating reliance on semantic content. Three representative channels are presented. See main text for an explanation of model types. Timestep 0 is the stimulus onset and timestep 50 is the stimulus offset.

(ChannelGPT2-1label) using a single condition label for all trials. This tests whether the model cheats by learning an average evoked response instead of adapting to each condition.

As evident in Figure 5.19, both models failed to generate distinct evoked responses for different semantic conditions. This demonstrates that ChannelGPT2 leverages both timing and semantic information in the conditioning labels, rather than simply learning a stereotyped temporal template. Quantitatively, evoked response correlation with real data dropped to 44% and 56% for ChannelGPT2-randomlabel and ChannelGPT2-1label, respectively, compared to 74% for the full ChannelGPT2. Both the visual analysis and the correlation numbers indicate that ChannelGPT2-1label was somewhat closer to matching ChannelGPT2.

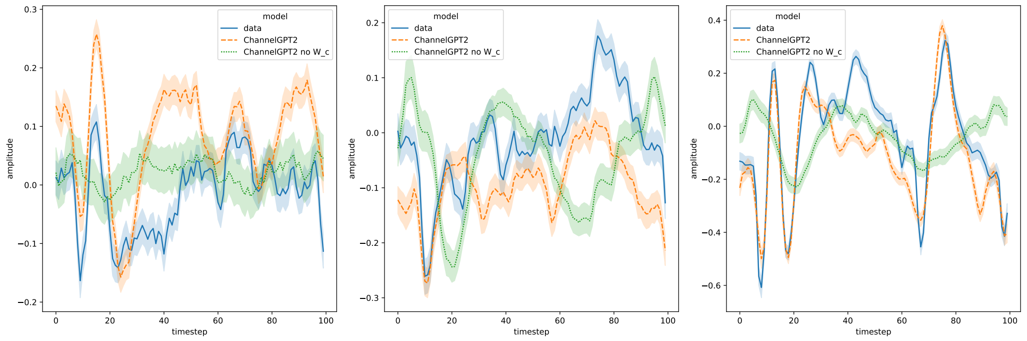
We also investigated the contributions of the channel and condition embeddings, by training two separate ablated models. As shown in Figure 5.20, removing the channel embeddings resulted in very similar PSD across channels in the generated data, indicating the model relies heavily on these embeddings to adapt generation per channel. The evoked responses in Figure 5.21 confirm that without channel embeddings, variability between channels is reduced. Removing the condition embeddings resulted in noisier power spectra of the generated data and no 20 Hz peak.

Finally, we found that the channel embeddings encode spatial relationships, as sensors that are near to each other in the real sensor montage tend to have more similar embeddings. This is shown through a t-SNE and PCA projection of the embedding space in Figure 5.22. Correlation between pairwise Euclidean distances of channels in physical space and embedding space was 0.45 (see Figure C.13 in the Appendix).



(a) `ChannelGPT2` (b) `ChannelGPT2` no $\mathbf{W}_c$ (c) `ChannelGPT2` no $\mathbf{W}_y$

Figure 5.20: Generated power spectra for full model (left) versus ablations. Both channel (middle) and condition embeddings (right) are critical for accurate spectral content.

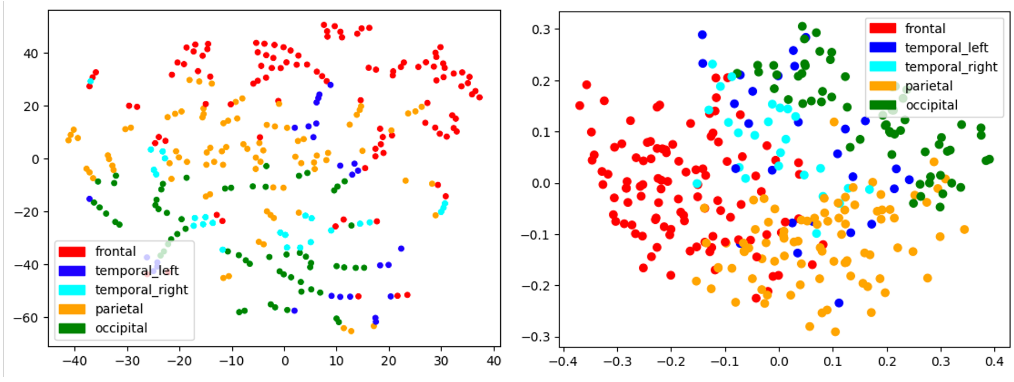


(a) Channel 1 (b) Channel 2 (c) Channel 3

Figure 5.21: Comparison of generated evoked responses from `ChannelGPT2` and the model with ablated channel embeddings (`ChannelGPT2` no W_c) across 3 representative channels. Without channel embeddings the model fails to adapt evoked responses to different channels. Timestep 0 is the stimulus onset and timestep 50 is the stimulus offset.

5.4 Discussion

In this chapter, we have presented our initial efforts at developing a general forecasting model for M/EEG data. After carefully evaluating the trade-offs between various modelling approaches, we settled on two main architectures: one based on Wavenet (van den Oord et al., 2016), and one based on GPT-2 (Radford et al., 2019). These models have proven successful in the audio and natural language



(a) t-SNE

(b) PCA

Figure 5.22: 2D projection of the channel embeddings from `ChannelGPT2-group` with t-SNE (left) and PCA (right). Channels are coloured by their location on the scalp grouped into 5 major brain areas.

domains, which share similarities with the time series nature of brain signals. We systematically compared different variants of our proposed models on both simulated and real M/EEG datasets.

We found that on real MEG data the forecasting performance was comparable between Wavenet and AR models according to next-timestep prediction metrics (results in Appendix). This suggests such metrics may be limited in their ability to effectively evaluate model dynamics beyond one-step prediction. Generated data analysis provided more discerning model comparisons. While the channel-independent AR and Wavenet models accurately reproduced the overall power spectral density, only the Transformer-based models captured more abstract multivariate statistics like inter-channel covariance and HMM state dynamics.

Critically, the `ChannelGPT2` model-generated data closely matched real MEG recordings across both temporal and spectral domains. Analysis of the discovered latent brain states showed `ChannelGPT2` reproduced variable oscillatory states similar to those inferred from human recordings (Vidaurre et al., 2018b). Each state captured distinct spectral content, while the linear and Wavenet-based models failed to achieve this degree of heterogeneity in their dynamics. It is possible that this does not indicate a failing of the Wavenet architecture, but rather that different conditioning methods may be needed. One such approach that we have not tested is using the same type of channel embeddings as for `ChannelGPT2`.

The Deep Recurrent Encoder (DRE) proposed by Chehab et al. (2022) is a highly relevant architecture to our approaches, as it demonstrates the advantages of modelling spatiotemporal dynamics for encoding neural data. DRE aims to predict MEG brain responses to visual word stimuli. Standard linear encoding models like temporal receptive fields (TRFs) face limitations in capturing the rich nonlinear dynamics, variability, and interactions inherent in MEG signals. DRE seeks to address these challenges by leveraging a convolutional LSTM architecture to model the intricate spatiotemporal neural dynamics across subjects.

While motivated as an encoding model, DRE can also be viewed through the lens of forecasting, with the addition of auxiliary task features. Forecasting holds inherent advantages over pure encoding, as it enables reconstructing real data and modelling complex spatiotemporal relationships, beyond just learning abstract representations.

Multiple analyses consistently demonstrated `ChannelGPT2`'s strengths in realistic conditional timeseries generation. `ChannelGPT2`-generated evoked responses had high correlation to real MEG data. However, modelling single-trial variability and between-subject differences remain challenging areas needing further work. Scaling to multiple subjects showed promise. The model was able to adapt its generated data based on the input subject label and generate task trials with variability more similar to real recordings than a single-subject model.

One consideration in our modelling approach is the use of sensor-space data as opposed to source space. Forecasting in the source space could more accurately capture the statistics of the data and inherently deal with between-subject differences. However, the measured raw data is in the sensor-space, and we wanted our models to receive this as input without any additional transformation (to source space). This delegates all modelling to the model itself. Similarly we let the model learn between-subject structure in a data-driven manner through the subject embeddings. We leave source space forecasting for future work.

Ablation studies quantified the importance of channel embeddings and task conditioning for accurate MEG modelling. Removing channel embeddings resulted in near identical generation across sensors, failing to capture spatial heterogeneity. Analysis of `ChannelGPT2`'s channel embeddings revealed spatial relationships between sensors were learned, with proximal channels having more similar embeddings. With incorrect or with no task labels, `ChannelGPT2` produced noisy evoked responses, indicating the model leverages both timing and label semantics. Furthermore, the model trained on 0.5s trials only, was able to produce reasonable responses to longer or shorter trials, showcasing generalisation. These results demonstrates the value of multi-faceted conditioning for realistic brain data modelling.

A key investigation involved analysing the classification of generated data according to the condition labels. The trials generated by the group-level model were classified with much higher accuracy (closer to real data) than those of the single-subject model. We also demonstrated that generated data can improve decoding of real trials via transfer learning (Torrey and Shavlik, 2010), with benefits scaling with generated data quantity. The classifiability of generated trials and transfer learning results highlight the utility of forecasting models like `ChannelGPT2` for decoding real MEG data. Further analysis could involve permutation feature importance of the decoding model trained on generated data to gain insights into learned representations. Transfer learning also requires more thorough evaluation across diverse decoding tasks. It would be important to also investigate other more direct finetuning or transfer learning approaches of the forecasting model akin to vision or language domains. These could involve additional output layers and losses for finetuning on downstream tasks.

Overall, the proposed analyses enable thorough interrogation of forecasting model dynamics beyond standard predictive metrics. However, experiments were limited to a single dataset, lacking evaluation across heterogeneous datasets and tasks. Testing on more diverse and larger-scale datasets with multiple recording systems and experimental paradigms is needed to fully validate transfer learning capabilities for forecasting, encoding and decoding. Applying the models to different modalities like EEG would also be informative of generalisation.

The full potential of self-supervised learning is only realised with large-scale data. This remains challenging for brain imaging compared to vision and language. Lowering barriers to data access and promoting data sharing is critical to realise the promise of foundation models in neuroimaging (Poldrack and Gorgolewski, 2014).

Investigation of the proposed models on simulated data could shed light on which model features are necessary for good modelling. In Appendix C.2.1 we provide some insights into how Wavenet models are better able to capture distinct oscillatory activity in simulated data compared to linear AR models.

A core limitation of the channel-independent GPT2 model is no direct leveraging of cross-channel information for each sensor prediction. Our `FlatGPT2` approach incorporating this performed worse. Different architectures or more data may enable proper utilisation of cross-channel dependencies. We tried various other approaches to mixing channel information beyond those reported, without success. For the Wavenet model, we incorporated all channels in the input by concatenating embeddings, and for the GPT2 models, we tried mixing channels with convolutions. We tried concatenating the output of each channel and then predicting from this shared output using a different projection for each channel. We also attempted to increase receptive field, dropout, and model size. One limitation in our approaches is the use of a next-timestep prediction loss. Future work should continue exploring architectures and different self-supervised or multi-timestep losses to leverage cross-channel information and improve modelling capabilities. In the context of BCIs, additional losses could be incorporated to predict mental states or decoding targets instead of future timesteps.

We did not analyse the inner representations of `ChannelGPT2` to explain its predictive abilities. Attention weight and activation visualisations could provide insights into important input features (Vig, 2019). PFI analysis may also illuminate influential temporal, spatial, and spectral input features for forecasting.

In conclusion, this work demonstrates that deep forecasting models can accurately reproduce complex neural dynamics of both ongoing and task-related activity and provides an extensive analysis methodology. Key limitations are small-scale experiments, the lack of working channel-mixing methods and multi-dataset testing. Future work should explore more flexible conditioning, study different self-supervised and transfer learning frameworks, and critically, apply similar analyses when scaling up across diverse, large electrophysiology datasets. This has the potential to enable powerful transfer learning and advance foundational brain modelling and decoding.

6 | Decoding thoughts

In the previous chapters, we have presented methodological advancements for dealing with various types of variability in M/EEG data. The development of these methods was inspired by the central thesis of improving the modelling and decoding of brain activity. Specifically, decoding in the context of brain-computer interfaces (BCI) aimed at communication is of particular interest. However, methodological advancements can only go so far, and we were also interested in tackling this challenge experimentally. A BCI is simply the real-time (online) application of a decoding algorithm to brain data that is being streamed continuously to the computer. Most often BCIs are aimed at decoding brain data into intents, actions, text, or speech. While in this work we have not built a real-time BCI, our data and offline methods are aimed at enabling such BCIs in the future.

For improving communication speed with BCIs, we posited two distinct solutions, applying decoding to inner speech, and improving decoding by collecting a large number of trials. We hope that our investigations will contribute to the field of neural speech prosthetics (Metzger et al., 2022), which one day may be capable of restoring communication to people with locked-in syndrome, a condition where people are unable to move or speak. To be clear this chapter is in the spirit of proof of concept with the aim to gather preliminary evidence as to the extent to which it is possible to decode inner speech noninvasively using electrophysiology, ideally self-generated inner speech, but otherwise elicited.

Despite the prevalence of inner speech in everyday life, research on this has been limited, particularly when it comes to non-invasive methods (Panachakel and Ramakrishnan, 2021). This chapter aims to fill this gap by using EEG and MEG to collect data from three different inner speech paradigms, and by conducting an initial decoding analysis. Specifically, we tested silent reading, repetitive inner speech, and generative inner speech tasks. We hypothesised that silent reading would yield the most decodable signals due to the visual presentation of words. While inner (covert) speech refers to the internal voice/monologue that most people possess and is a purely cognitive process, silent reading involves visual processing of the presented text and thus has additional sensory activity.

We collect a high number of inner speech trials from a few participants. Besides comparing across recording modalities we also compare across inner speech types. Our aim is to analyse the decodability of inner speech within each task and between tasks by the use of transfer learning. We find that in both EEG and MEG, silent reading can be decoded relatively well with 30-40% accuracy across 5 words. However, the decoding performance of both types of inner speech is mostly at chance level. This prohibited further transfer learning investigations between tasks. While the inner speech results are primarily negative, we believe our exploration of data size and various decoding methods is valuable. The dataset itself is useful for the research community as it contains a much larger number of trials within one participant than any other inner speech dataset. Having multiple sessions also allows for testing across-session performance.

Finally, we systematically compare silent reading decoding performance within 3 participants across four non-invasive modalities. These are EEG, 2 types of MEG machines, Elekta and CTF, and optically-pumped magnetometers (OPMs). We also compare the spatiotemporal dynamics of silent reading between these modalities. This is especially aimed at validating OPMs as a new kind of non-invasive brain recording technology. We find comparable performance to EEG, but OPM performance did not reach traditional MEG.

6.1 Introduction

Inner speech, also known as verbal thinking or covert self-talk, refers to the internal monologue that occurs within one's mind. This phenomenon has been extensively studied in psychology and cognitive science (Alderson-Day and Fernyhough, 2015). More recently, neuroimaging techniques have allowed researchers to examine the neural correlates of inner speech directly. As discussed by Geva (2018), early 20th century studies used measurements of tiny muscle movements during imagined speech production to infer inner speech. However, the advent of modern neuroimaging techniques like MEG and fMRI has enabled more direct investigation of the brain regions involved in inner speech compared to overt speech.

As mentioned in Chapter 1, traditional brain-computer interface (BCI) systems are relatively slow and do not leverage inner speech, which has the potential to enable communication at natural speech rates. Some progress has been made in decoding visual stimuli during reading limited sets of words or sentences (Mugler et al., 2014; Hultén et al., 2021; Moses et al., 2019), as well as decoding speech perception and overt speech production where muscle movements are present (Dash et al., 2020b; Défossez et al., 2022). However, detecting brain signals associated specifically with inner speech remains challenging given the lack of external stimuli or produced behaviour to provide timing information.

While we focus here on the hard problem of non-invasive inner speech decoding, more tractable approaches exist. These include applying invasive methods to inner speech, as well as applying non-invasive methods to related tasks that share features with inner speech. Related tasks either provide external timing information through stimuli like silent reading/listening, or leverage produced behaviour like overt loud or silent speech. However, it is unclear how insights from these tasks might transfer to decoding inner speech itself. Disentangling task-related activity (e.g. visual, auditory, or motor) from pure inner speech is difficult. Similarly, models tuned to detect muscle activation during silent speech may not transfer well to pure inner speech decoding (Dash et al., 2020b).

There are also experimental challenges inherent to studying inner speech. While people often experience spontaneous inner speech during mind wandering, there is no way to align such endogenous thought with recorded brain activity. This leaves two options: a purely unsupervised learning approach, or using more constrained, artificial experiments. The latter often employs visual or auditory cues to elicit inner speech in a time-locked manner. Some papers refer to cued silent reading tasks as a form of inner speech. However, we argue this confounds inner speech with concurrent visual processing of the words.

Pure inner speech paradigms can be achieved by using identical cues across different conditions. Brain responses to the cues themselves will be consistent, while activity varying between inner speech conditions can be disentangled. A limitation is that using identical cues provides no overt record of the specific condition of each trial. Participants can either be instructed on the inner speech to generate for each cue beforehand (*repetitive* inner speech), or report the contents after each trial (*generative* inner speech) (Parker Jones and Voets, 2021). Both approaches enable studying inner speech independently from external stimuli or behaviour.

The aims here are multifaceted, but guided by advancing non-invasive BCI communication. We approach this challenge through comparing:

1. Non-invasive recording modalities: EEG, MEG, OPM.
2. Types of inner speech: silent reading, repetitive, generative.
3. Data quantities: number of trials and sessions.

4. Decoding methods.

Our focus is on enabling BCI applications rather than basic cognitive neuroscience of inner speech per se. We review relevant research on decoding inner speech using invasive and non-invasive neural recording methods in Section 6.4. Next, we describe our experimental paradigms and analysis methods for investigating non-invasive inner speech decoding.

6.2 Methods

Our experimental paradigm follows previous efforts to delineate repetitive and generative inner speech. In a similar line of work Parker Jones and Voets (2021) investigate whether neural decoders trained on elicited inner speech data can be successfully transferred to decode self-generated inner speech (see Figure 6.1 for task paradigms). Using fMRI data from one subject, the authors trained deep neural networks on a large dataset of elicited inner speech collected during covert reading and repeating tasks. They then tested these models on new self-generated inner speech data collected while the subject freely imagined syllables. The transferred decoders predicted unseen phonemes with high accuracy. The successful zero-shot task transfer demonstrates the viability of leveraging elicited speech to train models that can decode self-generated inner speech. This has practical significance for developing inner speech brain-computer interfaces, since elicited tasks allow collection of labelled training data even from locked-in patients.

While previous studies have investigated various language units such as characters, phonemes, words, and even whole phrases, we focused our experiments on the word level. A limited set of words already has direct benefits for potential patients. Working with words also allows scaling up to cover the entire language lexicon in future research. Since we are interested in potential clinical applications, we

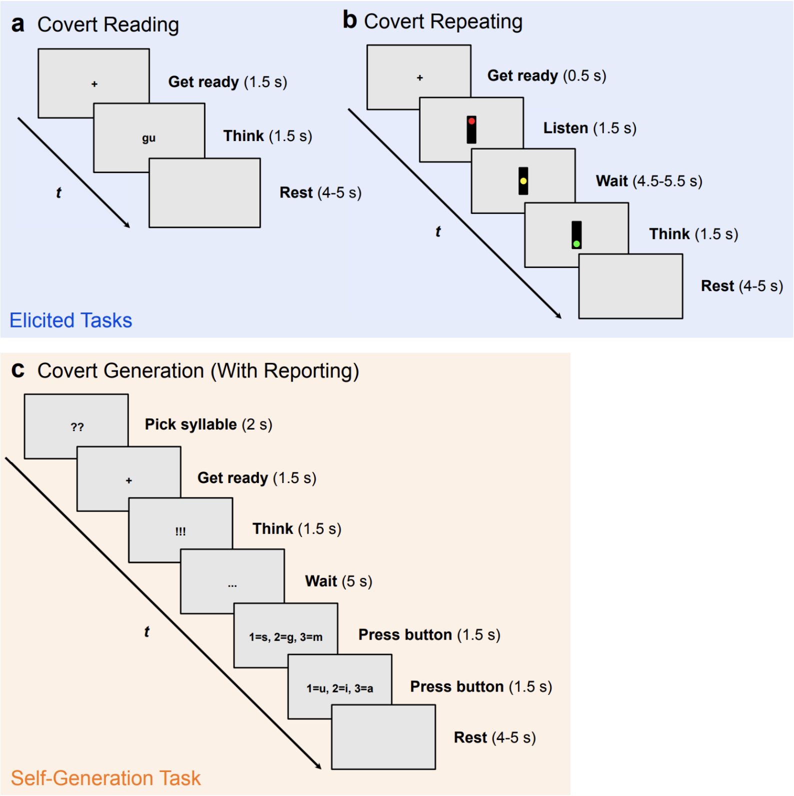


Figure 6.1: Visualisation of silent reading (a), repetitive (b) and generative inner speech (c) paradigms used in Parker Jones and Voets (2021). Figure from Parker Jones and Voets (2021).

chose a set of words that could be most useful for patients: help, hungry, tired, pain, thirsty. Decoding at the individual rather than group level is critical for practical applications. Thus we conducted our experiments with a small number of participants. Selecting only 5 words also allows us to test how collecting a large number of trials per word can improve decoding performance.

We collected data across two versions of the experimental paradigm, adapting it as we gained results to better align with our objectives. In version 1, we collected 1-second silent reading trials followed by 4 consecutive 1-second repetitive and generative inner speech trials. We collected a small MEG dataset, concentrating efforts on obtaining a high number of EEG sessions from the same participant to assess between-session transferability. In version 2 we omitted inner speech entirely and collected only silent reading data across 4 modalities - two MEG systems (Elekta and CTF[1]), EEG, and optically pumped magnetometers (OPMs). In version 1, we used an Elekta Neuromag Triux 306-channel system for MEG scans. We used a Neuroscan 64-channel cap for standalone EEG (standard 10-20 layout), with MEG-compatible Easycap EEG used for combined MEG and EEG.

For combined M/EEG, EEG ground and reference were on the left cheek and nose, respectively. For standalone EEG, the Cz and POz locations served as reference and ground, and we placed extra electrodes on the two mastoids. These could also serve as reference in offline analysis. Voltage and thus signal is always measured relative to a reference electrode in EEG. This means that the signal is the difference in voltage between the reference and other electrodes. Thus, the choice of reference greatly influences the characteristics of the recorded signal. It is best practice to place the reference electrode on the head but away from locations which might contain signals of interest. In our case both reference placements satisfy these criteria. While the signal shape and evoked responses will look different with different reference choices, this does not matter for decoding applications, as long as the signal of interest is not accidentally removed.

For most scans, we simultaneously collected electrooculogram (EOG) and electrocardiogram (ECG) data for easier artifact removal. ECG electrodes were placed on the wrists, with horizontal EOG on the outer side of the eyes and vertical EOG above and below the left eye. Electromyography (EMG) electrodes on the jaw monitored subtle mouth movements. Structural MRI scans were obtained for all participants in versions 1. During Elekta scans, we video recorded the mouth of

[1] https://www.nottingham.ac.uk/research/groups/spmic/facilities/ctf-meg-scanner.aspx

participants to ensure no task-related motion. Eye tracking was performed for all MEG and EEG scans. We collected 5 minutes of resting state data before and after each scan. Stimuli were delivered via PsychoPy. The experiment was reviewed and approved by the Medical Sciences interdivisional research ethics committee at the University of Oxford (reference number R75957/RE001).

6.2.1 Experimental paradigm

Version 1 In the first version of the experiment, participants silently read words displayed individually on a screen, followed by 4 consecutive visual fixation-cross cues to covertly repeat the word (Figure 6.2). This phase lasted approximately 5 minutes. In the next phase, participants continued reading and repeating words, but were now prompted after 0-2 read/repeat trials to imagine speaking a different word from the 5-word set (the generative inner speech task). 0-2 means that we randomly sampled either 0, 1, or 2 consecutive read and repeat trials. Similarly to the repetitive task we prompted the inner speech with 4 consecutive visual fixation-cross cues. Participants then indicated their imagined word with a button press. The random 0-2 read/repeat trials before each generative prompt ensured that participants did not pre-select words, better resembling unconstrained inner speech. We noticed that pure generative blocks let participants pre-plan words upon indicating the previous selection. Introducing random read/repeat trials limited this behaviour.

Each cross was displayed for 0.3 seconds followed by a 0.7 second-long blank screen. Word stimuli were displayed for 0.8-1.0 seconds, followed by a 0.8-1.0 second-long blank screen. Word order was randomised. The total second phase duration was approximately 50 minutes in 4 blocks with breaks between blocks. We collected simultaneous MEG and EEG data at the Oxford Centre for Human Brain Activity (OHBA). While we collected some combined MEG, we focused

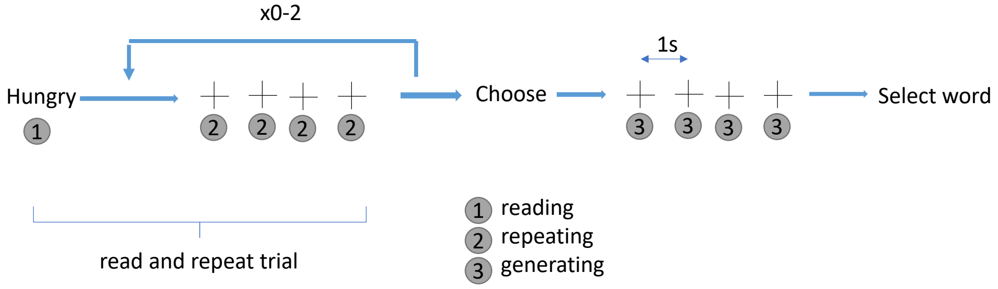


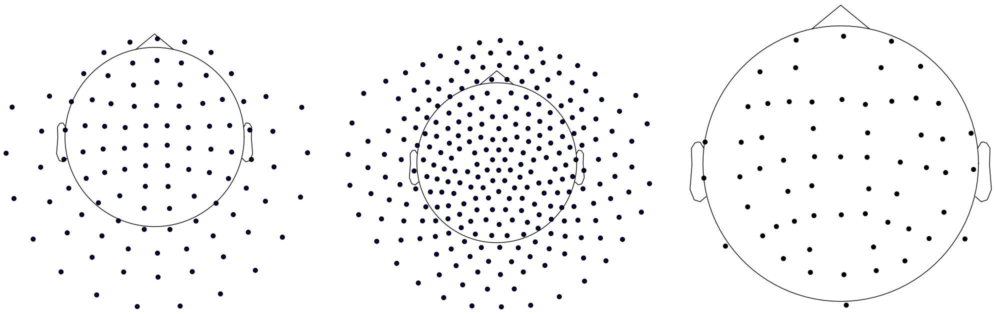
Figure 6.2: Paradigm for version 1 of our experiments. The participant silently *reads* ‘Hungry’, then *repeats* it four times at 1-second intervals cued by crosses. This can repeat 0-2 times before *generating* a new word from the 5-word set at four 1-second cross cues, avoiding the previous read/repeat word(s).

on obtaining multiple EEG-only sessions from one participant to assess between-session transferability.

Version 2 As the inner speech tasks yielded poor results, the second version focused solely on silent reading trials to collect more data within the 1-hour sessions. We expanded our aims and collected combined M/EEG at OHBA along with CTF-MEG and OPM data (same participants) in collaboration with the OPM Lab led by Matthew Brookes at the University of Nottingham. The simple paradigm displayed the 5 words randomly for 0.8-1.0 seconds with 0.8-1.0 second breaks. After every 10 trials, participants indicated the last word read to monitor attention.

6.2.2 Analysis

Data acquisition and preprocessing Elekta and EEG data were acquired at a sampling rate of 1000 Hz with a built-in bandpass filter between 0.03 and 330 Hz, while CTF and OPM scans were acquired at a sampling rate of 1200 Hz with a built-in lowpass filter of 600 Hz. The Elekta system contained 102 magnetometers and 204 planar gradiometers (102 sensors x 2 orientations), totalling 306 channels. CTF contained 265 axial gradiometers. Sensor configurations for three modalities



(a) Elekta (b) CTF (c) EEG

Figure 6.3: Sensor locations across scanning systems. CTF contained 1 gradiometer per location, while Elekta had 2 gradiometers and 1 magnetometer. Please note that OPM sensor layouts are reported in Section 6.3.1.

are illustrated in Figure 6.3. OPM data were recorded using a variable number of triaxial magnetometers (typically 150-180) in 60 fixed scalp locations, measuring magnetic fields along orthogonal axes. Channel configurations for individual participants are reported in Section 6.3.1.

Elekta data were preprocessed using Maxwell filtering for movement compensation and signal space separation using the MaxFilter algorithm (Taulu and Simola, 2006). Noisy OPM channels identified during recording were removed prior to analysis. For all systems, data were bandpass filtered (1-25 Hz typically, with higher lowpass for some experiments), notch filtered at 50 and 100 Hz, and subjected to automated bad channel detection (except OPM) using the oslpy package[2] (Quinn et al., 2022b). Bad segments were identified via a multi-pass procedure across progressively wider temporal windows (200 to 800 ms) with a significance threshold of 0.1. Independent component analysis was applied for dimensionality reduction (64 components for MEG/OPM, 32 for EEG). Components reflecting ocular or cardiac artifacts were removed before downsampling to 100 Hz and epoching.

[2] https://github.com/OHBA-analysis/oslpy

An additional mean field correction was applied to OPM data after preprocessing:

$$\mathbf{O} = \begin{pmatrix} \mathbf{O}_x \; \mathbf{O}_y \; \mathbf{O}_z \end{pmatrix} \tag{6.1}$$

$$\mathbf{M} = \mathbf{I} - \mathbf{O}\mathbf{O}^{\dagger} \tag{6.2}$$

$$\mathbf{X}_m = \mathbf{M}\mathbf{X} \tag{6.3}$$

Where $\mathbf{O}_x$, $\mathbf{O}_y$, $\mathbf{O}_z$ are sensor orientation vectors, $\mathbf{O}$ stacks them vertically, $\mathbf{I}$ is the identity matrix, and $\mathbf{M}$ is the final mixing matrix applied to the data $\mathbf{X}$. The aim of this transformation is to remove any spatially homogeneous field not coming from the brain and is described in detail in Tierney et al. (2021). Elekta data already includes such corrections in the MaxFilter algorithm, and CTFs do not need it as they measure the field gradient with gradiometers only.

Basic analyses involved comparison of evoked responses across channels, conditions, sessions, and modalities. Single-trial covariance matrices were also computed and visualised using t-SNE (Maaten and Hinton, 2008).

Decoding We employed several decoding methods for both the silent reading and inner speech tasks. A standard pipeline for silent reading involved our full-epoch LDA and neural network models from Chapter 3. For inner speech we tried several methods, but mainly used the covariance matrix over each trial as features for an LDA model. Channels were standardised prior to decoding. We tried concatenating across the 4 consecutive trials to form 4-second epochs, as well as averaging over trials, before decoding. Specific methods and hyperparameters are detailed in the Results.

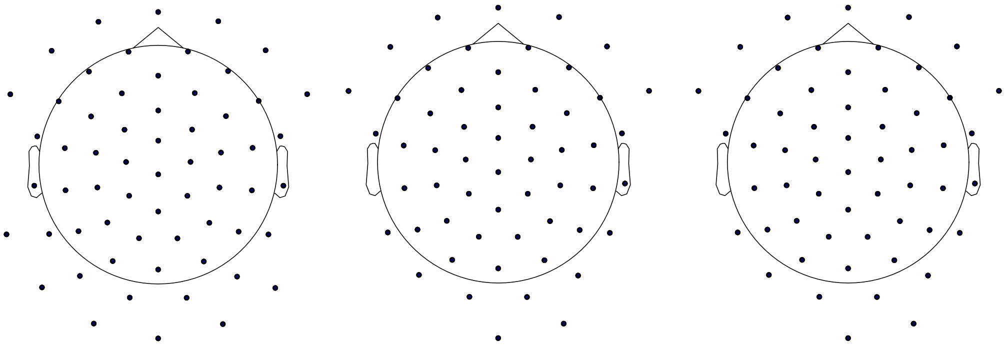
6.3 Results

6.3.1 Data statistics

A total of 4 male participants (P2, P4, P5, P6) between the ages of 20 and 40 participated across the two versions of our study. The participant pool included both native and non-native English speakers, though all had C2 level proficiency in English. Participant 4 (P4) is the author of this thesis and a non-native English speaker.

The two versions of the experiment were conducted with different goals. Version 1 involved evaluating the feasibility of decoding inner speech in 3 participants. Since EEG and MEG provided comparable decoding accuracy, 10 EEG sessions were collected from P4 in version 1 to examine improvements from increased data size and test decoder adaptability across sessions. In version 2, the inner speech task was removed and silent reading trials were collected using combined M/EEG, CTF, and OPMs across 3 participants. P4 and P5 also participated in version 1. The high number of silent reading trials (1250 per session) enabled thorough investigation of this paradigm. The OPM sensor layouts are shown in Figure 6.4.

Tables 6.1 and 6.2 summarise the number of sessions and trials for the participants in each version. While target numbers of trials are reported, minor variations occurred due to randomisation. The extensive datasets collected enabled thorough investigation of silent reading and inner speech paradigms. The multiple sessions also allow examination of between-session and between-modality variability. Overall, the dataset provides a unique resource to advance decoding of covert speech from non-invasive electrophysiological signals.



(a) P4 (b) P5 (c) P6

Figure 6.4: OPM sensor configurations across the three participants in version 2 of the experiment. Each location contained an OPM sensor measuring the magnetic field in three orthogonal directions. Sensor layouts and number of sensors are different due to technical difficulties with operating all sensors without overheating, excessive noise, or other issues.

	MEG	EEG
Participant 2 (P2)	1	
Participant 4 (P4)	1	10
Participant 5 (P5)	1	1
total silent reading trials	519	2076
total repetitive inner speech trials	2076	8304
total generative inner speech trials	1920	7680

Table 6.1: Number of sessions for each participant in version 1 of the study (top 3 rows). Total number of trials is given across all sessions and participants. Number of trials may be slightly lower or higher than shown due to randomness. Note that for P2 we conducted a combined M/EEG session while for the other participants MEG and EEG scans were separate.

6.3.2 Data analysis

In this section we present our non-decoding analyses of the collected data. The aim of this analysis is to validate data quality and uncover any insights into differences between tasks. These visualisations were primarily conducted on the 10 EEG sessions of P4, as this participant had the most inner speech trials (version 1 of

	combined M/EEG	CTF	OPM
Participant 4 (P4)	1	1	1
Participant 5 (P5)	1	1	1
Participant 6 (P6)	1	1	1
total silent reading trials	3750	3750	3750

Table 6.2: Number of sessions for each participant in version 2 of the study (top 3 rows). Total number of trials is given across all sessions and participants.

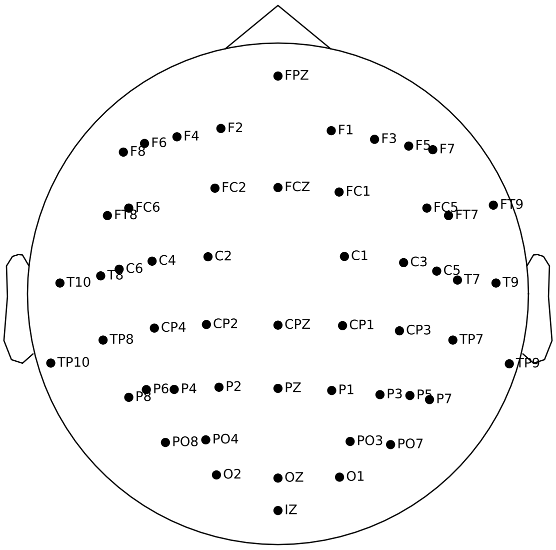


Figure 6.5: EEG electrode locations for P4 in version 1 of the experiment.

the study) and sessions, allowing investigations into between-session variability. For the visualisations in this section, no independent component analysis (ICA) artifact removal was performed on the data. We plot the electrode positions and their names over the scalp in Figure 6.5.

To visualise between-session variability, we compute the covariance across each 1-second inner speech trial, and average these across individual sessions. Figure 6.6 exhibits the averaged covariance of each session. Channels demonstrate high covariance within brain regions, such as the frontal and visual areas. Across

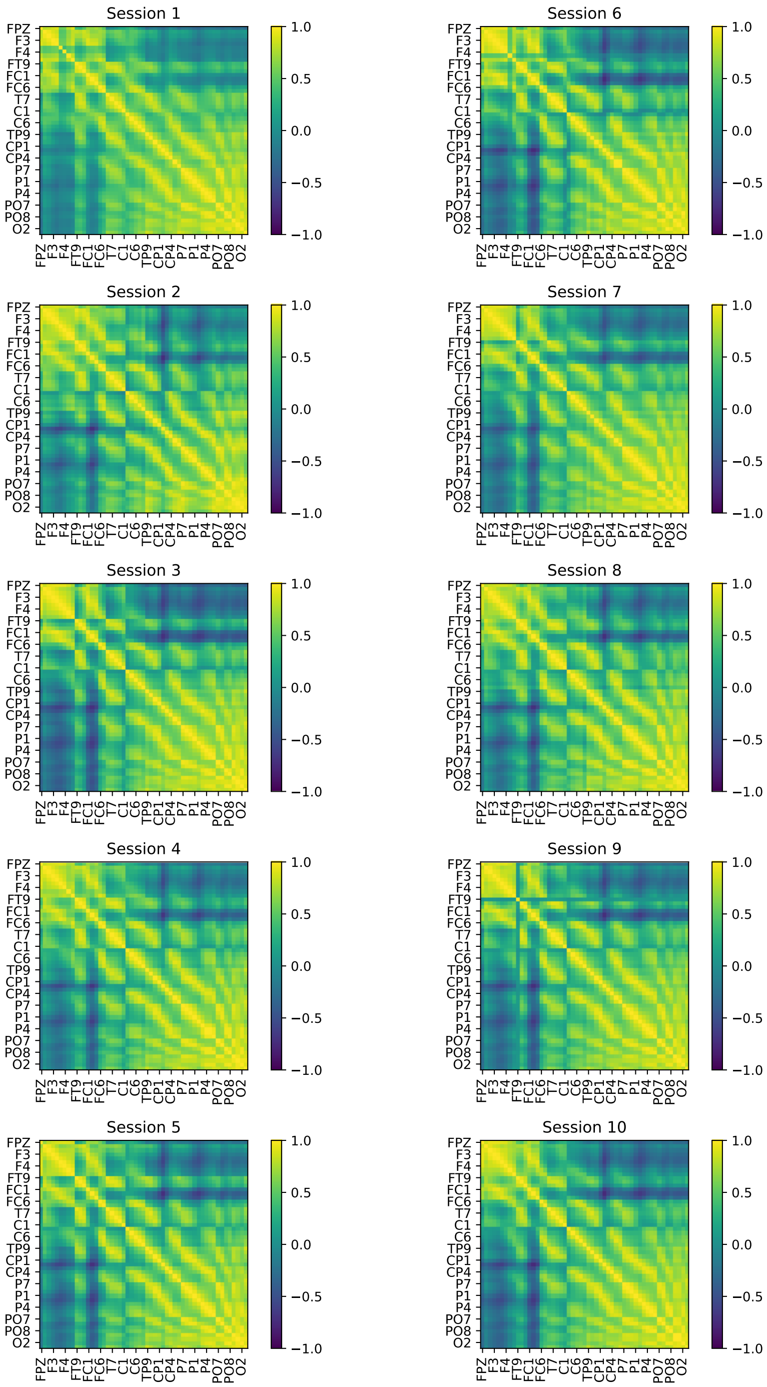


Figure 6.6: Averaged trial-covariances across the 10 EEG sessions of P4 in version 1 of the experiment. Each matrix represents a different session.

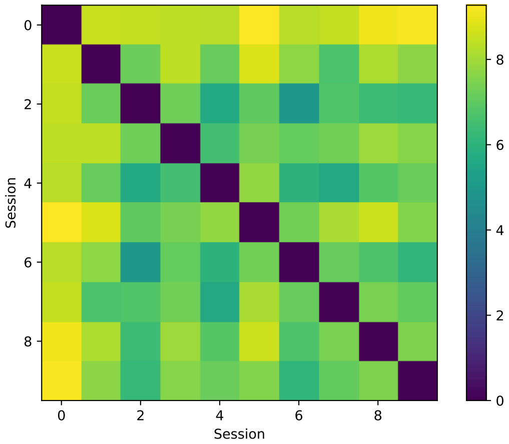


Figure 6.7: Riemann distance matrix between the average session-covariances across the 10 EEG sessions of P4 in version 1.

sessions, average covariances appear similar. To quantify similarity between sessions, we computed the Riemannian distances of the covariances between pairs of sessions for all possible pairs. This produces a session-by-session distance matrix (Figure 6.7). This can provide insight into between-session differences. For instance, the first session seems quite distant from the other sessions. This can mean that a decoder trained on other sessions may not perform very well on this session.

Finally, we investigated whether the covariance representations demonstrate interesting structure when visualised in 2D. To this end, we simply applied t-SNE to the individual trial covariances to project them into 2 dimensions and visualised the result. Figure 6.8 portrays this projection with two types of labelling. When trials are labelled by their corresponding condition (word), no apparent clustering emerges. This further bolsters that differentiating between words in inner speech is challenging. Structure can be discerned in the projection when labelled by session. This is anticipated since trials within one session are more similar to each other than to other sessions. These findings imply that decoding inner speech may be an

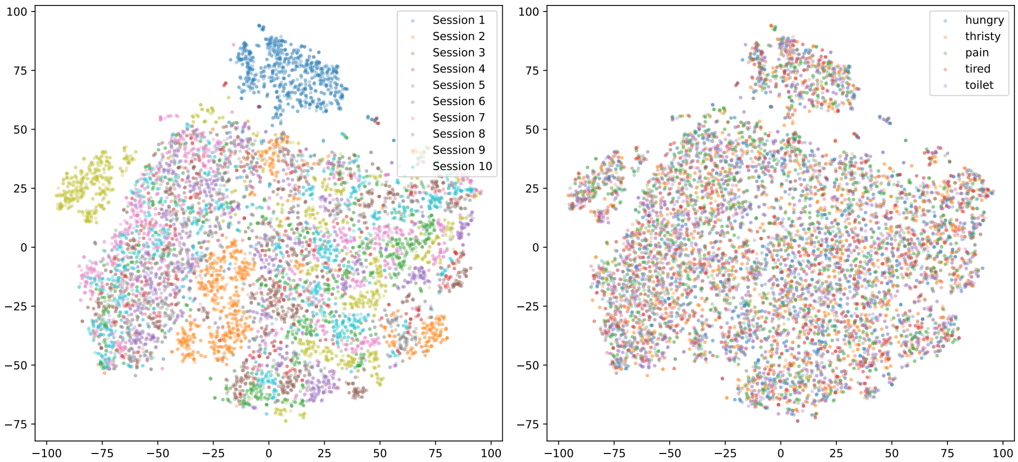


Figure 6.8: t-SNE projection of the per-trial covariances across the 10 EEG sessions of P4 in version 1. These are coloured according to the session label on the left, and according to the condition (word) on the right.

equally challenging endeavour. An investigation of evoked responses is provided in Appendix D.1.1.

6.3.3 Decoding inner speech in experiment version 1

While we collected reading data in all experiments, we will only present results from version 2, since that version was specifically aimed at analysing the silent reading task. In this section we will first analyse the MEG data, followed by results from the 10 EEG sessions of P4.

On the MEG data we attempted the methods from Chapter 3, such as sliding window LDA, full-epoch LDA, and the linear neural network. Running full-epoch models on the 1-second inner speech trials yielded chance-level results when decoding which word is being used in inner speech. We also attempted sliding-window decoding on a subset of MEG channels overlying the language area. However, the decoding accuracy timecourse exhibited substantial fluctuations and never exceeded 24%, where 20% is chance level. Thus, this is also a negative result.

On the EEG data of the generative inner speech trials of P4, LDA models (see Section 2.4.2) were trained on each session utilising the channel-covariance over the 1-second epoch as features with 5-fold cross-validation. For this analysis preprocessing involved a 1-40 Hz bandpass filter and no ICA artefact removal was employed. Before computing covariance, trials were normalised to unit variance and zero mean. We found above 25% validation accuracy in only 3 sessions (Figure 6.9), with chance level being 20%. However, when correcting for multiple comparisons none of the sessions had significantly better performance than chance. It may be that by running more cross-validation folds the performance in some sessions reaches significance. Decoding the repetitive inner speech trials, or both types together did not produce better results.

Nevertheless, we trained a single LDA model across the 3 best sessions, achieving 33% cross-validation accuracy. The same per-session folds were employed as in the previous analysis. To train a single LDA model across sessions, we made some modifications to the decoding pipeline. Rather than using the 1-second trial, we utilised the entire 4-second epoch with the four consecutive cues to compute covariance. To account for between-session differences, the mean session-level evoked response was subtracted from each trial before covariance computation, and the mean session-level covariance was also subtracted from each trial's covariance. Since we evaluated many different methods on this data, and we selected these 3 sessions based on the previous analysis, there is a risk this result is inflated. Running the same decoding approach on all 10 sessions reduced cross-validated accuracy to 23.2%.

While the EEG data provides promising results, with some sessions exhibiting above-chance decoding accuracy, we must be cautious about drawing robust con-

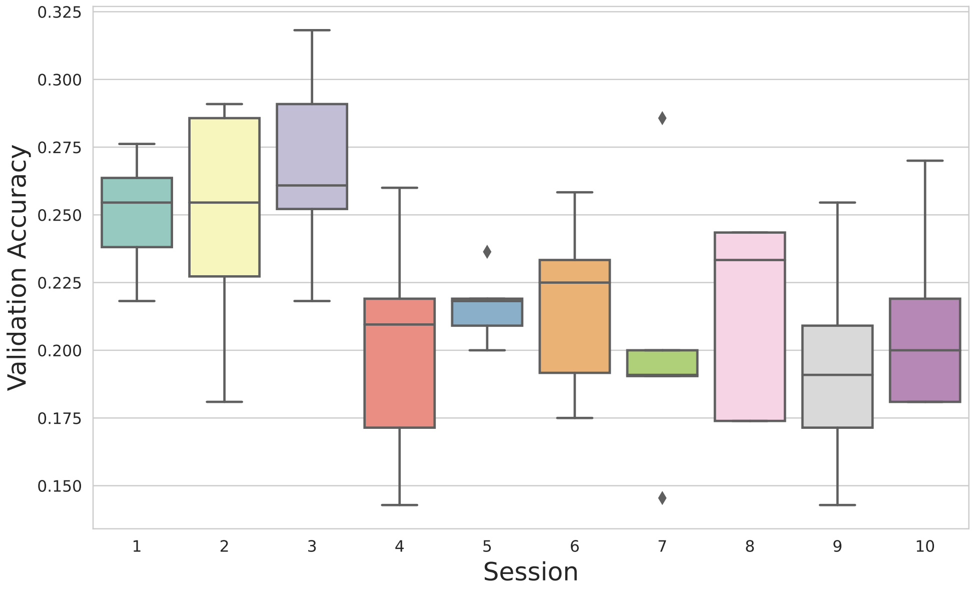


Figure 6.9: Validation accuracy distributions across the 5 folds of the 10 EEG sessions of P4 in experiment version 1. Separate LDA models are trained and evaluated on each fold and session to decode which of the 5 words is being used in the 1-second inner speech trials. Chance level is 0.2.

clusions, due to the limited performance and risk of overfitting. The limited inner speech performance precluded assessment of transfer between silent reading and inner speech tasks. Evaluating transferability between sessions was also not feasible as most sessions displayed chance-level performance.

6.3.4 Decoding silent reading in experiment version 2

In this version, we collected a substantial number of silent reading trials only from 3 participants, across 4 modalities. For each session, we implemented the LDA-NN approach from Chapter 3 with 5-fold cross-validation (Figure 6.10). We utilised the 500 ms following word onset as our examples for decoding. For CTF, Elekta, and OPM data, the dimensionality of the LDA-NN reduction was set to 50, and for EEG to 20.

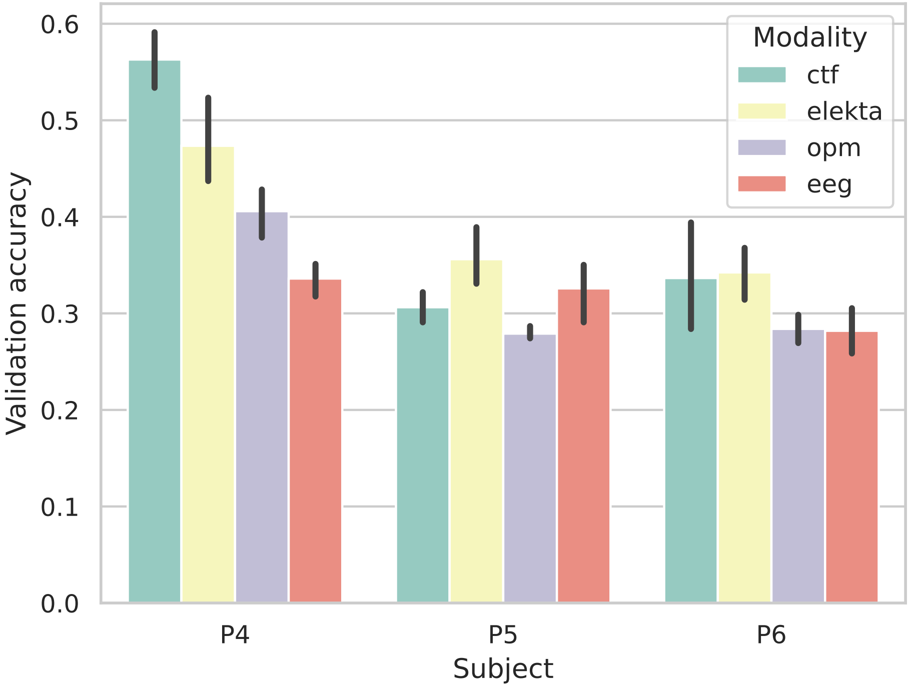


Figure 6.10: Validation accuracy (across 5 folds) for each session in experiment version 2. Separate LDA-NN (see Chapter 3) models are trained and evaluated on each fold and session to decode which word is presented during the 1-second trials. Black bars indicate 95% confidence interval. Chance level is 0.2 due to having 5 words with equal trial counts.

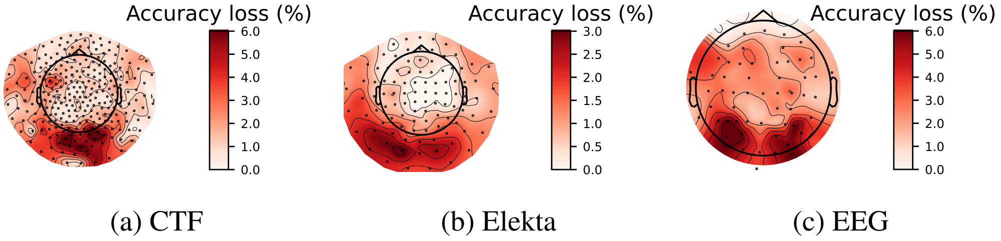
Accuracies are generally low, providing evidence that even with the visual component and numerous trials, silent reading is a challenging decoding task. P4 exhibits higher performance than the other two participants across all modalities except EEG. While CTF data achieved the best performance for P4, followed by Elekta, OPM and EEG, this is not the case for the other participants. Across modalities, P5 and P6 display more comparable performance. CTF and Elekta appear higher, while OPM and EEG are slightly lower but similar (especially for P6). The discrepancy between the CTF results for P4 and P6/P5 is particularly surprising. As depicted in Figure 6.4, unfortunately, due to experimental difficulties, the OPM sensor coverage of the visual area was inferior in these participants compared to

P4. This could potentially explain the lower OPM performance.

It is difficult to derive conclusions from these results, and the experiment should be replicated across more subjects to enable a more robust comparison between modalities. It seems that traditional MEG scanners exhibit the best performance, while EEG and OPM lag behind but are comparable. This provides evidence that challenging decoding tasks such as silent reading are feasible with OPMs. Further innovation and better spatial coverage should enhance OPM decoding performance to approach traditional MEG.

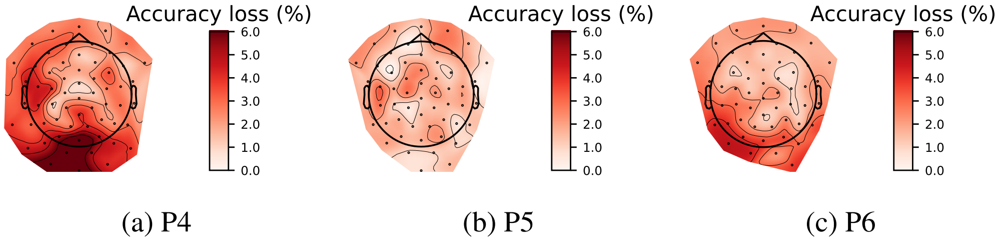
Next, we investigated the temporal and spatial PFI of the decoding models. We followed the methodology presented in Chapter 3. We expect that PFI should appear similar across modalities. When plotting the spatial PFI for each modality we average across subjects and cross-validation folds. We utilised 20 permutations and set the number of nearby sensor locations for the spatial window to 4 in all modalities. Note this equates to 12 channels for Elekta, 8 channels for CTF, and 12 channels for OPMs, since these modalities contained multiple sensors at the same site. We depict the spatial PFI of Elekta, CTF, and EEG in Figure 6.11. It is clear that the visual area drives decoding across all modalities. We plot the OPM accuracy loss maps separately for each participant due to variability in available sensors (Figure 6.12). This demonstrates similar visual importance in P4 and P6. PFI was ineffective in P5, possibly due to limited decoding performance.

Finally, we illustrate temporal PFI across subjects and modalities in Figure 6.13. We utilised 20 permutations and a temporal window of 100 ms. Since decoding is driven by the visual region, we would anticipate peak accuracy loss around 150ms, which is indeed evident for P4. This subject also exhibits much less noisy timecourses. Across modalities, timecourses appear similar. Notably, all modalities except OPM display a second, smaller peak around 250ms. In other



(a) CTF (b) Elekta (c) EEG

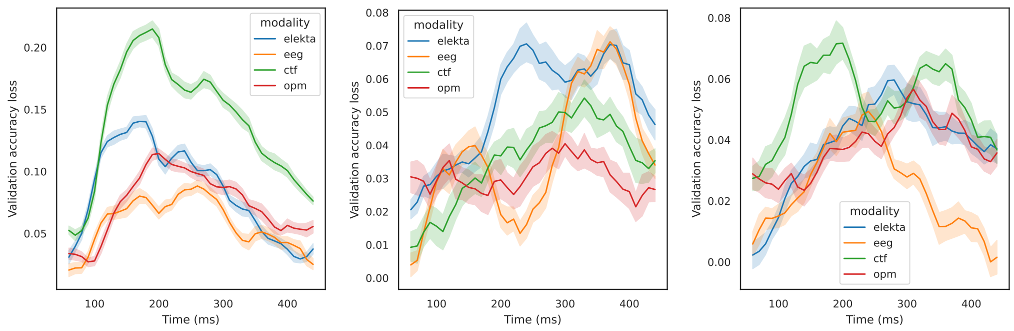
Figure 6.11: Sensor importance maps averaged across subjects for 3 modalities in experiment version 2. The importance maps are obtained by running spatial PFI on the trained LDA-NN decoding models (see Chapter 3 for methods). Darker red shading indicates higher accuracy loss and thus higher stimulus-related information content.



(a) P4 (b) P5 (c) P6

Figure 6.12: Sensor importance maps across subjects on the OPM recordings of experiment version 2. The importance maps are obtained by running spatial PFI on the trained LDA-NN decoding models (see Chapter 3 for methods). Darker red shading indicates higher accuracy loss and thus higher stimulus-related information content. Note that P5 and P6 had less channels available, hence the smaller topographic map.

subjects, accuracy loss peaks later in the trial. This could reflect slower reaction times when silently reading. Interestingly, the CTF data for P6 and the EEG data for P5 exhibit two distinct peaks. This could indicate decoding is driven by both visual word processing and language processing due to silent reading. These plots highlight substantial between-subject variability in task-related brain-activity.



(a) P4 (b) P5 (c) P6

Figure 6.13: Temporal PFI across the 3 subjects (P4, P5, P6) and 4 modalities (lines with different colours) in experiment version 2. The timecourses are obtained by running temporal PFI on the trained LDA-NN decoding models (see Chapter 3 for methods). Shading indicates 95% confidence interval across PFI permutations. The horizontal axis indicates time since stimulus onset.

6.4 Discussion

In this discussion we will first present and review related works for decoding inner speech from invasive and non-invasive modalities. Then in Section 6.4.5 we summarise our findings and conclude this chapter.

Typical experimental paradigms used in neuroimaging studies of inner speech include silent word repetition and silent reading, similar to ours. As reviewed by Geva (2018), these studies consistently demonstrate that compared to overt speech conditions, inner speech leads to reduced activation in motor regions like primary motor cortex and auditory sensory areas like primary auditory cortex. Inner speech also shows less engagement of sensory feedback regions like the superior temporal sulcus. However, inner speech robustly activates left hemisphere language regions. The authors suggest this enhanced activation in phonological and semantic regions may serve as an automatic compensatory mechanism that augments inner speech performance given the reduced sensory and motor feedback.

However, accurately interpreting the results of neuroimaging studies on inner speech remains challenging. Methodological limitations include controlling for overt speech production during inner speech tasks and ensuring participants are actually using inner speech rather than alternative cognitive strategies. Research on visual and motor imagery highlights that imagery across modalities relies on networks similar to actual perception and action (Kosslyn and Pylyshyn, 1994), reflecting shared neural processing. However, identifying a consistent substrate for imagery across all sensory modalities has proven more difficult.

When comparing invasive and non-invasive methods, the only major differences are in the brain signals and noise characteristics of each recording modality. The cognitive process of interest - inner speech itself - remains constant across invasive and non-invasive recordings. Invasive methods can record neural activity with much higher spatial resolution and signal-to-noise ratio (SNR). Thus, the key question becomes determining how reduced resolution and SNR in non-invasive EEG or MEG impacts detectability of inner speech processes. We discuss relevant invasive research of inner speech next.

6.4.1 Invasive methods

Inner speech has been most successfully studied through invasive methods. Electrocorticography (ECoG), where electrode arrays are placed below the skull directly on the brain surface, is one such technique. More invasive methods involve implanting electrodes within cortical tissue to record single neuron activity.

Martin et al. (2016) demonstrate one of the first successful decodings of individual words during imagined speech from direct cortical recordings in humans. Their study design involved recording high gamma activity using ECoG during listening, overt speech, and imagined speech conditions for 6 different words. They developed a binary classification approach using support vector machines that incorporated dynamic time warping (DTW) to account for temporal variability in speech production. At the group level, classification accuracy was significantly above chance for imagined speech, with the best word pair reaching 88% accuracy. However, across all word pairs accuracy was much lower, and performance was variable between subjects. Discriminative information was located primarily in the superior temporal gyrus, inferior frontal gyrus, and sensorimotor cortex, consistent with their role in speech processing (Hickok and Poeppel, 2007). This work can inform future noninvasive research on which brain areas to focus on and employ techniques like DTW to handle temporal variability.

Recent work by Wandelt et al. (2022) demonstrates the feasibility of decoding internal speech from single neuron activity in the supramarginal gyrus (SMG) and somatosensory cortex (S1) of a tetraplegic human participant. Their study design allowed comparison of neural activity between visual word reading, listening, vocalised, and inner speech across 8 words. The authors found individual SMG neurons showed selective tuning to specific words during the internal speech condition. Using these neural signals, they achieved up to 91% decoding accuracy for internal speech words using a real-time setting.

Importantly, the authors found strong shared neural representations in SMG between internal speech, reading visually presented words, and vocalised speech production. This points to the involvement of common underlying cognitive processes between these tasks. Their decoder was also robust to different internal speech strategies, such as auditory vs. visual imagery, suggesting flexibility for individual mental strategies in future applications. Specifically, they tested the generalisation performance of the decoder between all tasks. Decoders trained on auditory cue trials were less generalisable to inner and vocalised speech than those trained on written cue trials. This shows silent reading brain activity may be closer to pure inner speech. While the cue modalities were separable during the cue-phase brain activity, they overlapped during subsequent phases. Thus, internal and vocalised speech representations may not be influenced by the cue modality, a promising result for repetitive inner speech paradigms.

In a different line of work, Willett et al. (2021b) demonstrate the potential for real-time decoding of attempted handwriting movements from neural activity as a high-speed BCI. In this study, a participant with tetraplegia from spinal cord injury attempted to handwrite letters and words by imagining holding a pen and writing. Neural activity was recorded from intracortical electrode arrays implanted in the hand area of motor cortex. They found individual neurons showed selective patterns of activation for different handwritten letters, enabling reconstruction of pen trajectories. While not direct inner speech, it remains a purely imagined task with no external stimuli or produced behaviour. In online experiments, the participant achieved remarkable typing speeds of over 90 characters per minute, with over 94% raw accuracy. While an invasive approach was used, the neural dynamics revealed in motor cortex potentially inform non-invasive BCI design. Attempted handwriting may be a promising paradigm for EEG and imaging BCIs if cortical patterns can be sufficiently resolved.

While limited in humans for ethical reasons, invasive recordings in neurological patients provide unparalleled detailed characterisation of the neurophysiology underlying inner speech phenomena. These studies unequivocally show that inner speech and movement imagery decoding are possible invasively, and demonstrate their potential in BCI applications. Next, we turn to non-invasive studies of inner speech.

6.4.2 EEG

While invasive studies are rare due to the nature of intracranial recordings, EEG studies of inner speech are also uncommon because of the difficulty in overcoming the low signal-to-noise ratio and spatial resolution inherent in scalp EEG recordings.

Cooney et al. (2019b) investigate using CNNs to classify imagined spoken word-pairs from EEG signals. Their dataset contained 6 Spanish words imagined by 15 subjects. All 15 possible word-pairs were extracted and EEG signals corresponding to an early imagined speech time-window were used. Results showed a deep CNN achieved the best average accuracy of 62.37% across subjects and word-pairs. This performance however is still barely above chance level, indicating the ongoing difficulty of decoding imagined speech from noisy EEG recordings.

Ling et al. (2019) investigate how visual words are represented in the brain using EEG-based decoding and image reconstruction techniques. Their study had 14 participants view 80 high-frequency nouns while recording EEG data. They then used multivariate pattern analysis on the EEG data to decode visual and orthographic properties of the words. Specifically, they were able to decode pairwise word discriminability well above chance across participants, with peak performance around 170ms after stimulus onset. They also applied representational similarity analysis to show the word decoding results correlated with visual and orthographic similarity but not semantic similarity. This is perhaps unsurprising as decoding visual activity is well-studied with EEG.

6.4.3 MEG

Défossez et al. (2022) present a novel method for decoding natural continuous speech from non-invasive MEG and EEG recordings. The authors leverage recent advances in self-supervised speech representation learning, specifically wav2vec 2.0 (Baevski et al., 2020), to obtain semantically meaningful speech embeddings from raw audio. These speech embeddings are aligned with M/EEG signals recorded while participants passively listened to audio samples. A joint CNN architecture with a contrastive loss is used to predict the speech embeddings from the neural signals. Without any individual calibration, their model can identify 3-second speech segments with up to 72.5% top-10 accuracy across nearly 1,600 samples for MEG and 19.1% across 2,600 samples for EEG.

For decoding inner speech, this study provides a promising framework to handle individual variability and extract meaningful speech features from limited data. The zero-shot decoding is particularly impressive, as it avoids constraints of classifiers trained on small stimulus sets. However, additional work is needed to apply this to inner speech due to the lack of audible signals for alignment. Methodologically, this work sits at the interface of our efforts in Chapters 4 and 5. They leverage both group modelling and training large models across multiple datasets. Their approach is well suited to incorporating forecasting or other self-supervised objectives. The contrastive loss allows for out-of-distribution decoding, as it is not limited by the categorical nature of standard classifiers. Furthermore by incorporating multiple feature extractors (e.g. CNNs trained on images), the same contrastive approach and brain model can be applied to various decoding tasks.

Direct MEG investigations of inner speech are limited. Dash et al. (2020a) decode 5 imagined and overtly spoken phrases from MEG. Three decoding methods were tested: an artificial neural network (ANN) using statistical features, a CNN on time-frequency images, and a CNN with combined spatial, spectral and temporal features. The CNN approaches significantly outperformed the ANN, achieving up to 96% accuracy for spoken phrases and 93% for imagined phrases with the combined features. A key limitation is using phrases which likely contain more decodable information but are harder to scale up. A contrastive approach, or decoding at the phoneme/word level is more desirable.

6.4.4 OPM-MEG

While superconducting quantum interference devices (SQUIDs) are traditionally used for MEG, optically pumped magnetometers (OPMs) have recently emerged as a promising alternative for MEG measurements (Boto et al., 2018). OPMs offer several advantages over SQUIDs including room-temperature operation, lower cost, higher sensitivity, and allow head motion (Boto et al., 2017). Their compact size also enables flexible sensor arrays that can be customized to target specific brain regions or conform to individual head shapes (Boto et al., 2018).

A key application of MEG is non-invasive decoding of mental states and cognitive processes from neural activity patterns. Some initial studies have now explored using OPM-MEG for neural decoding. Wittevrongel et al. (2021) demonstrate OPM-MEG enables robust single-trial analysis and real-time decoding for BCI applications. They compared OPM-MEG and EEG for decoding visual evoked responses, including event-related potentials/fields (ERPs/ERFs) to motion-onset and steady-state visual evoked potentials (SSVEPs) to flickering stimuli. For motion-onset, OPM-MEG and EEG showed similar ERP/ERF components (N/M200, P/M300) with comparable signal-to-noise ratios. For SSVEPs, OPMs had higher SNR in the high frequency range (25-29 Hz) while EEG was better at low frequencies (8-12 Hz). In a real-time SSVEP spelling task, OPM-MEG achieved 97.7% average accuracy comparable to state-of-the-art EEG systems. These results validate OPM-MEG for robust single-trial decoding in BCI applications. The improved SNR and spatial resolution suggest OPMs could enable more advanced decoding capabilities. Their wearable and flexible nature makes them well-suited for practical BCI applications.

6.4.5 Conclusion

This chapter presented an in-depth investigation into decoding inner speech and silent reading from non-invasive electrophysiological recordings. The key findings were the following. Silent reading of words could be decoded from EEG, MEG, and OPMs with 30-40% accuracy across 5 words, driven by early visual processing. Comparing modalities showed traditional MEG had the best performance for decoding silent reading, while OPMs and EEG had lower but comparable accuracies. Inner speech decoding was mostly at chance levels in EEG and MEG across multiple decoding approaches. The highest accuracy reached for inner speech was 33% across 3 EEG sessions using covariance features, but the validity of this result is debatable.

Our silent reading results demonstrate the feasibility of decoding visual representations of words from non-invasive recordings, consistent with prior EEG and MEG decoding studies (Chan et al., 2011; Ling et al., 2019). The decoding appeared to be driven by early visual responses, with a later peak potentially reflecting higher-level language processing (Kutas and Van Petten, 1988). This late component merits further investigation as a marker of semantic processing. While more subjects are needed for a robust comparison, OPMs achieved decoding accuracy comparable to that of EEG. In one participant with good spatial coverage, OPM decoding performance was even better than EEG, highlighting their promise given advantages like wearability. Dense coverage of visual regions may be critical for the investigated decoding task. Our results also underscored the high between-subject variability, both in overall performance and in the timing of informative decoding features.

In contrast to silent reading, our extensive efforts to decode two types of inner speech were largely unsuccessful across EEG and MEG. While we explored various decoding algorithms and experimental designs, accuracy never substantially exceeded chance levels. This contrasts with more promising results from intracranial recordings in humans (Martin et al., 2016; Wandelt et al., 2022), and suggests non-invasive signals may not adequately capture the subtle dynamics of inner speech. There was also substantial between-session variability.

In addition to the analyses presented, numerous unsuccessful decoding approaches were pursued on the inner speech data. On the MEG recordings, these included logistic regression, CNNs, SVMs, concatenating or averaging consecutive trials, and per-session versus aggregated-session decoding. For EEG, besides the MEG-based methods, other unsuccessful attempts involved temporal alignment of trials, PCA denoising, Riemannian classifiers, baseline correction, and wavelet features. We tried several referencing approaches such as common average reference, mastoid references, and current source density estimation through the Laplacian method. We also tried wider bandwidth filters (up to 100Hz), however we noticed no improvement with the addition of gamma activity, even though it is reported in invasive inner speech studies. Noise and superposition in scalp measurements may overshadow weak gamma oscillations.

Several factors could underlie the difficulty of decoding inner speech non-invasively. Inner speech lacks the external stimuli and muscle activations present during overt tasks, reducing the signal-to-noise ratio. There is also high inter-individual variability in inner speech strategies. Here we focused on collecting large trial counts from a few participants rather than a small sample across many subjects. Our cross-cue paradigm may also induce visual confounds that overshadow inner speech signals. Having participants repeatedly imagine brief, single words likely differs from natural inner speech involving longer phrases. Further limitations of our work include the small number of participants and the small set of words.

Future investigations could explore alternative paradigms more representative of natural speech, such as imagining longer phrases or reading whole sentences silently. Transfer learning and self-supervision may help extract robust inner speech representations amidst noise (Défossez et al., 2022). Intracranial findings point to superior temporal, inferior frontal, and motor areas as promising decoding targets. For non-invasive BCIs, approaches beyond word-level decoding may be needed for inner speech-based communication, such as decoding phonemes, or imagined handwriting.

One downside of our task design is that the low-level characteristics of the visual appearance of the words might be a confounding factor in our silent reading results. Future work should consider using varied graphical representations of the same word across trials.

In summary, our results highlight the significant challenges in decoding inner speech correlates non-invasively compared to overt tasks. Substantial innovation in experiments and analyses will likely be essential to enhance the fidelity of decoded inner speech for BCIs. While current decoding performance was limited, our proof of concept work provides a useful platform with extensive trial counts for future efforts at modelling inner speech. Having multiple sessions allows for testing across-session generalisability. Neuroscientific understanding of inner speech may be deepened through comparing the different experimental paradigms.

7 | Discussion

This thesis delves into the realm of brain modelling, targeting the enhancement of decoding performance and BCI communication speeds. The significance of this research becomes apparent when considering clinical populations, particularly individuals with locked-in syndrome, who heavily rely on such technology for communication. In the broader context, BCIs symbolise the culmination of the seamless integration of advanced technological tools into human lives.

Historically, human civilisation has witnessed an ongoing assimilation of tools to augment our capabilities. Although this has spanned millennia, the advent of computers and subsequently smartphones represented important leaps. These devices, coupled with the proliferation of wearables like smartwatches, underscore an evolving paradigm of human-machine symbiosis. Pre-computer age tools predominantly showcased mechanical prowess; however, with the dawn of the digital age, this shifted towards enhancing human cognitive capabilities. The field of artificial intelligence (AI) stands testament to this shift, wherein computers, with their unparalleled cognitive processing capabilities, are reshaping our understanding of intelligence (Letheren et al., 2020; Chollet, 2019).

An evident lack remains in the realm of interfaces bridging human cognition with these technological advancements. Predominant tools still rely heavily on manual interaction, be it through keyboards or touch displays. Thus, there is a need for a more direct, and arguably more intuitive interface, directly with the human brain. The path to BCI improvement can be split into two primary avenues. The first involves the development of sophisticated hardware that facilitates more detailed brain recordings. The second encompasses the design of innovative methods capable of circumventing the constraints of current hardware. Our research aligns with the latter approach. Aiming for maximal societal impact, our focus was on leveraging non-invasive technologies, with a particular emphasis on electrophysiology, given its fast temporal dynamics.

Delving deeper into current non-invasive hardware, two frontiers emerge in the pursuit of BCI enhancement. The first encompasses purely software-centric solutions, extensively explored in Chapters 3, 4, and 5. The second frontier pertains to experimental methodologies, explored in Chapter 6. Our work addressed critical challenges in both domains, though the actual application to end-users is left for future exploration.

BCI technology hinges on machine learning methods. To harness the full potential of such methodologies, complex, nonlinear models are indispensable, as are large datasets. A notable challenge with non-invasive electrophysiology is the pronounced variability across time, participants, and tasks. Most task datasets possess limited data from individual participants, and often lack a diverse participant pool to encapsulate the full spectrum of brain variability. Thus, a pragmatic approach to BCI improvement involves crafting methods adept at navigating variability within constrained datasets.

7.1 Variability within individuals

Chapter 3 embarked on addressing variability intrinsic to individual brains. Our findings show the efficacy of linear decoding on full epochs of stimulus presentation. A crucial revelation was the performance improvement gained from the integration of a dimensionality reduction technique targeting the channel dimension during supervised training of the decoder. To understand these improvements better we should analyse the challenges in the application of machine learning to M/EEG data. At the outset, the data size is modest, encompassing 30 examples for each condition across a total of 118 conditions. The 306 MEG channels exhibit substantial covariance, thereby presenting a high-dimensional input with correlated features. It is for this reason that PCA frequently emerges as a method to reduce the dimensionality of the channel space.

While PCA effectively decouples the channels, it might inadvertently eliminate task-specific information due to its unsupervised nature. Consequently, it stands to reason that executing a similar dimensionality reduction but within the decoding objective yields superior results. Once such a projection is learned, it mirrors the utility of PCA in feature extraction. These supervised features are then amenable to integration with any conventional model, such as Linear Discriminant Analysis (LDA). Alternative methodologies for supervised dimensionality reduction, such as the Riemannian classification method, do exist (Barachant, 2014), however, they are particularly effective when the number of classes is minimal, a scenario that diverged from the datasets explored in this thesis.

Our approach is contingent on the availability of ample data, as corroborated by the limited performance observed in the small replay dataset. When operating in data-scarce environments, the *curse of dimensionality* becomes a considerable challenge, due to using features extracted from the entire epoch. In such cases methods for extracting higher-level features, such as power in different frequency bands may prove better (Higgins et al., 2022b; Hu et al., 2011). This is evidenced by the long list of various decoding features proposed in the BCI literature (Panachakel and Ramakrishnan, 2021). A further significant limitation of our work is the lack of application across diverse tasks and modalities, such as EEG.

A parallel challenge with deploying machine learning models on high-dimensional inputs is the loss of neuroscientific interpretability. We demonstrated the versatility of Permutation Feature Importance (PFI) as a tool that can be employed across various input dimensions—temporal, spatial, or spectral—to glean insights into task-associated brain activity patterns. The adaptability of PFI is commendable, allowing for analyses on a per-participant or per-condition basis. Through the window size parameter, it offers the flexibility to strike a balance between noise mitigation and pattern resolution. However, PFI does not ensure that identified patterns truly encompass task-related information, due to how model optimisation works. Nonetheless, empirical analyses have affirmed its congruence with direct methods, such as sliding window analysis.

7.2 Modelling variability between individuals

Chapter 4 shifted to another form of variability. Beyond the confines of intra-participant variability, inter-participant variability manifests itself in terms of anatomical differences, functional localisation variations, and divergent neural dynamics (Saha and Baumert, 2020). Such heterogeneities often prohibit the creation of universally applicable decoding models that exhibit consistency across individuals. From a BCI perspective, the ability to amalgamate data from a multitude of participants would be an invaluable asset. It paves the way for deploying a pre-trained model on a new individual's data without any finetuning.

The subject embedding technique investigated in Chapter 4 emerged as a salient solution. It circumvents individual variability by learning a low-dimensional representation for each participant (Chehab et al., 2022). By integrating this embedding as an input to a shared decoder across participants, we were able to approach the performance levels of subject-specific models. This showcases the potential for more potent applications of deep learning by capitalising on multi-subject datasets. Yet, this methodology warrants more exhaustive assessments, especially on larger datasets that contain a broader set of participants.

Our findings also underscored a crucial observation: while nonlinearity might not improve performance in single-subject scenarios, it becomes indispensable for incorporating information from subject embeddings into a group model. We believe that in single-subject scenarios, the limited dataset size might constrain the applicability of nonlinear models. Furthermore, our research illuminated the applicability of PFI to kernels within a convolutional network, thereby facilitating the extraction of interpretable importance maps, particularly in the spectral domain. This is in line with the efforts of the field of interpretable artificial intelligence (Linardatos et al., 2020).

Our approach has significant limitations. We did not venture into assessing the method across different task types. Nonetheless, the efficacy of subject embedding has received validation in recent works across different datasets (Défossez et al., 2022; Chehab et al., 2022). An intriguing question that emerges is whether, beyond a critical threshold of participants, the inherent variability can be implicitly modelled by a sufficiently complex model.

Another point of contention is the observed decrease in performance when training models on data from one participant and subsequently deploying it on another. We did not observe any marked improvement in such scenarios. Although the subject embedding technique was not conceptualised for these specific applications, their utility in BCI contexts is undeniable. Recent advancements in the field have witnessed the exploration of alternative methodologies, such as the subject-specific layer, which projects input data into a standardised space (Défossez et al., 2022). However, this approach also requires training on each new participant. For achieving genuine zero-shot performance, the development of models that innately learn inter-subject variability may be needed.

7.3 Towards foundational electrophysiology models

Chapter 5 embarks on addressing the inherent challenges of integrating deep learning methodologies with electrophysiological data. Several difficulties emerge, stemming from the use of different scanners, varied tasks, and diverse experimental setups. However, a common denominator across these scenarios is the presence of multichannel time series data with a high sampling rate. Drawing inspiration from the recent strides in large language models, where a single sequence model can handle a plethora of language-related tasks, one might envisage creating an analogous model for electrophysiological data. Pursuing this line of thought offers two distinct advantages. Firstly, such a model could harness data spanning multiple datasets, providing a robust framework to model variability. Secondly, drawing a parallel to language models, if we can build a deep learning model that can 'understand' brain data, it should inherently possess the ability to execute auxiliary tasks, be it encoding or decoding brain signals.

The versatility of such models could manifest in their ability to perform tasks either with or without dedicated fine-tuning for specific downstream applications. For example, our proposed Transformer-based model includes forecasting, encoding associated with task stimuli, and decoding based on Bayes' theorem. However, the decoding performance was somewhat limited, suggesting that transfer learning, facilitated through fine-tuning, might emerge as a better avenue. We did not experiment with this approach in our research, but our results showed that the performance of downstream decoding can be boosted by simulating training data. This is an alternative application of foundation models to downstream tasks.

A significant portion of our research efforts was channelled into analysing the input structure of M/EEG data. Our goal was to discern the types of architectures or inductive biases that might be optimally suited to model such data. Given the inherently sequential nature, coupled with the unparalleled performance of Transformer-based models across diverse sequential data modalities—ranging from language to audio and video—we gravitated towards the GPT2 model, the autoregressive forecasting variant of the Transformer. When compared with conventional CNN-based architectures, such as Wavenet, our findings revealed that the Transformer has the ability to generate data that exhibited a higher congruence with real data. This suggests a superior capability of the Transformer in mirroring the dynamics inherent to real-world data.

A crucial limitation of our approach is the inability to inherently accommodate information from different channels. In essence, our method could be characterised as univariate, although channel embeddings played a pivotal role in tailoring the model to individual channels. Our endeavours to include multiple channels into the input were met with limited success. We think that maintaining the innate inductive biases of Transformers, which emphasise 1D sequence modelling on embeddings of discrete tokens, is paramount. While our FlatGPT2 model did not achieve good performance, alternative strategies might hold promise. For instance, one might consider adapting the neural network-driven vector quantisation techniques exemplified by models like Jukebox (Dhariwal et al., 2020). While this model used a vector-quantised variational autoencoder (VQ-VAE, Van Den Oord et al. (2017)) in the time domain, adaptation to the channel dimension should be possible.

Some of our findings substantiated that predicting the next timestep may not serve as a robust measure of model performance. Future research should contemplate adopting multi-timestep or contrastive loss frameworks. A plausible strategy could involve deploying the VQ-VAE model across both channel and temporal dimensions, aiming to distill a more coarse sequence of discrete tokens. Nevertheless, any quantisation-centric approach must carefully consider reconstruction error. We posit that a significant portion of the signal dynamics should be entrusted to the Transformer, given its adeptness in capturing complex dynamics.

A notable challenge with forecasting models tailored for electrophysiology is the absence of external data. Intrinsically, brain activity is influenced by a plethora of external stimuli and physiological processes, many of which elude the experimenter. Consequently, the task of forecasting suffers from uncertainties. However, large language models have demonstrated remarkable efficacy, despite the fact that the vast expanse of text on the internet is not typically conditioned on the underlying motivations or contexts of its human authors. This again motivates the need to scale large electrophysiology models.

A constraint in our modelling approach is its reliance on categorical task stimuli labels. Such an approach, while effective in our context, does not readily lend itself to scalability across diverse tasks and datasets. However, it is conceivable to construct robust representations tailored for various stimulus modalities—ranging from images to audio. These representations can then serve as conditioning embeddings. As shown by Défossez et al. (2022), tools such as wav2vec (Baevski et al., 2020) can be leveraged to derive informative representations of auditory stimuli.

Our results in Chapter 5 were confined to a single dataset. A pivotal avenue for future research would be the rigorous evaluation of these models across an array of datasets and tasks. Scaling, spanning datasets, tasks, modalities, and broader research domains, via transfer learning, emerges as a promising strategy. This approach holds the potential to navigate inter-dataset variability, in pursuit of versatile, generalisable foundational and decoding models.

7.4 Probing the limits of non-invasive BCIs

Transitioning to the experimental frontier, we recognise that the efficacy of a model, regardless of its sophistication, can be substantially undermined by limitations in data volume and quality. This understanding motivated our experimental explorations in the concluding Chapter 6. While recent invasive studies have shown impressive communication rates, especially with naturalistic paradigms like imagined speech and handwriting, their non-invasive BCI counterparts appear to lag behind.

Prevailing EEG-based BCI methodologies predominantly revolve around the relatively slower P300 and SSVEP paradigms (Guan et al., 2004; İşcan and Nikulin, 2018). These artificial paradigms emerged due to the noisiness of the EEG signal. To build robust BCIs researchers needed to resort to the strong signals of the visual cortex. Motor imagery has also shown promise but is usually limited to a handful of decodable classes (Saha and Baumert, 2020; Halme and Parkkonen, 2018). A more positive outlook on these BCI methods is that by being smart about experimental paradigms EEG-based BCIs can achieve previously unimaginable performance.

Our research endeavoured to analyse the feasibility of inner speech in non-invasive modalities and to discern the potential enhancement in performance with increasing data volume. Regrettably, even with hundreds of trials from a single word, the decoding performance for inner speech hovered around chance levels. It is also worth noting that these analyses were conducted at the session level. We did not venture into cross-session or cross-participant decoding. While the decoding of silent reading showed potential, our investigations revealed that this was driven predominantly by the visual processing associated with word presentation. Therefore it is crucial for future research to investigate the shared representations of silent reading, listening, vocalised and inner speech in non-invasive modalities.

The limited success in inner speech decoding experiments could be attributed to the fact that the experimental paradigm did not emulate naturalistic phenomena. This was primarily due to the repetitiveness of tasks and the focus on isolated words. A better approach would involve evaluating decoding efficacy during free dialogue or imagination. It may well be that by using a different set of words which are phonetically or semantically more dissimilar, decoding performance would increase. While counter-intuitive, collecting inner speech data across a much larger set of words in naturalistic settings, akin to the listening experiments in Défossez et al. (2022), could also enable better performance. The categorical classification approach could be replaced by contrastive objectives with word embeddings, leveraging the representational space of language models.

An overarching limitation that permeated all our research was the confinement to small datasets with homogeneous tasks. Several other limitations warrant mention. Across all chapters, the cohort of human participants was relatively modest in size. This was particularly pronounced in the multimodal decoding experiments, which compared EEG, MEG, and OPM recordings. This requires expanded replication to enable robust conclusions. The restricted sensor coverage in our experiments limited information content in OPMs. In future OPM studies dense coverage of visual and language areas should be ensured. With advancements leading to more comprehensive sensor coverage, OPMs could potentially eclipse traditional MEG systems, especially if they facilitate the deployment of custom sensor arrays (Boto et al., 2018).

7.5 The future of brain modelling for BCIs

Future research should focus on scaling across multiple facets: datasets, participant cohorts, and the transferable knowledge across studies. This will pave the way to more effectively capture and model the intrinsic variability embedded within cognitive neural dynamics. The progression of self-supervised and few-shot learning paradigms, particularly tailored for electrophysiology data, holds the promise of revolutionising the domain. Such advancements will be crucial in harnessing the full potential of limited labelled datasets (Banville et al., 2021).

The establishment of expansive, open-access corpora comprising raw neural recordings is pivotal. These repositories, encompassing data from a multitude of diverse studies, could serve as a bedrock for training robust, generalisable models. Realising this vision necessitates fostering collaborative data-sharing initiatives that operate within the confines of privacy regulations (Poldrack and Gorgolewski, 2014). Additionally, the development and deployment of automated annotation and processing tools could further streamline this process. A salient research trajectory involves testing whether foundational models can enhance the sample efficiency and adaptability of BCIs, spanning tasks, sessions, and participants. Such advancements could have profound implications, significantly elevating the quality of life for clinical populations.

Another integral component revolves around rapid modalities of imagined communication, akin to those explored in invasive data studies. Imagined paradigms have received limited attention in non-invasive research, primarily due to the challenges posed by faint signals relative to ongoing brain activity and noise. Several prospective directions have been identified, such as delving into various facets of inner speech, e.g., imagining mouth movements or auditory imagination of speech, and even delving into different linguistic units, e.g., phonemes. Different modalities, like imagined handwriting, too, need deeper exploration through non-invasive methods. While in silent speech paradigms EMG is more practical for BCI applications, inner speech lacks any motor output and thus can only be detected in the brain. However, augmenting inner speech with motor imagery (e.g. imagining mouth movements), could provide stronger signals in the motor cortex compared to pure inner speech.

We are particularly inspired by the contrastive methodology delineated by Défossez et al. (2022). While adapting to inner speech presents challenges, we posit that a closed-loop setup augmented with real-time feedback could show promise. In such a paradigm, participants would commence by reading a word aloud, followed by auditory exposure to the word's pronunciation at a designated cue. As the experiment progresses, the audio rendition could be mixed with noise or its volume incrementally reduced. The objective is to seamlessly transition the participant to rely predominantly on inner speech, especially as the audio cue becomes increasingly unintelligible.

Leveraging the CLIP approach (Radford et al., 2021) on this data could be instrumental. This involves distilling positive samples by identifying audio snippets that exhibit maximal similarity with the inner speech representations. While in the beginning of the experiment the methodology would be similar to the one used in Défossez et al. (2022), as the participant switches to inner speech the contrastive learning through paired audio segments becomes challenging.

Our experimental endeavours highlight that the most significant constraints might be deeply rooted in the hardware. Non-invasive electrophysiological recordings may be too noisy, thereby challenging the extraction of salient signals associated with inner speech. We believe that significant strides in hardware innovation and recording modalities are imperative. Given the empirical evidence supporting the efficacy of inner speech decoding in invasive modalities, the key is overcoming challenges posed by factors like signal-to-noise ratio and spatial resolution.

More theoretical and experimental work should be conducted to understand the channel information capacity of non-invasive modalities. While here we think that noise or superposition of signals drowns out any inner speech-related activity, it would be important to be able to show this. A possible experiment could involve simultaneous ECoG and EEG recordings during an inner speech task to directly compare the synchronised signals from the two modalities. In addition, through source modelling, we could pin down inner speech activity from ECoG and subject it through a forward head model to see how signals manifest on the scalp. This would provide more concrete evidence whether the channel capacity of EEG is the limiting factor in detecting inner speech.

One promising innovation in recent times has been the advent of OPMs. These devices, in theory, possess a superior signal-to-noise ratio, and thanks to engineering advancements, a larger number of sensors can be densely packed on the human scalp. Yet, there remains a divide between theoretical potential and actual performance, necessitating further engineering innovations. As our silent reading experiments have shown, even the most advanced OPM systems, as of now, do not quite match traditional MEG scanners. It is conceivable that even future OPMs might fall short in discerning the subtle signals of inner speech. Hence, pushing the envelope in non-invasive techniques will invariably hinge on pioneering research spanning the complex physics and biology of the human brain.

7.6 Conclusion

In summary, this thesis presents a range of methodological innovations and empirical insights, each crafted to enable subsequent research in non-invasive brain decoding. At the crux of our research lies an integrated modelling paradigm, encapsulating the multifaceted variability inherent to neural dynamics. Our findings underscore the significance of both advanced hardware capabilities and powerful data-driven models in the pursuit of enhanced BCI communication speeds.

As the boundaries between technology and human cognition continue to blur, the onus is on the scientific community to harness these advancements, ensuring they are both accessible and adaptable. By carefully addressing the challenges and harnessing the opportunities that lie ahead, we believe that the horizon holds the promise of a world where BCIs are not just niche assistive tools but an integral extension of human expression and capability.

Bibliography

Akbari, H., Khalighinejad, B., Herrero, J. L., Mehta, A. D., and Mesgarani, N. (2019). Towards reconstructing intelligible speech from the human auditory cortex. Scientific reports, 9(1):1–12.

Alammar, J. (2019). The illustrated gpt-2 (visualizing transformer language models). Jalammar. github. io. https://jalammar. github. io/illustrated-gpt2.

Albright, T. D. (1984). Direction and orientation selectivity of neurons in visual area mt of the macaque. Journal of neurophysiology, 52(6):1106–1130.

Alderson-Day, B. and Fernyhough, C. (2015). Inner speech: development, cognitive functions, phenomenology, and neurobiology. Psychological bulletin, 141(5):931.

Altmann, A., Toloşi, L., Sander, O., and Lengauer, T. (2010). Permutation importance: a corrected feature importance measure. Bioinformatics, 26(10):1340–1347.

Anumanchipalli, G. K., Chartier, J., and Chang, E. F. (2019). Speech synthesis from neural decoding of spoken sentences. Nature, 568(7753):493–498.

Arlot, S. and Celisse, A. (2010). A survey of cross-validation procedures for model selection.

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Information fusion, 58:82–115.

Azevedo, F. A., Carvalho, L. R., Grinberg, L. T., Farfel, J. M., Ferretti, R. E., Leite, R. E., Filho, W. J., Lent, R., and Herculano-Houzel, S. (2009). Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. Journal of Comparative Neurology, 513(5):532–541.

Babenko, A. and Lempitsky, V. (2014). Additive quantization for extreme vector compression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 931–938.

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. Advances in neural information processing systems, 33:12449–12460.

Baillet, S. (2017). Magnetoencephalography for brain electrophysiology and imaging. Nature neuroscience, 20(3):327–339.

Baillet, S., Mosher, J. C., and Leahy, R. M. (2001). Electromagnetic brain mapping. IEEE Signal processing magazine, 18(6):14–30.

Baker, A. P., Brookes, M. J., Rezek, I. A., Smith, S. M., Behrens, T., Probert Smith, P. J., and Woolrich, M. (2014). Fast transient networks in spontaneous human brain activity. elife, 3:e01867.

Banville, H., Chehab, O., Hyvärinen, A., Engemann, D.-A., and Gramfort, A. (2021). Uncovering the structure of clinical eeg signals with self-supervised learning. Journal of Neural Engineering, 18(4):046020.

Barachant, A. (2014). Meg decoding using riemannian geometry and unsupervised classification. Grenoble University: Grenoble, France.

Barachant, A. et al. (2022). pyriemann/pyriemann: v0. 3. Zenodo, Jul.

Bârzan, H., Ichim, A.-M., Moca, V. V., and Mureşan, R. C. (2022). Time-frequency representations of brain oscillations: which one is better? Frontiers in Neuroinformatics, 16:25.

Bashivan, P., Rish, I., Yeasin, M., and Codella, N. (2015a). Learning representations from eeg with deep recurrent-convolutional neural networks. arXiv preprint arXiv:1511.06448.

Bashivan, P., Rish, I., Yeasin, M., and Codella, N. (2015b). Learning representations from EEG with deep recurrent-convolutional neural networks. arXiv preprint arXiv:1511.06448.

Bastiaansen, M. and Hagoort, P. (2006). Oscillatory neuronal dynamics during language comprehension. Progress in brain research, 159:179–196.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. The annals of mathematical statistics, 41(1):164–171.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias–variance trade-off. Proceedings of the National Academy of Sciences, 116(32):15849–15854.

Bell, A. J. and Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. Neural computation, 7(6):1129–1159.

Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150.

Benz, K. R. (2020). Hyperalignment in meg: A first implementation using auditory evoked fields.

Berger, H. (1929). Über das elektroenkephalogramm des menschen. Archiv für psychiatrie und nervenkrankheiten, 87(1):527–570.

Bigdely-Shamlo, N., Mullen, T., Kothe, C., Su, K.-M., and Robbins, K. A. (2015). The prep pipeline: standardized preprocessing for large-scale eeg analysis. Frontiers in neuroinformatics, 9:16.

Bijsterbosch, J. D., Woolrich, M. W., Glasser, M. F., Robinson, E. C., Beckmann, C. F., Van Essen, D. C., Harrison, S. J., and Smith, S. M. (2018). The relationship between spatial configuration and functional connectivity of brain regions. elife, 7:e32992.

Bilmes, J. A. et al. (1998). A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. International computer science institute, 4(510):126.

Birbaumer, N., Ghanayim, N., Hinterberger, T., Iversen, I., Kotchoubey, B., Kübler, A., Perelmouter, J., Taub, E., and Flor, H. (1999). A spelling device for the paralysed. Nature, 398(6725):297–298.

Blankertz, B., Tomioka, R., Lemm, S., Kawanabe, M., and Muller, K.-R. (2007). Optimizing spatial filters for robust eeg single-trial analysis. IEEE Signal processing magazine, 25(1):41–56.

Borovykh, A., Bohte, S., and Oosterlee, C. W. (2018). Dilated convolutional neural networks for time series forecasting. Journal of Computational Finance, Forthcoming.

Boto, E., Holmes, N., Leggett, J., Roberts, G., Shah, V., Meyer, S. S., Muñoz, L. D., Mullinger, K. J., Tierney, T. M., Bestmann, S., et al. (2018). Moving magnetoencephalography towards real-world applications with a wearable system. Nature, 555(7698):657–661.

Boto, E., Meyer, S. S., Shah, V., Alem, O., Knappe, S., Kruger, P., Fromhold, T. M., Lim, M., Glover, P. M., Morris, P. G., et al. (2017). A new generation of magnetoencephalography: Room temperature measurements using optically-pumped magnetometers. NeuroImage, 149:404–414.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers, pages 177–186. Springer.

Boudewyn, M. A., Luck, S. J., Farrens, J. L., and Kappenman, E. S. (2018). How many trials does it take to get a significant erp effect? it depends. Psychophysiology, 55(6):e13049.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020a). Language Models are Few-Shot Learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, Advances in Neural Information Processing Systems, volume 33, pages 1877–1901. Curran Associates, Inc.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020b). Language models are few-shot learners. arXiv preprint arXiv:2005.14165.

Brüers, S. and VanRullen, R. (2018). Alpha power modulates perception independently of endogenous factors. Frontiers in neuroscience, 12:279.

Brumberg, J. S. and Guenther, F. H. (2010). Development of speech prostheses: current status and recent advances. Expert review of medical devices, 7(5):667–679.

Buxton, R. B. (2013). The physics of functional magnetic resonance imaging (fmri). Reports on Progress in Physics, 76(9):096601.

Buzsaki, G. (2006). Rhythms of the Brain. Oxford university press.

Buzsáki, G., Anastassiou, C. A., and Koch, C. (2012). The origin of extracellular fields and currents—eeg, ecog, lfp and spikes. Nature reviews neuroscience, 13(6):407–420.

Buzsaki, G. and Draguhn, A. (2004). Neuronal oscillations in cortical networks. science, 304(5679):1926–1929.

Buzsáki, G., Stark, E., Berényi, A., Khodagholy, D., Kipke, D. R., Yoon, E., and Wise, K. D. (2015). Tools for probing local circuits: high-density silicon probes combined with optogenetics. Neuron, 86(1):92–105.

Carlson, T., Tovar, D. A., Alink, A., and Kriegeskorte, N. (2013). Representational dynamics of object vision: the first 1000 ms. Journal of vision, 13(10):1–1.

Carlson, T. A., Hogendoorn, H., Kanai, R., Mesik, J., and Turret, J. (2011). High temporal resolution decoding of object position and category. Journal of vision, 11(10):9–9.

Chambon, S., Galtier, M. N., Arnal, P. J., Wainrib, G., and Gramfort, A. (2018). A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 26(4):758–769.

Chan, A. M., Halgren, E., Marinkovic, K., and Cash, S. S. (2011). Decoding word and category-specific spatiotemporal representations from meg and eeg. Neuroimage, 54(4):3028–3039.

Chaudhary, U., Birbaumer, N., and Ramos-Murguialday, A. (2016). Brain–computer interfaces for communication and rehabilitation. Nature Reviews Neurology, 12(9):513–525.

Chehab, O., Defossez, A., Loiseau, J.-C., Gramfort, A., and King, J.-R. (2022). Deep Recurrent Encoder: A scalable end-to-end network to model brain signals. Neurons, Behavior, Data Analysis and Theory, 1.

Chiarion, G., Sparacino, L., Antonacci, Y., Faes, L., and Mesin, L. (2023). Connectivity analysis in eeg data: A tutorial review of the state of the art and emerging trends. Bioengineering, 10(3):372.

Chollet, F. (2019). On the measure of intelligence. arXiv preprint arXiv:1911.01547.

Cichy, R. M., Khosla, A., Pantazis, D., and Oliva, A. (2017). Dynamics of scene representations in the human brain revealed by magnetoencephalography and deep neural networks. NeuroImage, 153:346–358.

Cichy, R. M., Khosla, A., Pantazis, D., Torralba, A., and Oliva, A. (2016). Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. Scientific Reports, 6(1):1–13.

Cichy, R. M. and Pantazis, D. (2017). Multivariate pattern analysis of meg and eeg: A comparison of representational structure in time and space. NeuroImage, 158:441–454.

Cichy, R. M., Pantazis, D., and Oliva, A. (2014). Resolving human object recognition in space and time. Nature neuroscience, 17(3):455–462.

Cohen, D. (1968). Magnetoencephalography: evidence of magnetic fields produced by alpha-rhythm currents. Science, 161(3843):784–786.

Cohen, M. X. (2014). Analyzing neural time series data: theory and practice. MIT press.

Cooney, C., Folli, R., and Coyle, D. (2019a). Optimizing layers improves CNN generalization and transfer learning for imagined speech decoding from EEG. In 2019 IEEE international conference on systems, man and cybernetics (SMC), pages 1311–1316. IEEE.

Cooney, C., Korik, A., Raffaella, F., and Coyle, D. (2019b). Classification of imagined spoken word-pairs using convolutional neural networks. In The 8th Graz BCI Conference, 2019, pages 338–343. Verlag der Technischen Universitat Graz.

Csaky, R., van Es, M. W., Jones, O. P., and Woolrich, M. (2023a). Interpretable many-class decoding for meg. NeuroImage, 282:120396.

Csaky, R., van Es, M. W. J., Jones, O. P., and Woolrich, M. (2023b). Group-level brain decoding with deep learning. Human Brain Mapping, 44(17):6105–6119.

Dash, D., Ferrari, P., and Wang, J. (2020a). Decoding imagined and spoken phrases from non-invasive neural (MEG) signals. Frontiers in Neuroscience, 14:290.

Dash, D., Ferrari, P., and Wang, J. (2020b). Decoding Speech Evoked Jaw Motion from Non-invasive Neuromagnetic Oscillations. In 2020 International Joint Conference on Neural Networks (IJCNN), pages 1–8. IEEE.

Dash, D., Wisler, A., Ferrari, P., and Wang, J. (2019). Towards a Speaker Independent Speech-BCI Using Speaker Adaptation. In INTERSPEECH, pages 864–868.

Dayan, P. and Abbott, L. F. (2005). Theoretical neuroscience: computational and mathematical modeling of neural systems. MIT press.

Deco, G., Jirsa, V. K., Robinson, P. A., Breakspear, M., and Friston, K. (2008). The dynamic brain: from spiking neurons to neural masses and cortical fields. PLoS computational biology, 4(8):e1000092.

Défossez, A., Caucheteux, C., Rapin, J., Kabeli, O., and King, J.-R. (2022). Decoding speech from non-invasive brain recordings. arXiv preprint arXiv:2208.12266.

Delorme, A. and Makeig, S. (2004). Eeglab: an open source toolbox for analysis of single-trial eeg dynamics including independent component analysis. Journal of neuroscience methods, 134(1):9–21.

Demanuele, C., James, C. J., and Sonuga-Barke, E. J. (2007). Distinguishing low frequency oscillations within the 1/f spectral behaviour of electromagnetic brain signals. Behavioral and Brain Functions, 3(1):1–14.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019a). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proc. of NAACL.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019b). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dhariwal, P., Jun, H., Payne, C., Kim, J. W., Radford, A., and Sutskever, I. (2020). Jukebox: A generative model for music. arXiv preprint arXiv:2005.00341.

Dikker, S., Assaneo, M. F., Gwilliams, L., Wang, L., and Kösem, A. (2020). Magnetoencephalography and language. Neuroimaging Clinics, 30(2):229–238.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Drewes, J., Muschter, E., Zhu, W., and Melcher, D. (2022). Individual resting-state alpha peak frequency and within-trial changes in alpha peak frequency both predict visual dual-pulse segregation performance. Cerebral Cortex.

Elango, V., Patel, A. N., Miller, K. J., and Gilja, V. (2017). Sequence transfer learning for neural decoding. bioRxiv, page 210732.

Engel, A. K. and Fries, P. (2010). Beta-band oscillations—signalling the status quo? Current opinion in neurobiology, 20(2):156–165.

Faisal, A. A., Selen, L. P., and Wolpert, D. M. (2008). Noise in the nervous system. Nature reviews neuroscience, 9(4):292–303.

Farwell, L. A. and Donchin, E. (1988). Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. Electroencephalography and clinical Neurophysiology, 70(6):510–523.

Fatourechi, M., Bashashati, A., Ward, R. K., and Birch, G. E. (2007). Emg and eog artifacts in brain computer interface systems: A survey. Clinical neurophysiology, 118(3):480–494.

Fedus, W., Zoph, B., and Shazeer, N. (2022). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. The Journal of Machine Learning Research, 23(1):5232–5270.

Fiedler, P., Fonseca, C., Supriyanto, E., Zanow, F., and Haueisen, J. (2022). A high-density 256-channel cap for dry electroencephalography. Human brain mapping, 43(4):1295–1308.

Fisher, A., Rudin, C., and Dominici, F. (2019). All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. Journal of Machine Learning Research, 20(177):1–81.

Forney, G. D. (1973). The viterbi algorithm. Proceedings of the IEEE, 61(3):268–278.

Friederici, A. D. (2011). The brain basis of language processing: from structure to function. Physiological reviews, 91(4):1357–1392.

Friston, K. (2010). The free-energy principle: a unified brain theory? Nature reviews neuroscience, 11(2):127–138.

Friston, K. J. (2005). Models of brain function in neuroimaging. Annu. Rev. Psychol., 56:57–87.

Fukushima, K. and Miyake, S. (1982). Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. Pattern recognition, 15(6):455–469.

Geirhos, R., Temme, C. R., Rauber, J., Schütt, H. H., Bethge, M., and Wichmann, F. A. (2018). Generalisation in humans and deep neural networks. Advances in neural information processing systems, 31.

Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. Neural computation, 4(1):1–58.

Geva, S. (2018). Inner speech and mental imagery. Inner speech: New voices, pages 1–32.

Ghahramani, Z. (2003). Unsupervised learning. In Summer school on machine learning, pages 72–112. Springer.

Gifford, A. T., Dwivedi, K., Roig, G., and Cichy, R. M. (2022). A large and rich eeg dataset for modeling human visual object recognition. NeuroImage, 264:119754.

Gohil, C., Huang, R., Roberts, E., van Es, M. W., Quinn, A. J., Vidaurre, D., and Woolrich, M. W. (2023). osl-dynamics: A toolbox for modelling fast dynamic brain activity. bioRxiv, pages 2023–08.

Gohil, C., Roberts, E., Timms, R., Skates, A., Higgins, C., Quinn, A., Pervaiz, U., van Amersfoort, J., Notin, P., Gal, Y., et al. (2022). Mixtures of large-scale dynamic functional brain network modes. NeuroImage, 263:119595.

Goldberg, D. E. and Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. In Foundations of genetic algorithms, volume 1, pages 69–93. Elsevier.

Good, I. J. (1952). Rational decisions. Journal of the Royal Statistical Society: Series B (Methodological), 14(1):107–114.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep Learning. MIT Press.

Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Goj, R., Jas, M., Brooks, T., Parkkonen, L., et al. (2013). MEG and EEG data analysis with MNE-Python. Frontiers in neuroscience, page 267.

Grootswagers, T., Wardle, S. G., and Carlson, T. A. (2017). Decoding dynamic brain patterns from evoked responses: A tutorial on multivariate pattern analysis applied to time series neuroimaging data. Journal of cognitive neuroscience, 29(4):677–697.

Grootswagers, T., Wardle, S. G., and Carlson, T. A. (2020). Decoding dynamic brain patterns from evoked responses: A tutorial on multivariate pattern analysis applied to time series neuroimaging data. Journal of cognitive neuroscience, 32(4):677–697.

Gross, J., Baillet, S., Barnes, G. R., Henson, R. N., Hillebrand, A., Jensen, O., Jerbi, K., Litvak, V., Maess, B., Oostenveld, R., et al. (2013). Good practice for conducting and reporting meg research. Neuroimage, 65:349–363.

Gruber, T., Müller, M. M., Keil, A., and Elbert, T. (1999). Selective visual-spatial attention alters induced gamma band responses in the human eeg. Clinical neurophysiology, 110(12):2074–2085.

Guan, C., Thulasidas, M., and Wu, J. (2004). High performance p300 speller for brain-computer interface. In IEEE International Workshop on Biomedical Circuits and Systems, 2004., pages S3–5. IEEE.

Guggenmos, M., Sterzer, P., and Cichy, R. M. (2018). Multivariate pattern analysis for MEG: A comparison of dissimilarity measures. Neuroimage, 173:434–447.

Hadida, J., Sotiropoulos, S. N., Abeysuriya, R. G., Woolrich, M. W., and Jbabdi, S. (2018). Bayesian optimisation of large-scale biophysical networks. Neuroimage, 174:219–236.

Halme, H.-L. and Parkkonen, L. (2018). Across-subject offline decoding of motor imagery from MEG and EEG. Scientific reports, 8(1):1–12.

Hämäläinen, M., Hari, R., Ilmoniemi, R. J., Knuutila, J., and Lounasmaa, O. V. (1993). Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain. Reviews of modern Physics, 65(2):413.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. Nature, 585(7825):357–362.

Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. Journal of the royal statistical society. series c (applied statistics), 28(1):100–108.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction, volume 2. Springer.

Haufe, S., Meinecke, F., Görgen, K., Dähne, S., Haynes, J.-D., Blankertz, B., and Bießmann, F. (2014). On the interpretation of weight vectors of linear models in multivariate neuroimaging. Neuroimage, 87:96–110.

Haxby, J. V., Gobbini, M. I., Furey, M. L., Ishai, A., Schouten, J. L., and Pietrini, P. (2001). Distributed and overlapping representations of faces and objects in ventral temporal cortex. Science, 293(5539):2425–2430.

Haynes, J.-D. and Rees, G. (2006). Decoding mental states from brain activity in humans. Nature reviews neuroscience, 7(7):523–534.

Hebart, M. N., Contier, O., Teichmann, L., Rockter, A., Zheng, C. Y., Kidder, A., Corriveau, A., Vaziri-Pashkam, M., and Baker, C. I. (2022). Things-data: A multimodal collection of large-scale datasets for investigating object representations in human brain and behavior. bioRxiv, pages 2022–07.

Herrmann, C. S., Fründ, I., and Lenz, D. (2010). Human gamma-band activity: a review on cognitive and behavioral correlates and network models. Neuroscience & Biobehavioral Reviews, 34(7):981–992.

Hickok, G. and Poeppel, D. (2007). The cortical organization of speech processing. Nature reviews neuroscience, 8(5):393–402.

Higgins, C., Vidaurre, D., Kolling, N., Liu, Y., Behrens, T., and Woolrich, M. (2022a). Spatiotemporally resolved multivariate pattern analysis for m/eeg. Human Brain Mapping, 43(10):3062–3085.

Higgins, C. J., van Es, M. W., Quinn, A. J., Vidaurre, D., and Woolrich, M. W. (2022b). The relationship between frequency content and representational dynamics in the decoding of neurophysiological data. bioRxiv.

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., and Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition. IEEE Signal Processing Magazine, 29:82–97.

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

Holdgraf, C. R., Rieger, J. W., Micheli, C., Martin, S., Knight, R. T., and Theunissen, F. E. (2017). Encoding and decoding models in cognitive electrophysiology. Frontiers in systems neuroscience, 11:61.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. (2020). The curious case of neural text degeneration. In International Conference on Learning Representations.

Hornik, K., Stinchcombe, M., and White, H. (1989a). Multilayer feedforward networks are universal approximators. Neural networks, 2(5):359–366.

Hornik, K., Stinchcombe, M., White, H., et al. (1989b). Multilayer feedforward networks are universal approximators. Neural networks, 2(5):359–366.

Hu, D., Li, W., and Chen, X. (2011). Feature extraction of motor imagery eeg signals based on wavelet packet decomposition. In The 2011 IEEE/ICME international conference on complex medical engineering, pages 694–697. IEEE.

Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of physiology, 160(1):106.

Hultén, A., van Vliet, M., Kivisaari, S., Lammi, L., Lindh-Knuutila, T., Faisal, A., and Salmelin, R. (2021). The neural representation of abstract words may arise through grounding word meaning in language itself. Human brain mapping, 42(15):4973–4984.

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing In Science & Engineering, 9(3):90–95.

Hyv"arinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. IEEE transactions on Neural Networks, 10(3):626–634.

Hyvarinen, A. and Oja, E. (2001). Independent component analysis: algorithms and applications. Neural networks, 13(4-5):411–430.

Insel, T. R. and Landis, S. C. (2013). Twenty-five years of progress: the view from nimh and ninds. Neuron, 80(3):561–567.

İşcan, Z. and Nikulin, V. V. (2018). Steady state visual evoked potential (ssvep) based brain-computer interface (bci) performance under different perturbations. PloS one, 13(1):e0191673.

Izhikevich, E. M. (2004). Which model to use for cortical spiking neurons? IEEE transactions on neural networks, 15(5):1063–1070.

Izhikevich, E. M. (2007). Dynamical systems in neuroscience. MIT press.

Jensen, O. and Mazaheri, A. (2010). Shaping functional architecture by oscillatory alpha activity: gating by inhibition. Frontiers in human neuroscience, 4:186.

Jorntell, H. and Kesgin, K. (2023). Singular superlet transform achieves markedly improved time-frequency super-resolution for separating complex neural signals. bioRxiv, pages 2023–02.

Jung, T.-P., Makeig, S., Humphries, C., Lee, T.-W., Mckeown, M. J., Iragui, V., and Sejnowski, T. J. (2000). Removing electroencephalographic artifacts by blind source separation. Psychophysiology, 37(2):163–178.

Kanai, R. and Rees, G. (2011). The structural basis of inter-individual differences in human behaviour and cognition. Nature Reviews Neuroscience, 12(4):231–242.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models. arXiv preprint arXiv:2001.08361.

Kay, K. N., Naselaris, T., Prenger, R. J., and Gallant, J. L. (2008). Identifying natural images from human brain activity. Nature, 452(7185):352–355.

Kiebel, S. J., Garrido, M. I., Moran, R. J., and Friston, K. J. (2008). Dynamic causal modelling for eeg and meg. Cognitive neurodynamics, 2:121–136.

Kietzmann, T. C., Spoerer, C. J., Sörensen, L. K., Cichy, R. M., Hauk, O., and Kriegeskorte, N. (2019). Recurrence is required to capture the representational dynamics of the human visual system. Proceedings of the National Academy of Sciences, 116(43):21854–21863.

King, J.-R. and Dehaene, S. (2014). Characterizing the dynamics of mental representations: the temporal generalization method. Trends in cognitive sciences, 18(4):203–210.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Kitaev, N., Kaiser, Ł., and Levskaya, A. (2020). Reformer: The efficient transformer. arXiv preprint arXiv:2001.04451.

Klimesch, W. (1999). Eeg alpha and theta oscillations reflect cognitive and memory performance: a review and analysis. Brain research reviews, 29(2-3):169–195.

Klimesch, W. (2012). Alpha-band oscillations, attention, and controlled access to stored information. Trends in cognitive sciences, 16(12):606–617.

Klimesch, W., Sauseng, P., and Hanslmayr, S. (2007). Eeg alpha oscillations: the inhibition–timing hypothesis. Brain research reviews, 53(1):63–88.

Kobak, D., Brendel, W., Constantinidis, C., Feierstein, C. E., Kepecs, A., Mainen, Z. F., Qi, X.-L., Romo, R., Uchida, N., and Machens, C. K. (2016). Demixed principal component analysis of neural population data. elife, 5:e10989.

Koizumi, K., Ueda, K., and Nakao, M. (2018). Development of a cognitive brain-machine interface based on a visual imagery method. In 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pages 1062–1065. IEEE.

Kosslyn, S. M. and Pylyshyn, Z. (1994). Image and brain: The resolution of the imagery debate. Nature, 372(6503):289–289.

Kostas, D., Aroca-Ouellette, S., and Rudzicz, F. (2021). Bendr: using transformers and a contrastive self-supervised learning task to learn from massive amounts of eeg data. Frontiers in Human Neuroscience, page 253.

Kostas, D. and Rudzicz, F. (2020). Thinker invariance: enabling deep neural networks for bci across more people. Journal of Neural Engineering, 17(5):056008.

Kriegeskorte, N. (2015). Deep neural networks: a new framework for modeling biological vision and brain information processing. Annual review of vision science, 1:417–446.

Kriegeskorte, N. and Douglas, P. K. (2018). Cognitive computational neuroscience. Nature neuroscience, 21(9):1148–1160.

Kriegeskorte, N., Mur, M., and Bandettini, P. A. (2008). Representational similarity analysis-connecting the branches of systems neuroscience. Frontiers in Systems Neuroscience, 2:4.

Krizhevsky, A. and Sutskever, I.and Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. In NIPS’2012.

Kubilius, J., Schrimpf, M., Kar, K., Rajalingham, R., Hong, H., Majaj, N., Issa, E., Bashivan, P., Prescott-Roy, J., Schmidt, K., et al. (2019). Brain-like object recognition with high-performing shallow recurrent anns. Advances in neural information processing systems, 32.

Kurth-Nelson, Z., Economides, M., Dolan, R. J., and Dayan, P. (2016). Fast sequences of non-spatial state representations in humans. Neuron, 91(1):194–204.

Kutas, M. and Van Petten, C. (1988). Event-related brain potential studies of language. Advances in psychophysiology, 3:139–187.

Lappe, C., Steinsträter, O., and Pantev, C. (2013). A beamformer analysis of meg data reveals frontal generators of the musically elicited mismatch negativity. PLoS One, 8(4):e61296.

Lawhern, V. J., Solon, A. J., Waytowich, N. R., Gordon, S. M., Hung, C. P., and Lance, B. J. (2018). Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. Journal of neural engineering, 15(5):056013.

Lebedev, M. A. and Nicolelis, M. A. (2006). Brain–machine interfaces: past, present and future. TRENDS in Neurosciences, 29(9):536–546.

Lemm, S., Blankertz, B., Dickhaus, T., and Müller, K.-R. (2011). Introduction to machine learning for brain imaging. Neuroimage, 56(2):387–399.

Letheren, K., Russell-Bennett, R., and Whittaker, L. (2020). Black, white or grey magic? our future with artificial intelligence. Journal of Marketing Management, 36(3-4):216–232.

Lewis, M. and MTSA, S. (1997). A-law and mu-law companding implementations using the tms320c54x. Application Note SPRA163A, Texas Instrum., Dallas, TX, USA.

Li, J., Galley, M., Brockett, C., Spithourakis, G. P., Gao, J., and Dolan, B. (2016). A Persona-Based Neural Conversation Model. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 994–1003. Association for Computational Linguistics.

Li, J., Pan, J., Wang, F., and Yu, Z. (2021). Inter-Subject MEG Decoding for Visual Information with Hybrid Gated Recurrent Network. Applied Sciences, 11(3):1215.

Linardatos, P., Papastefanopoulos, V., and Kotsiantis, S. (2020). Explainable ai: A review of machine learning interpretability methods. Entropy, 23(1):18.

Ling, S., Lee, A. C., Armstrong, B. C., and Nestor, A. (2019). How are visual words represented? insights from eeg-based visual word decoding, feature derivation and image reconstruction. Human brain mapping, 40(17):5056–5068.

Link, A., Elster, C., Sander, T., Lueschow, A., Curio, G., and Trahms, L. (2002). Meg-analysis using the hilbert transform.

Liu, S., Bremer, P.-T., Thiagarajan, J. J., Srikumar, V., Wang, B., Livnat, Y., and Pascucci, V. (2017). Visual exploration of semantic relationships in neural word embeddings. IEEE transactions on visualization and computer graphics, 24(1):553–562.

Liu, S., Lu, H., and Shao, J. (2015). Improved residual vector quantization for high-dimensional approximate nearest neighbor search. arXiv preprint arXiv:1509.05195.

Liu, Y., Dolan, R. J., Kurth-Nelson, Z., and Behrens, T. E. (2019). Human replay spontaneously reorganizes experience. Cell, 178(3):640–652.

Livingstone, M. and Hubel, D. (1988). Segregation of form, color, movement, and depth: anatomy, physiology, and perception. Science, 240(4853):740–749.

Long, M., Cao, Y., Wang, J., and Jordan, M. (2015). Learning transferable features with deep adaptation networks. In International conference on machine learning, pages 97–105. PMLR.

Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., and Yger, F. (2018). A review of classification algorithms for eeg-based brain–computer interfaces: a 10 year update. Journal of neural engineering, 15(3):031005.

Luck, S. J. (2014). An introduction to the event-related potential technique. MIT press.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. Journal of machine learning research, 9(Nov):2579–2605.

Makeig, S., Bell, A., Jung, T.-P., and Sejnowski, T. J. (1995). Independent component analysis of electroencephalographic data. Advances in neural information processing systems, 8.

Makeig, S., Bell, A. J., Jung, T. P., and Sejnowski, T. J. (1999). Independent component analysis of electroencephalographic data. Advances in neural information processing systems, pages 145–151.

Mallat, S. (1999). A wavelet tour of signal processing. Elsevier.

Mantini, D., Perrucci, M. G., Del Gratta, C., Romani, G. L., and Corbetta, M. (2007). Electrophysiological signatures of resting state networks in the human brain. Proceedings of the National Academy of Sciences, 104(32):13170–13175.

Martin, S., Brunner, P., Iturrate, I., Millán, J. d. R., Schalk, G., Knight, R. T., and Pasley, B. N. (2016). Word pair classification during imagined speech using direct brain recordings. Scientific reports, 6(1):25803.

Martin, S., Iturrate, I., Millán, J. d. R., Knight, R. T., and Pasley, B. N. (2018). Decoding inner speech using electrocorticography: Progress and challenges toward a speech prosthesis. Frontiers in neuroscience, 12:422.

Maunsell, J. H. and Newsome, W. T. (1987). Visual processing in monkey extrastriate cortex. Annual review of neuroscience, 10(1):363–401.

Metzger, S. L., Liu, J. R., Moses, D. A., Dougherty, M. E., Seaton, M. P., Littlejohn, K. T., Chartier, J., Anumanchipalli, G. K., Tu-Chan, A., Ganguly, K., et al. (2022). Generalizable spelling using a speech neuroprosthesis in an individual with severe limb and vocal paralysis. Nature Communications, 13(1):6510.

Meyes, R., Lu, M., de Puiseau, C. W., and Meisen, T. (2019). Ablation studies in artificial neural networks. arXiv preprint arXiv:1901.08644.

Michalke, L., Dreyer, A. M., Borst, J. P., and Rieger, J. W. (2023). Inter-individual single-trial classification of meg data using m-cca. NeuroImage, 273:120079.

Michel, C. M. and Brunet, D. (2019). Eeg source imaging: a practical review of the analysis steps. Frontiers in neurology, 10:325.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. In Bengio, Y. and LeCun, Y., editors, 1st International Conference on Learning Representations, ICLR 2013, Workshop Track Proceedings.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed Representations of Words and Phrases and their Compositionality. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, Advances in Neural Information Processing Systems 26, pages 3111–3119. Curran Associates, Inc.

Miller, K. J., Sorensen, L. B., Ojemann, J. G., and Den Nijs, M. (2009). Power-law scaling in the brain surface electric potential. PLoS computational biology, 5(12):e1000609.

Moca, V. V., Bârzan, H., Nagy-Dăbâcan, A., and Mureșan, R. C. (2021). Time-frequency super-resolution with superlets. Nature communications, 12(1):337.

Morin, A. (2005). Possible links between self-awareness and inner speech theoretical background, underlying mechanisms, and empirical evidence. Journal of Consciousness Studies, 12(4-5):115–134.

Moses, D. A., Leonard, M. K., Makin, J. G., and Chang, E. F. (2019). Real-time decoding of question-and-answer speech dialogue using human cortical activity. Nature Communications, 10(1):1–14.

Mridha, M. F., Ohi, A. Q., Monowar, M. M., Hamid, M., Islam, M., Watanobe, Y., et al. (2021). U-vectors: Generating clusterable speaker embedding from unlabeled data. Applied Sciences, 11(21):10079.

Mugler, E. M., Patton, J. L., Flint, R. D., Wright, Z. A., Schuele, S. U., Rosenow, J., Shih, J. J., Krusienski, D. J., and Slutzky, M. W. (2014). Direct classification of all American English phonemes using signals from functional speech motor cortex. Journal of Neural Engineering, 11(3):035015.

Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., and Yu, B. (2019). Definitions, methods, and applications in interpretable machine learning. Proceedings of the National Academy of Sciences, 116(44):22071–22080.

Murguialday, A. R., Hill, J., Bensch, M., Martens, S., Halder, S., Nijboer, F., Schoelkopf, B., Birbaumer, N., and Gharabaghi, A. (2011). Transition from the locked in to the completely locked-in state: a physiological analysis. Clinical Neurophysiology, 122(5):925–933.

Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT press.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In ICML.

Naseer, N. and Hong, K.-S. (2015). fnirs-based brain-computer interfaces: a review. Frontiers in human neuroscience, 9:3.

Naselaris, T., Kay, K. N., Nishimoto, S., and Gallant, J. L. (2011). Encoding and decoding in fmri. Neuroimage, 56(2):400–410.

Nasiotis, K., Clavagnier, S., Baillet, S., and Pack, C. C. (2017). High-resolution retinotopic maps estimated with magnetoencephalography. NeuroImage, 145:107–117.

Nenonen, J., Taulu, S., Kajola, M., and Ahonen, A. (2007). Total information extracted from meg measurements. In International Congress Series, volume 1300, pages 245–248. Elsevier.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. (2022). A time series is worth 64 words: Long-term forecasting with transformers. arXiv preprint arXiv:2211.14730.

Nunez, P. L. (2000). Toward a quantitative description of large-scale neocortical dynamic function and eeg. Behavioral and Brain Sciences, 23(3):371–398.

Nunez, P. L. and Srinivasan, R. (2006). Electric fields of the brain: the neurophysics of EEG. Oxford University Press, USA.

Obleser, J. and Kotz, S. A. (2011). Multiple brain signatures of integration in the comprehension of degraded speech. Neuroimage, 55(2):713–723.

Olivetti, E., Kia, S. M., and Avesani, P. (2014). MEG decoding across subjects. In 2014 International Workshop on Pattern Recognition in Neuroimaging, pages 1–4. IEEE.

Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748.

Panachakel, J. T. and Ramakrishnan, A. G. (2021). Decoding covert speech from eeg-a comprehensive review. Frontiers in Neuroscience, 15:392.

Panahi, M. R., Abrevaya, G., Gagnon-Audet, J.-C., Voleti, V., Rish, I., and Dumas, G. (2021). Generative models of brain dynamics–a review. arXiv preprint arXiv:2112.12147.

Parker Jones, O. and Voets, N. L. (2021). A note on decoding elicited and self-generated inner speech. bioRxiv.

Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., and Tran, D. (2018). Image transformer. In International conference on machine learning, pages 4055–4064. PMLR.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems 32, pages 8024–8035. Curran Associates, Inc.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830.

Peelle, J. E. and Davis, M. H. (2012). Neural oscillations carry speech rhythm through to comprehension. Frontiers in psychology, 3:320.

Pessoa, L. (2022). Emergent processes in cognitive-emotional interactions. Dialogues in clinical neuroscience.

Poldrack, R. A. and Gorgolewski, K. J. (2014). Making big data open: data sharing in neuroimaging. Nature neuroscience, 17(11):1510–1517.

Poldrack, R. A., Halchenko, Y. O., and Hanson, S. J. (2009). Decoding the large-scale structure of brain function by classifying mental states across individuals. Psychological science, 20(11):1364–1372.

Quinn, A. J., Atkinson, L. Z., Gohil, C., Kohl, O., Pitt, J., Zich, C., Nobre, A. C., and Woolrich, M. W. (2022a). The glm-spectrum: A multilevel framework for spectrum analysis with covariate and confound modelling. bioRxiv, pages 2022–11.

Quinn, A. J., van Ede, F., Brookes, M. J., Heideman, S. G., Nowak, M., Seedat, Z. A., Vidaurre, D., Zich, C., Nobre, A. C., and Woolrich, M. W. (2019). Unpacking transient event dynamics in electrophysiological power spectra. Brain topography, 32(6):1020–1034.

Quinn, A. J., van Es, M., Gohil, C., and Woolrich, M. W. (2022b). Ohba software library in python (osl).

Rabiner, L. R. (1989a). A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2):257–286.

Rabiner, L. R. (1989b). A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2):257–286.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In International conference on machine learning, pages 8748–8763. PMLR.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. (2022). Robust speech recognition via large-scale weak supervision.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8):9.

Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., and Sohl-Dickstein, J. (2017). On the expressive power of deep neural networks. In international conference on machine learning, pages 2847–2854. PMLR.

Raichle, M. E., MacLeod, A. M., Snyder, A. Z., Powers, W. J., Gusnard, D. A., and Shulman, G. L. (2001). A default mode of brain function. Proceedings of the national academy of sciences, 98(2):676–682.

Ramkumar, P., Jas, M., Pannasch, S., Hari, R., and Parkkonen, L. (2013). Feature-specific information processing precedes concerted activation in human visual cortex. Journal of Neuroscience, 33(18):7691–7699.

Rauber, P. E., Fadel, S. G., Falcao, A. X., and Telea, A. C. (2016). Visualizing the hidden activity of artificial neural networks. IEEE transactions on visualization and computer graphics, 23(1):101–110.

Ravishankar, S., Toneva, M., and Wehbe, L. (2021). Single-trial meg data can be denoised through cross-subject predictive modeling. Frontiers in Computational Neuroscience, 15:737324.

Richards, B. A., Lillicrap, T. P., Beaudoin, P., Bengio, Y., Bogacz, R., Christensen, A., Clopath, C., Costa, R. P., de Berker, A., Ganguli, S., et al. (2019). A deep learning framework for neuroscience. Nature Neuroscience, 22(11):1761–1770.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical Report ICS 8504, Institute for Cognitive Science, University of California, San Diego, California.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088):533–536.

Saha, S. and Baumert, M. (2020). Intra-and inter-subject variability in eeg-based sensorimotor brain computer interface: a review. Frontiers in computational neuroscience, 13:87.

Saito, Y., Takamichi, S., and Saruwatari, H. (2019). DNN-based speaker embedding using subjective inter-speaker similarity for multi-speaker modeling in speech synthesis. arXiv preprint arXiv:1907.08294.

Sakkalis, V. (2011). Review of advanced techniques for the estimation of brain connectivity measured with eeg/meg. Computers in biology and medicine, 41(12):1110–1117.

Samek, W., Montavon, G., Vedaldi, A., Hansen, L. K., and Müller, K.-R. (2019). Explainable AI: interpreting, explaining and visualizing deep learning, volume 11700. Springer Nature.

Saxe, A. M., McClelland, J. L., and Ganguli, S. (2013). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. arXiv preprint arXiv:1312.6120.

Schirrmeister, R. T., Springenberg, J. T., Fiederer, L. D. J., Glasstetter, M., Eggensperger, K., Tangermann, M., Hutter, F., Burgard, W., and Ball, T. (2017a). Deep learning with convolutional neural networks for eeg decoding and visualization. Human brain mapping, 38(11):5391–5420.

Schirrmeister, R. T., Springenberg, J. T., Fiederer, L. D. J., Glasstetter, M., Eggensperger, K., Tangermann, M., Hutter, F., Burgard, W., and Ball, T. (2017b). Deep learning with convolutional neural networks for eeg decoding and visualization. Human brain mapping, 38(11):5391–5420.

Schlögl, A. and Supp, G. (2006). Analyzing event-related eeg data with multivariate autoregressive parameters. Progress in brain research, 159:135–147.

Schultz, W. (2002). Getting formal with dopamine and reward. Neuron, 36(2):241–263.

Schulz, H. (2008). Rethinking sleep analysis: Comment on the aasm manual for the scoring of sleep and associated events. Journal of Clinical Sleep Medicine, 4(2):99–103.

Shen, G., Horikawa, T., Majima, K., and Kamitani, Y. (2019). Deep image reconstruction from human brain activity. PLoS computational biology, 15(1):e1006633.

Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034.

Singer, W. (1999). Neuronal synchrony: a versatile code for the definition of relations? Neuron, 24(1):49–65.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1):1929–1958.

Sturm, I., Lapuschkin, S., Samek, W., and Müller, K.-R. (2016). Interpretable deep neural networks for single-trial eeg classification. Journal of neuroscience methods, 274:141–145.

Su, L., Fonteneau, E., Marslen-Wilson, W., and Kriegeskorte, N. (2012). Spatiotemporal searchlight representational similarity analysis in emeg source space in: 2012 second international workshop on pattern recognition in neuroimaging.

Sutton, R. (2019). The bitter lesson. Incomplete Ideas (blog), 13(1).

Sutton, R. S., Barto, A. G., et al. (1998). Introduction to reinforcement learning, volume 135. MIT press Cambridge.

Tabar, Y. R. and Halici, U. (2016). A novel deep learning approach for classification of eeg motor imagery signals. Journal of neural engineering, 14(1):016003.

Takalo, R., Hytti, H., and Ihalainen, H. (2005). Tutorial on univariate autoregressive spectral analysis. Journal of clinical monitoring and computing, 19:401–410.

Tallon-Baudry, C. and Bertrand, O. (1999). Oscillatory gamma activity in humans and its role in object representation. Trends in cognitive sciences, 3(4):151–162.

Tanaka, K. (1996). Inferotemporal cortex and object vision. Annual review of neuroscience, 19(1):109–139.

Taulu, S. and Simola, J. (2006). Spatiotemporal signal space separation method for rejecting nearby interference in meg measurements. Physics in Medicine & Biology, 51(7):1759.

Teplan, M. et al. (2002). Fundamentals of eeg measurement. Measurement science review, 2(2):1–11.

Tierney, T. M., Alexander, N., Mellor, S., Holmes, N., Seymour, R., O'Neill, G. C., Maguire, E. A., and Barnes, G. R. (2021). Modelling optically pumped magnetometer interference in meg as a spatially homogeneous magnetic field. NeuroImage, 244:118484.

Timms, R. (2022). Time-varying source reconstruction. PhD thesis, University of Oxford.

Torrey, L. and Shavlik, J. (2010). Transfer learning. In Handbook of research on machine learning applications and trends: algorithms, methods, and techniques, pages 242–264. IGI global.

Tzovara, A., Murray, M. M., Michel, C. M., and De Lucia, M. (2012). A tutorial review of electrical neuroimaging from group-average to single-trial event-related potentials. Developmental neuropsychology, 37(6):518–544.

Urigüen, J. A. and Garcia-Zapirain, B. (2015). Eeg artifact removal—state-of-the-art and guidelines. Journal of neural engineering, 12(3):031001.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). WaveNet: A Generative Model for Raw Audio. In Proc. 9th ISCA Workshop on Speech Synthesis Workshop (SSW 9), page 125.

Van Den Oord, A., Vinyals, O., et al. (2017). Neural discrete representation learning. Advances in neural information processing systems, 30.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. Journal of machine learning research, 9(11).

Van Dijk, K. R., Sabuncu, M. R., and Buckner, R. L. (2012). The influence of head motion on intrinsic functional connectivity mri. Neuroimage, 59(1):431–438.

van Vliet, M. and Salmelin, R. (2020). Post-hoc modification of linear models: Combining machine learning with domain information to make solid inferences from noisy data. NeuroImage, 204:116221.

Vapnik, V. (1999). The nature of statistical learning theory. Springer science & business media.

Varoquaux, G., Raamana, P. R., Engemann, D. A., Hoyos-Idrobo, A., Schwartz, Y., and Thirion, B. (2017). Assessing and tuning brain decoders: cross-validation, caveats, and guidelines. NeuroImage, 145:166–179.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, Advances in Neural Information Processing Systems 30, pages 5998–6008. Curran Associates, Inc.

Vidaurre, C., Quinn, A. J., Baker, A. P., Dupret, D., Tejero-Cantero, A., and Woolrich, M. W. (2018a). Discovering dynamic brain networks from big data in rest and task. Neuroimage, 180:646–656.

Vidaurre, D., Abeysuriya, R., Becker, R., Quinn, A. J., Alfaro-Almagro, F., Smith, S. M., and Woolrich, M. W. (2018b). Discovering dynamic brain networks from big data in rest and task. NeuroImage, 180:646–656.

Vidaurre, D., Hunt, L. T., Quinn, A. J., Hunt, B. A., Brookes, M. J., Nobre, A. C., and Woolrich, M. W. (2018c). Spontaneous cortical activity transiently organises into frequency specific phase-coupling networks. Nature Communications, 9(1):1–13.

Vig, J. (2019). Bertviz: A tool for visualizing multihead self-attention in the bert model. In ICLR workshop: Debugging machine learning models, volume 23.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17:261–272.

Vorwerk, J., Cho, J.-H., Rampp, S., Hamer, H., Knösche, T. R., and Wolters, C. H. (2014). A guideline for head volume conductor modeling in eeg and meg. NeuroImage, 100:590–607.

Wainio-Theberge, S., Wolff, A., and Northoff, G. (2021). Dynamic relationships between spontaneous and evoked electrophysiological activity. Communications Biology, 4(1):741.

Waldert, S. (2016). Invasive vs. non-invasive neuronal signals for brain-machine interfaces: will one prevail? Frontiers in neuroscience, 10:295.

Wandell, B. A., Dumoulin, S. O., and Brewer, A. A. (2007). Visual field maps in human cortex. Neuron, 56(2):366–383.

Wandelt, S. K., Bjanes, D., Pejsa, K., Lee, B., Liu, C., and Andersen, R. A. (2022). Online internal speech decoding from single neurons in a human participant. medRxiv, pages 2022–11.

Wang, C., Subramaniam, V., Yaari, A. U., Kreiman, G., Katz, B., Cases, I., and Barbu, A. (2023). Brainbert: Self-supervised representation learning for intracranial recordings. arXiv preprint arXiv:2302.14367.

Wang, K., Gao, X., Zhao, Y., Li, X., Dou, D., and Xu, C.-Z. (2019). Pay attention to features, transfer learn faster CNNs. In International conference on learning representations.

Wang, S., Li, B. Z., Khabsa, M., Fang, H., and Ma, H. (2020). Linformer: Self-attention with linear complexity. arXiv preprint arXiv:2006.04768.

Wang, X.-J. (2010). Neurophysiological and computational principles of cortical rhythms in cognition. Physiological reviews, 90(3):1195–1268.

Waskom, M. L. (2021). seaborn: statistical data visualization. Journal of Open Source Software, 6(60):3021.

Welch, P. (1967). The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. IEEE Transactions on Audio and Electroacoustics, 15(2):70–73.

Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., and Sun, L. (2022). Transformers in time series: A survey. arXiv preprint arXiv:2202.07125.

Wens, V. (2023). Exploring the limits of meg spatial resolution with multipolar expansions. NeuroImage, 270:119953.

Wes McKinney (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, Proceedings of the 9th Python in Science Conference, pages 56 – 61.

Widmann, A., Schröger, E., and Maess, B. (2015). Digital filter design for electrophysiological data–a practical approach. Journal of neuroscience methods, 250:34–46.

Willett, F. R., Avansino, D. T., Hochberg, L. R., Henderson, J. M., and Shenoy, K. V. (2021a). High-performance brain-to-text communication via handwriting. Nature, 593(7858):249–254.

Willett, F. R., Avansino, D. T., Hochberg, L. R., Henderson, J. M., and Shenoy, K. V. (2021b). High-performance brain-to-text communication via handwriting. Nature, 593(7858):249–254.

Wittevrongel, B., Holmes, N., Boto, E., Hill, R., Rea, M., Libert, A., Khachatryan, E., Van Hulle, M. M., Bowtell, R., and Brookes, M. J. (2021). Practical real-time meg-based neural interfacing with optically pumped magnetometers. BMC biology, 19(1):1–15.

Wolpaw, J. R. (2013). Brain–computer interfaces. In Handbook of clinical neurology, volume 110, pages 67–74. Elsevier.

Ying, X. (2019). An overview of overfitting and its solutions. In Journal of physics: Conference series, volume 1168, page 022022. IOP Publishing.

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., and Lipson, H. (2015). Understanding neural networks through deep visualization. arXiv preprint arXiv:1506.06579.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13, pages 818–833. Springer.

Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., and Weston, J. (2018). Personalizing Dialogue Agents: I have a dog, do you have pets too? In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pages 2204–2213. Association for Computational Linguistics.

Zhang, X., Wang, J., Cheng, N., and Xiao, J. (2020). MDCNN-SID: Multi-scale Dilated Convolution Network for Singer Identification. arXiv preprint arXiv:2004.04371.

Zhou, D., Zhang, G., Dang, J., Wu, S., and Zhang, Z. (2020). A multi-subject temporal-spatial hyper-alignment method for eeg-based neural entrainment to speech. In 2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pages 881–887. IEEE.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the AAAI conference on artificial intelligence, volume 35, pages 11106–11115.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. (2020). A comprehensive survey on transfer learning. Proceedings of the IEEE, 109(1):43–76.

Zoefel, B. and VanRullen, R. (2017). Oscillatory mechanisms of stimulus processing and selection in the visual and auditory systems: state-of-the-art, speculations and suggestions. Frontiers in Neuroscience, 11:296.

Zubarev, I., Zetter, R., Halme, H.-L., and Parkkonen, L. (2019). Adaptive neural network classifier for decoding meg signals. Neuroimage, 197:425–434.

A | Interpretable full-epoch decoding

A.1 Results

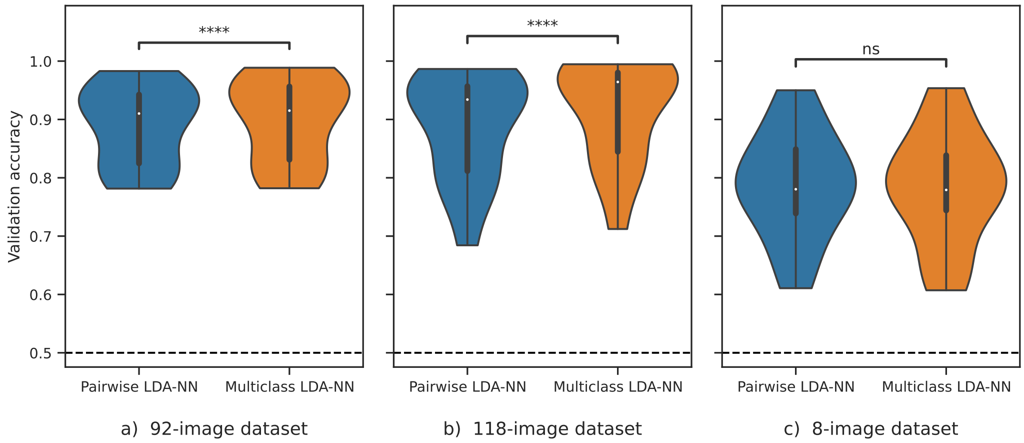
A.1.1 Multiclass versus pairwise decoding

We have demonstrated that multiclass full-epoch models are better than sliding window models while maintaining the same level of spatiotemporal information. Here we wish to highlight an additional advantage of using multiclass full-epoch models. Researchers frequently use pairwise models to analyse the representational differences between individual conditions or groups of conditions, such as in representational similarity analysis (RSA). However, this approach can be computationally intensive, especially when dealing with a large number of classes.

Here, we show how we can utilise a single trained multiclass full-epoch LDA-NN model to predict pairwise accuracy scores. This is done by iteratively taking all pairs of conditions, computing the predicted probabilities across all classes for each trial, and selecting the condition with the higher probability (of the two conditions) as the predicted class. By comparing this to the ground-truth labels, we can obtain pairwise accuracy scores for each pair of conditions i, j:

$$\text{accuracy}_{ij} = \frac{1}{N} \sum_{n=1}^{N} \left[\text{argmax}_{k \in i,j} p(y_n = k \mid \mathbf{X}_n; \theta) \right] == y_n \tag{A.1}$$

where N is the number of trials across these two conditions, $\mathbf{X}_n$ is the feature vector for trial n, y_n is the true class label for trial n, $p(y_n = k \mid \mathbf{X}_n; \theta)$ is the predicted probability of class k for trial n. argmax selects the condition (i or j) with the higher predicted probability for trial n.



a) 92-image dataset b) 118-image dataset c) 8-image dataset

Figure A.1: Comparison of pairwise full-epoch LDA-NN models (blue) with multiclass models evaluated for pairwise classification (orange) across the three datasets. In all datasets except the 8-image dataset, multiclass models evaluated in a pairwise fashion are significantly better (****, p<1e-4). The violin plot distributions are shown over the mean individual subject performance. The dashed line represents chance level.

In Figure A.1, we compared the results of this method with those obtained by training individual pairwise (full epoch LDA-NN) models as is typical in the literature. For the 92 and 118-image datasets, the multiclass model achieved modest, but significant higher pairwise accuracy than the individual pairwise models. The difference was not significant for the 8-image datasets. Therefore, using a multiclass model can yield pairwise results that are similar to or even better than those obtained from individual pairwise models. This provides a much more efficient way of obtaining pairwise accuracies for the purposes of RSA.

This approach is useful for RSA and reduces computation time by approximately half the number of conditions in the data, as pairwise models reuse this data for training while a multiclass model uses it only once. Although the data must still be reused for evaluation, we can assume that evaluation is much faster than training. The slight increase in performance when using multiclass models could be because decoding many classes together helps to better constrain the relationship between features and class labels compared to doing 2 classes at a time.

B | Group-level decoding

B.1 Methods

B.1.1 Model analysis

In *Kernel FIR Analysis*, we investigate the frequency characteristics of the convolutional kernels. Random noise is fed into a trained model, and the power spectral density of the output of specific kernels is computed to assess their finite impulse response (FIR) properties.

$$\mathbf{X} \sim \mathcal{N}(0, 1) \tag{B.1}$$

$$y = f(\mathbf{X}; \theta)_{l,i,o} \tag{B.2}$$

where y is the output of the kernel in layer l, applied to input channel i, contributing to output channel o. f is the trained model with parameters θ. Then we compute the PSD of y to assess the kernel's spectral properties. For group models we add the learned subject embedding to $\mathbf{X}$ as usual. An issue with this method is that there are thousands of kernels in each layer, and for visualisation purposes we simply do a random sampling of these kernels. Alternatively, this method can be applied to whole feature channels as well, instead of individual kernels. Note that this method does not only assess the specific kernel's spectral properties as the input to a kernel within the model has been transformed by previous layers as well. However, we found this to produce more interesting visualisations than simply computing the PSD from the kernel weights directly. This is because of our architectural choice of having only kernels of size 2.

B.2 Results

B.2.1 Kernel analysis

Kernel FIR analysis shows the power spectra of kernels' outputs when input examples are Gaussian noise (Figure B.3). See Appendix B.1.1 for method description. This analysis is answering a different question about the kernels in WaveNet compared to spectral PFI, which asks what frequency content of the input are kernels most sensitive to. In contrast, Kernel FIR analysis asks what are the input-output filtering characteristics of kernels. This provides more insight into how successive layers in Wavenet build up more and more complex filters. The subject embedding was set to a subject with average accuracy. The power spectra were normalised to make visual comparisons between kernels easier. Since the WaveNet architecture uses dilated filters with only 2 values per filter, early layers show broad filtering characteristics, but already in layer 2, more emphasis is put on lower frequencies. In deeper layers, filters (kernels) become more tuned to specific frequencies, generally below 20Hz. This is in line with the spectral properties of MEG data as discussed above. Both the spectral PFI and kernel FIR analysis show that there is significant variability between the spectral information encoded by various kernels.

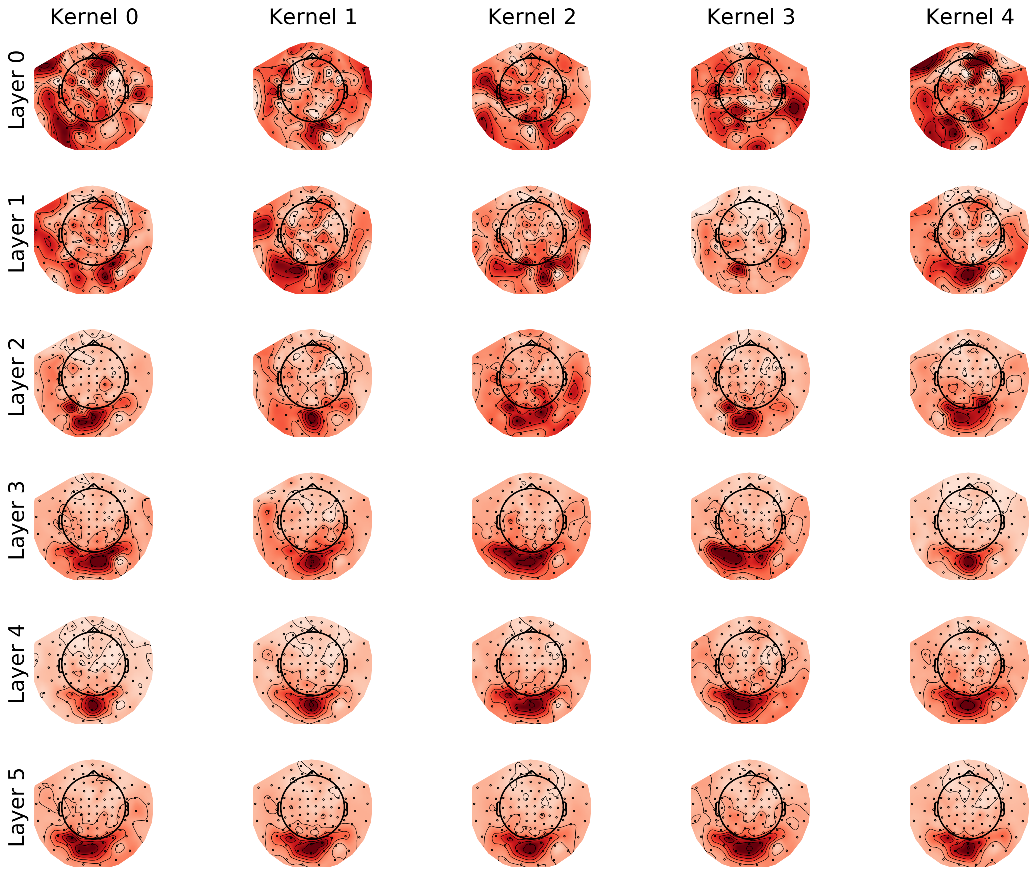
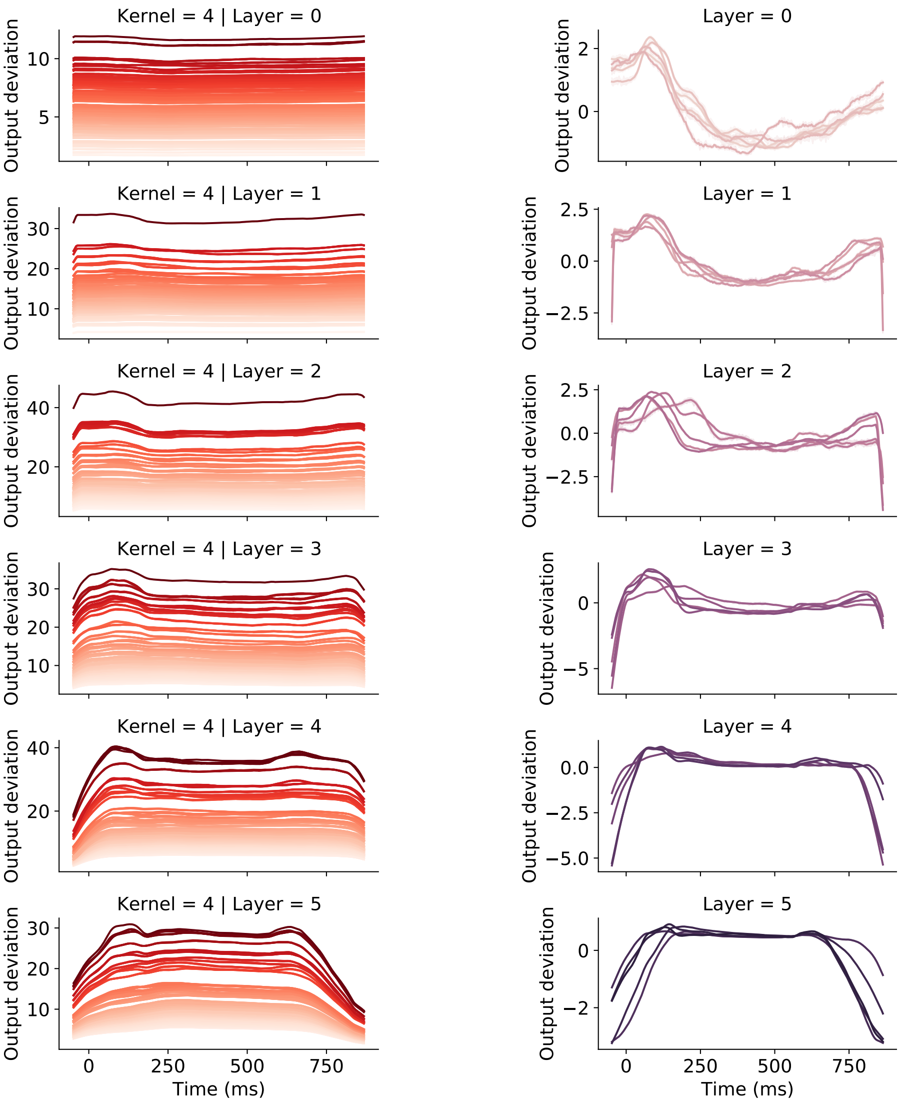


Figure B.1: Spatial PFI across 6 layers (rows) in the trained `non-linear group-emb` model, with 5 kernels per row. Darker reds mean higher output deviation.



(a) Channel-wise Temporal PFI

(b) Temporal PFI

Figure B.2: Channel-wise temporal PFI (a), and temporal PFI (b) across kernels of the `non-linear group-emb` model in 6 layers (rows). For temporal PFI 5 kernels (lines) are plotted together. Channel-wise temporal PFI shows the temporal PFI of each channel for Kernel 5. Channel colouring is matched to the corresponding spatial PFI map, and darker reds mean higher output deviation. For temporal PFI output deviation is normalised. The horizontal axis shows the time elapsed since the image presentation for both temporal PFI types. 95% confidence interval is shown with shading.

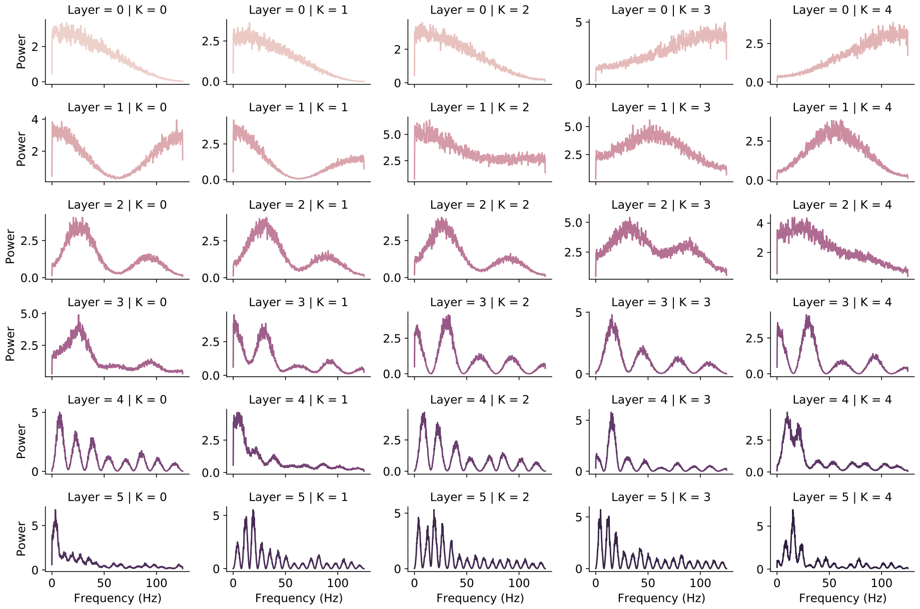


Figure B.3: Frequency characteristics of 5 kernels across 6 layers (rows) via kernel FIR analysis in the trained `non-linear group-emb` model. The power spectra are normalised.

C | Forecasting MEG signals

C.1 Methods

C.1.1 Simple Wavenet

As described in Section 2.3.4, a natural nonlinear extension of the linear multivariate autoregressive (AR) model is a convolutional neural network. Here we take inspiration from Wavenet (van den Oord et al., 2016) and design a simplified version for forecasting multichannel MEG data. This model is largely similar to the convolutional block of the Wavenet Classifier used in Chapter 4 (Figure 5.1). For `SimpleWavenet`, we only use the dilated convolutional layers applied directly to the raw continuous data, with the inverse hyperbolic sine as the activation function.

The network receives 512 timesteps as input, roughly 2 seconds at a sampling rate of 250 Hz. The model starts with a 1x1 convolution (kernel size of 1) which applies the same linear transformation at each time point, serving to increase the channel dimension (projecting from 306 channels to 612). This is followed by 9 dilated convolutional layers (Figure 5.1), where the number of channels is kept the same, but the dilation rate increases by a factor of 2 in each successive layer. Finally, the channel dimension is reduced back to the original size with another 1x1 convolution. As discussed in Chapter 4, using 9 layers provides a receptive field of exactly 512 timesteps, so the output is a single $\mathbb{R}^C$ representing the prediction vector of the continuous values of the data at the timestep:

$$\hat{\mathbf{x}}_{T+1} = \text{SimpleWavenet}(\mathbf{X}) \tag{C.1}$$

$$\mathcal{L}_{MSE} = \frac{1}{C}\sum_{i=1}^{C}(x_{t+1,i} - \hat{x}_{t+1,i})^2 \tag{C.2}$$

where $\mathbf{X} \in \mathbb{R}^{T \times C}$ is the MEG input segment of length T and C channels, $\hat{\mathbf{x}}_{T+1}$ is the predicted activity at time $T+1$, and $\mathbf{x}_{T+1}$ is the true brain activity. Compared to the Wavenet Classifier in Chapter 4, the main differences are the removal of the fully-connected block and the use of MSE loss $\mathcal{L}_{MSE}$ for forecasting instead of classification.

Normally we allow full mixing between channels since this is the default behaviour of standard convolutional layers, essentially implementing a fully-connected network across the channel dimension. We call this model type "multivariate." In some experiments we fit separate models to each channel, prohibiting cross-channel mixing. The ensemble of these channel-specific models is called a "univariate" model. This follows the nomenclature of univariate and multivariate AR models.

C.1.2 FlatGPT2

Directly vector quantising 300 channels to any vocabulary size would result in poor reconstruction. In `FlatGPT2` we perform the tokenisation on small groups of channels instead. First, we compute the covariance over channels in the training data. Then, we apply K-means clustering (Hartigan and Wong, 1979) on the covariance matrix to group channels into buckets. This ensures that each bucket contains channels with high covariance. This is important because tokenising a feature space (group of channels) with high covariance can be done with fewer tokens while maintaining low reconstruction error. We set the number of clusters

($B = 30$) based on manual tuning on the training data. Each cluster/bucket can contain a variable number of channels, usually between 5 and 20.

After assigning channels to buckets we apply the Residual Quantiser algorithm (Babenko and Lempitsky, 2014) from the faiss library[1] to each bucket b separately. This is a powerful additive quantiser (Liu et al., 2015) that achieves good reconstruction error with a relatively small vocabulary size V. Note that the total number of tokens, i.e. the vocabulary size will be BV, since we have B quantisers. Once fit to the training data the quantiser is fixed and can be applied to new data.

Mathematically, the covariance is obtained by:

$$\forall i, j \in 1, \ldots, C \quad \mathbf{C}_{ij} = \frac{1}{T} \sum_{t=1}^{T} (x_{t,i} - \mu_i)(x_{t,j} - \mu_j) \tag{C.3}$$

Where $x_{t,i}$ is the i^{th} channel at timestep t, μ_i is the mean of channel i over all timesteps, and C is the total number of channels. $\mathbf{C}$ is a symmetric matrix, and thus the feature and variable dimensions of K-means are the same. K-means computes buckets $\mathcal{C}_1, \ldots, \mathcal{C}_B$ which partition channels C into distinct sets with high within-bucket covariance.

The residual quantiser Q_b learns a codebook $\mathbf{C}_b \in \mathbb{R}^{V \times |\mathcal{C}_b|}$ for each bucket $\mathcal{C}_b$:

$$\forall t \in 1, \ldots, T \quad z_{t,b} = Q_b(\mathbf{x}_{t,b}; \mathbf{C}_b) \tag{C.4}$$

Where $z_{t,b}$ is the quantised representation (token/code) at timestep t of the channels $\mathbf{x}_{t,b} \in \mathbb{R}^{|\mathcal{C}_b|}$ in $\mathcal{C}_b$. The encoding in the quantiser is sequential, thus at stage m of the

[1] https://github.com/facebookresearch/faiss/wiki/Additive-quantizers

encoding of $\mathbf{x}_{t,b}$, the quantiser picks the entry i_m that best reconstructs the residual of $\mathbf{x}_{t,b}$ w.r.t. the previous encoding steps:

$$i_m = \underset{j}{\text{argmin}} ||\mathbf{T}_m(j) - (\mathbf{x}_{t,b} - \mathbf{T}_1[i_1] + ... + \mathbf{T}_{m-1}[i_{m-1}])||^2 \tag{C.5}$$

where $\mathbf{T}_m$ is a table of size K_m containing $|\mathcal{C}_b|$ dimensional vectors. For notational simplicity we omit the index b from i_m and $\mathbf{T}_m$ in the above. The quantisation provides a vector $[i_1, ..., i_M]$, where each element i_m comes from a set of size $\lceil \log_2(K_m) \rceil$ bits. This bit vector representation can be easily transformed to token indices ranging from 1 to $V = \sum_{m=1}^{M} \lceil \log_2(K_m) \rceil$. Note that this table description of the discrete code is just a different representation of the overall codebook $\mathbf{C}_b$. A code $[i_1, ..., i_M]$ can be reconstructed to obtain $\hat{\mathbf{x}}_{t,b}$ by retrieving the corresponding vectors from $(\mathbf{T}_1, \ldots, \mathbf{T}_m)$ and adding them up. Reconstruction error is computed by comparing $\hat{\mathbf{x}}_t$ and $\mathbf{x}_t$.

By using 30 buckets we obtain 30 tokens per timestep, which is already a 10-fold reduction of the original dimension space, but we have not reached our initial goal of having 1 token per timestep. To achieve this, we flatten the feature dimension (buckets) when feeding tokens to GPT2, hence the name `FlatGPT2`. Our total sequence length then becomes $B \cdot T$, where B is the number of buckets and T is the number of timesteps. This approach is also motivated by the observation that language models include extra information such as context within the sequence, instead of the feature space. Thus, when predicting the token of bucket b, we treat the previous timesteps of the other buckets as contextual information.

We also add an extra separator token z_{sep} between sequences of buckets corresponding to the same timestep to facilitate distinction between the bucket and time dimensions. An input sequence to `FlatGPT2` consists of tokens $z_{t,b}$ following a fixed order:

$$\mathbf{z} = \big(z_{sep}, z_{t=1,b=1}, z_{t=1,b=2}, \ldots, z_{t=1,b=B}, \tag{C.6}$$

$$z_{sep}, z_{t=2,b=1}, z_{t=2,b=2}, \ldots, z_{t=2,b=B}, \tag{C.7}$$

$$z_{sep}, \ldots, \ldots, z_{t=T,b=B}\big) \tag{C.8}$$

For each codebook $\mathbf{C}_b$ a separate embedding $\mathbf{W}_{e,b} \in \mathbb{R}^{V \times E}$ is learned. As in `ChannelGPT2` we add the appropriate conditioning embeddings to the input embedding with appropriate flattening across the channel/bucket dimension:

$$\mathbf{H}^{(0)} = \mathbf{Z}\mathbf{W}_e + \mathbf{W}_p + \mathbf{Y}\mathbf{W}_y + \mathbf{O}\mathbf{W}_o + \mathbf{W}_c + \mathbf{W}_t \tag{C.9}$$

where $+$ denotes element-wise addition and $\mathbf{Z} \in \mathbb{R}^{(B+1)T \times V}$ is the one-hot version of $\mathbf{z}$. The task labels $\mathbf{Y}$ can vary across time, but are the same across the buckets of one timepoint. $\mathbf{W}_c$ now contains distinct embeddings of buckets $b \in (1, \ldots, B)$, which are the same across timesteps. We also augment the input with $\mathbf{W}_t$, containing distinct embeddings for timesteps $t \in (1, \ldots, T)$, which are the same across buckets. This is the timestep version of $\mathbf{W}_c$.

As usual, the model is trained to autoregressively predict the next token in the sequence given all previous inputs. At timestep t and bucket b the model has access to the tokens $\mathbf{z}_{1:t-1}$ from all buckets (and thus information from all channels), and the tokens $\mathbf{z}_{t,1:b}$, and has to predict token $z_{t,b+1}$. The buckets of the same timestep are predicted sequentially, thus, bucket ordering could influence results. We use an arbitrary bucket ordering and do not experiment with different orderings of the input sequence.

Note that at the last bucket B in each timestep the prediction should be token z_{sep}, however, we simply discard this prediction during loss computation, as we do not require the model to predict separator tokens. The structure of the sequence already constrains the predictions such that a new timestep begins after every B tokens. Conversely, when computing the prediction at input token z_{sep}, the target is the token with bucket $b = 1$ of the next timestep. This is useful as in theory we could start the recursive generation of data with a single z_{sep} token.

At the output, the transpose of $\mathbf{W}_e$ can be used to predict probabilities over the vocabulary, or a separate linear projection can be learned. Note that because each codebook $\mathbf{C}_b$ has a separate vocabulary of size V assigned to it, we can speed up the output softmax by only computing probabilities over codes/tokens in $\mathbf{C}_b$ instead of the total vocabulary of size BV.

`FlatGPT2` contains important hyperparameters that affect design choices and performance (Table C.1). Increasing the number of buckets B improves reconstruction error, as the vector quantiser has to quantise less channels, but increases the length of the input sequence to `FlatGPT2`, and the total size of the vocabulary BV. The number of code tables M and the number of bits per code table define the size of the vocabulary $V = \sum_{m=1}^{M} K_m$. These were manually tuned, but generally, we observed that using fewer code tables with a higher number of bits achieves lower reconstruction error. For example, a vocabulary size of 16 bits can be achieved with both two 8-bit code tables and four 4-bit code tables. The trade-off is that using fewer code tables (with more bits) significantly increases computation time. Increasing the vocabulary V (through the number of code tables and bits per table) improves reconstruction error, as more codes are available for quantising a bucket of channels. However, this increases the total vocabulary BV of the model, resulting in a larger model.

Description	Parameter	Typical value
Number of buckets	B	30
Number of code tables	M	2
Number of bits per code table	$\lceil \log_2(K_m) \rceil$	7
Vocabulary size per bucket	$V = \sum_{m=1}^{M} K_m$	16384

Table C.1: Hyperparameters of the vector quantisation part of `FlatGPT2`.

In summary, key modifications compared to `ChannelGPT2` include vector quantisation (tokenisation) of channel groups, and flattening the channel dimension into the sequence. While in theory we could have flattened the full channel dimension without bucketing, this would have resulted in a 10x longer sequence length. However, we are limited by memory constraints since a standard GPT2 model scales quadratically with the sequence length. Memory-efficient Transformer variants are an active research area (Kitaev et al., 2020; Beltagy et al., 2020; Wang et al., 2020), but they have other drawbacks, and we leave their application to M/EEG data to future work.

C.1.3 Simulation

To understand the learning mechanisms of Wavenet and GPT2 models and compare them against the linear AR model, we generate simulated time series with controllable properties. Simulations allow us to precisely define key aspects of the data, such as frequency of underlying signals and signal-to-noise ratio (SNR). The simulated time series comprise a finite set of events that govern local dynamics (over hundreds of milliseconds) in the time course. Each event type manifests as a damped oscillation at a specific frequency. The sequence of events (i.e., which event follows the current event) is determined by a transition probability matrix containing conditional probabilities between all events. The lifetime of each event is randomly sampled from a Gamma distribution. For each event, a different

2nd-order autoregressive (AR(2)) model produces the damped oscillations using the following parameters:

$$\phi_1 = 2\cos\left(\frac{2\pi f}{S}\right) \tag{C.10}$$

$$\phi_2 = -1 \tag{C.11}$$

where f is the desired oscillation frequency and S is the sampling rate in Hz.

To simulate damping of the AR-generated oscillations over time t, we use:

$$x_t = z_t e^{-\lambda t} \tag{C.12}$$

where λ controls the damping rate. After generating the time series with damped oscillations and event transitions, we apply the inverse hyperbolic sine function to each time step independently to introduce non-linearity. Finally, independent Gaussian noise $\eta_t \sim \mathcal{N}(0, \sigma_\eta^2)$ is added to each time step.

In summary, a simulated time series $\mathbf{x}$ is constructed as follows. Let $s \in 1, 2, \ldots, K$ indicate an event, with K total event types. Event transitions follow a transition probability matrix P, where $P_{ij} = p(s_t = j|s_{t-1} = i)$ gives the probability of transitioning from event i to j.

The full generative model is:

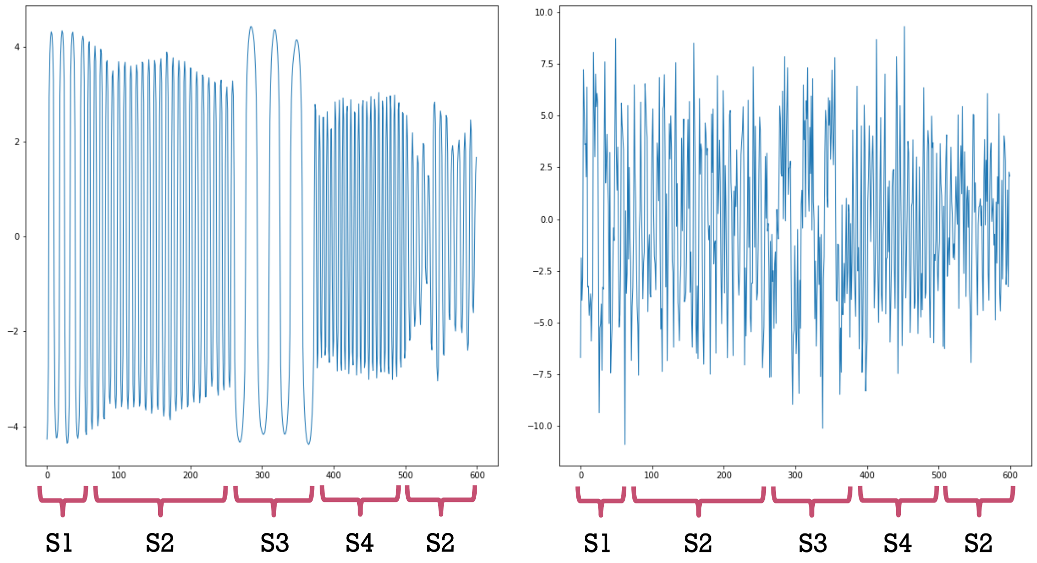


Figure C.1: Sample simulated timeseries with four events ($S_1 = 8$ Hz, $S_2 = 17$ Hz, $S_3 = 30$ Hz, $S_4 = 45$ Hz) at 250 Hz sampling rate. Each event has a different lifetime and AR process noise. The timeseries is shown before and after adding Gaussian noise on the left and right, respectively. The horizontal axis denotes timesteps.

$$s \sim \text{Categorical}(P_{s_{t-1}}) \quad if \quad t - 1 = T_{s_{prev}} \tag{C.13}$$

$$T_s \sim \text{Gamma}(\alpha, \beta) \tag{C.14}$$

$$z_t = \phi_{1,s} z_{t-1} + \phi_{2,s} z_{t-2} + \epsilon_{s,t} \tag{C.15}$$

$$x_t = \sinh^{-1}\left(z_t e^{-\lambda t}\right) + \eta_t \tag{C.16}$$

where

$$\epsilon_{s,t} \sim \mathcal{N}(0, \sigma_s^2) \tag{C.17}$$

$$\eta_t \sim \mathcal{N}(0, \sigma_\eta^2) \tag{C.18}$$

The event-specific AR process in Equation C.15 with parameters $\phi_{1,s}, \phi_{2,s}, \sigma_s^2$ generates values over timesteps $t = 1, \ldots, T_s$. A new event is only sampled at timestep T_s. The Gamma distribution shape and rate parameters are α and β, respectively. Figure C.1 shows a sample simulated timeseries before and after adding noise. We perform simulation experiments with univariate, single-channel data. Quantisation can be applied to simulated data similarly to real data.

C.2 Results

C.2.1 `SimpleWavenet` on simulated data

As a preliminary step, we evaluated our models on simulated data, where we could freely control the characteristics of the data and analyse how well the models could reproduce these features. In particular, we aimed to determine whether the continuous-data version of Wavenet (`SimpleWavenet`) could provide improved performance over a linear autoregressive (AR) model.

In order to obtain useful results with `SimpleWavenet`, it was crucial to set the signal-to-noise ratio (SNR) of the simulated data to approximately 1. This was observed empirically through extensive experimentation. While we do not have a good explanation for this, we believe that the point-estimate prediction of Wavenet make the model dynamics very sensitive, and it can easily diverge during generation. We generated single-channel simulations with 4, 8, and 12 distinct states, with state frequencies as follows:

1. 4 states: 10, 24, 36, 45 Hz
2. 8 states: 10, 14, 18, 22, 26, 33, 38, 45 Hz
3. 12 states: 8, 11, 14, 17, 20, 23, 26, 29, 35, 38, 41, 45 Hz

The state timecourse is generated by a probability transition matrix with state lifetimes sampled from a Gamma distribution, and each state plays out one of the oscillations above. For the rest of this section we use the terms events and states interchangeably. These frequency bands cover the physiologically relevant ranges typically observed in magnetoencephalography (MEG) data (Baillet, 2017). The total simulation duration was 3000 seconds at a 250 Hz sampling rate. The gamma distribution used for sampling event lifetimes had a shape and scale parameter of 10, yielding a probability density function with a peak around 90 ms, consistent with observed state lifetimes in empirical MEG data (Vidaurre et al., 2018c). State transition probabilities were drawn from a uniform distribution.

The noise for the AR(2) models was sampled from a uniform distribution between 0.8 and 1.0. The damping exponent was 0.005 and the variance of the added Gaussian noise was 2.5. Simulated data was split into training and validation sets with a 4:1 ratio. The training set was z-transformed to have zero mean and unit variance. The validation set was standardised using the same parameters. We trained an AR(64) model and a `SimpleWavenet` with 8 hidden channels on the training data and evaluated on the validation set. `SimpleWavenet` was trained until validation loss stabilised.

Figure C.2 compares the mean squared error (MSE) loss and variance of predictions for the AR and Wavenet models across multiple future timesteps, obtained by recursive generation without additional noise. `SimpleWavenet` achieved lower

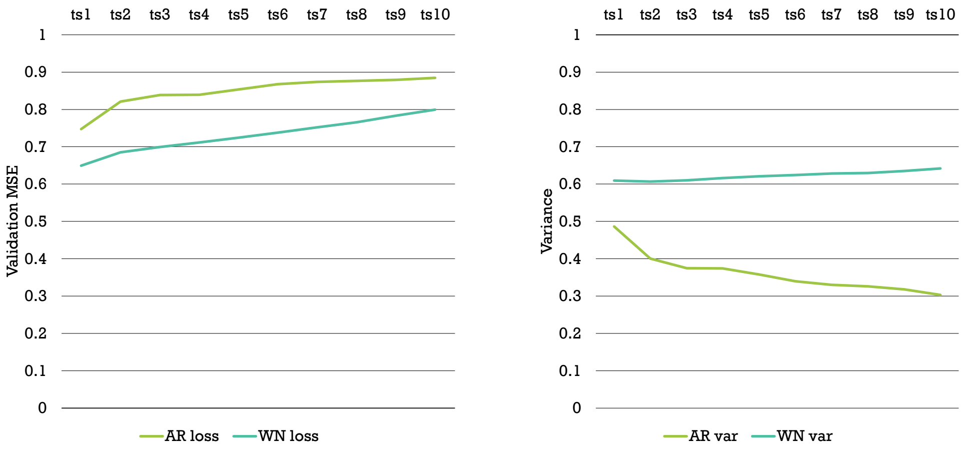


Figure C.2: MSE loss (left) and variance of predictions (right) for AR and Wavenet (WN) models. Performance is shown for recursive generation across future timesteps (horizontal axis, *ts*). Trainings were run on the simulated data with 8 states.

loss across all horizons, with variance close to the true data variance of 1. As expected, loss increased with longer prediction horizons. AR performance did not improve with higher model orders, suggesting `SimpleWavenet`'s superior performance stems from nonlinearity.

We generated 1000 seconds of data from the models trained on the 12-state simulations and computed the power spectral density using Welch's method (Welch, 1967). Figure C.3 shows the resulting power spectra. Both models accurately reproduced the frequency profile, with clear peaks at the true state frequencies.

To examine how frequencies evolve over time, we computed wavelet transforms (Mallat, 1999) for 36 seconds of generated data, from models trained on the simulation with 8 states. As shown in Figure C.4, qualitative differences emerged despite the similar power spectra. `SimpleWavenet` produced clear periods dominated by a single frequency, closely matching the true generative process. By contrast, the AR model blended frequencies. This demonstrates `SimpleWavenet`'s ability to capture greater signal complexity, likely due to nonlinearity.

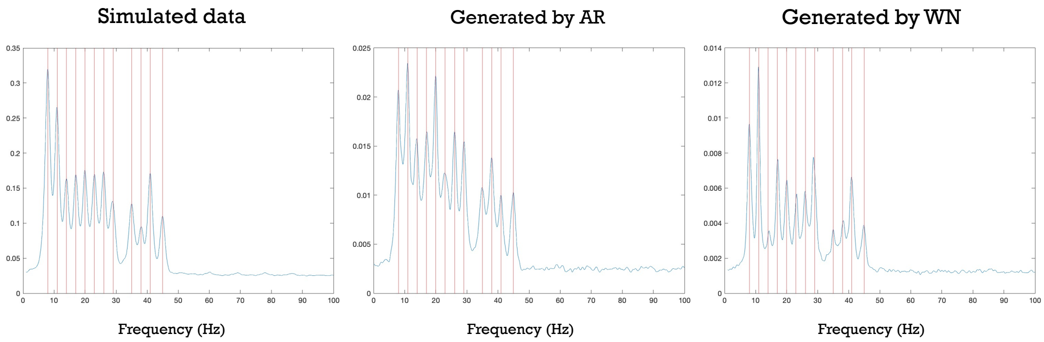


Figure C.3: Power spectra for simulated data (left), AR-generated data (middle), and Wavenet-generated data (WN, right). Vertical lines indicate ground-truth state frequencies.

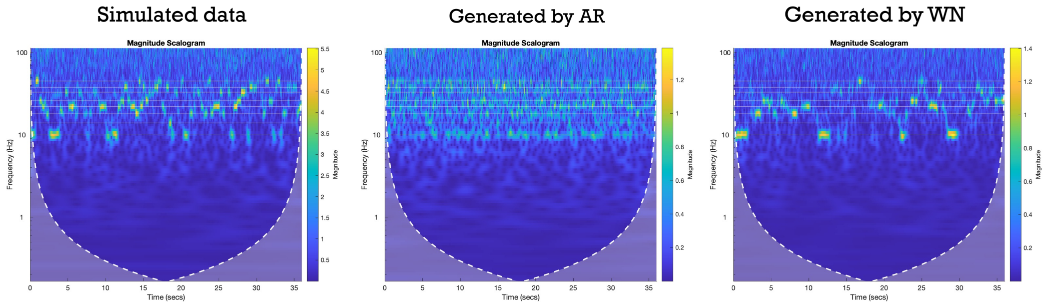


Figure C.4: Wavelet transforms for simulated data (left), AR-generated data (middle), and Wavenet-generated data (WN, right). White horizontal lines indicate ground-truth state frequencies.

To quantitatively evaluate how well `SimpleWavenet` reproduced the state-switching dynamics, we extracted state time courses from the generated data. The wavelet analysis clearly illustrated frequency switching, so we first extracted time courses for each of the 8 known frequencies. At each timestep, the frequency with maximum power was treated as the predicted state. For a more principled approach, we trained a HMM on these frequency time courses to infer states in an unsupervised manner (Vidaurre et al., 2018c). Figure C.5 shows the HMM-inferred state time course aligns closely with frequency switching in the wavelet transform. State time courses extracted by taking the most probable frequency (naive method)

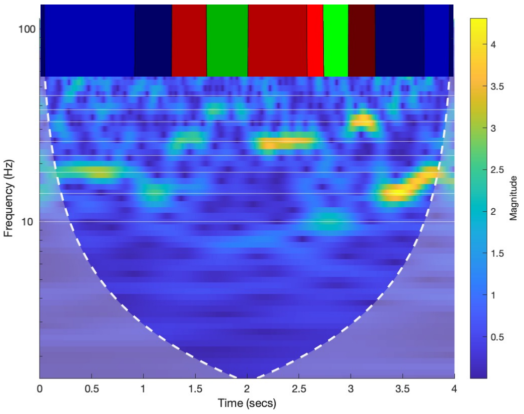


Figure C.5: Wavelet transform for simulated data with the HMM-inferred state time course superimposed. States coincide with distinct frequencies. Each state is a different colour.

at each timestep are compared in Figure C.6. Visually, `SimpleWavenet` largely reproduced the switching structure, while the AR model did not.

Finally, we analysed state lifetime distributions, which characterise state persistence. Taking the argmax of the HMM time course gave a state sequence from which lifetimes were calculated. Figure C.7 compares lifetime distributions for simulated and `SimpleWavenet` data. The noisy state extraction meant that distributions for simulated data differed from the true gamma distribution. However, the `SimpleWavenet` lifetimes closely matched the simulated data, with slightly more short states due to noisier state switching.

Additionally, we analysed the power spectra of kernels across layers to understand how the network processes frequencies. The power spectra predominantly exhibits the effects of dilation seen in Figure C.8 - sparser kernels have narrower periodic peaks. *Kernel FIR analysis*, however, reveals that deeper kernels became more selective for the 4 ground truth frequencies used in this simulation, likely reflecting the effect of dilations enabling longer temporal receptive fields (Figure C.9). Early layers can only apply wide filters, while deeper layers can be more selective.

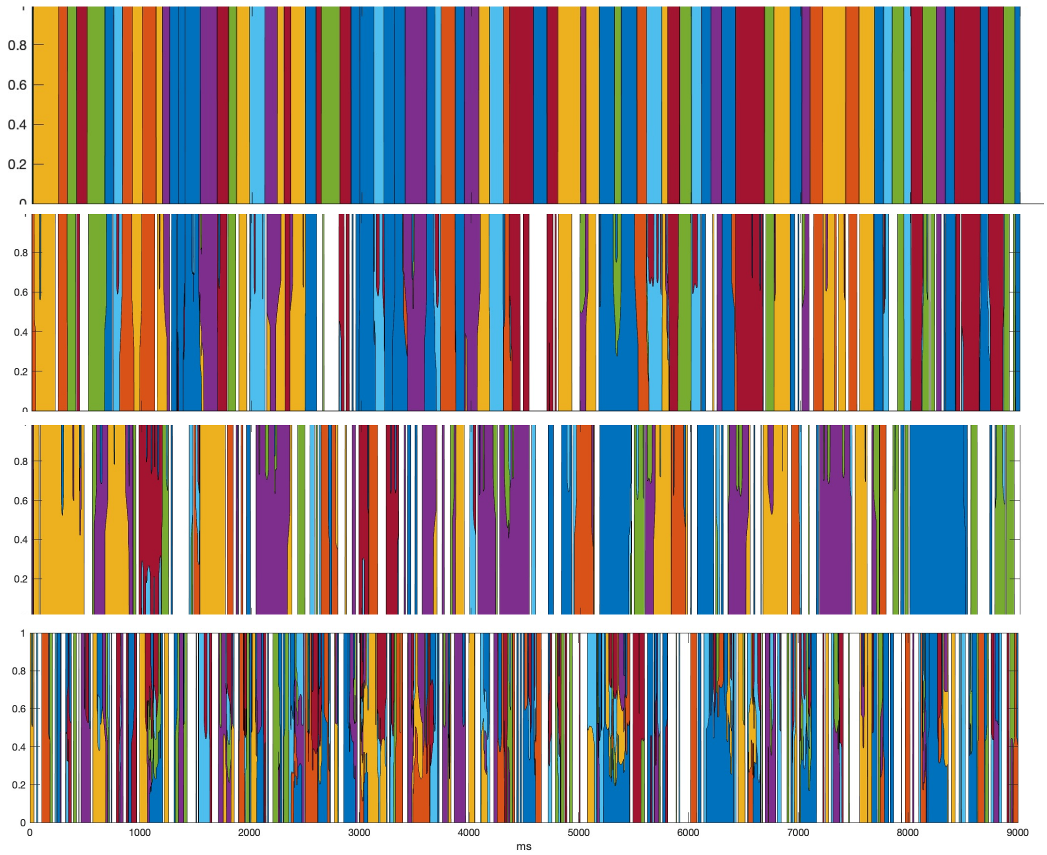


Figure C.6: Comparing state probability time courses extracted by the naive method for simulated data (2nd row), `SimpleWavenet` generated data (3rd row), and AR generated data (bottom). The top row shows the ground-truth state time course used to generate the simulated data. `SimpleWavenet` and AR time courses do not line up with the simulated data, since data is generated from random noise. The horizontal axis shows time in milliseconds (ms). The vertical axis shows the probability distribution of states represented by different colours.

The observed power spectra looks like a superposition of the dilation effect from Figure C.8 and the ground truth frequency peaks.

Together these results on controlled simulated data provide promising evidence that `SimpleWavenet` can capture structure in simulated electrophysiological signals better than linear models. The model demonstrated an ability to reproduce complex temporal dynamics, such as state switching.

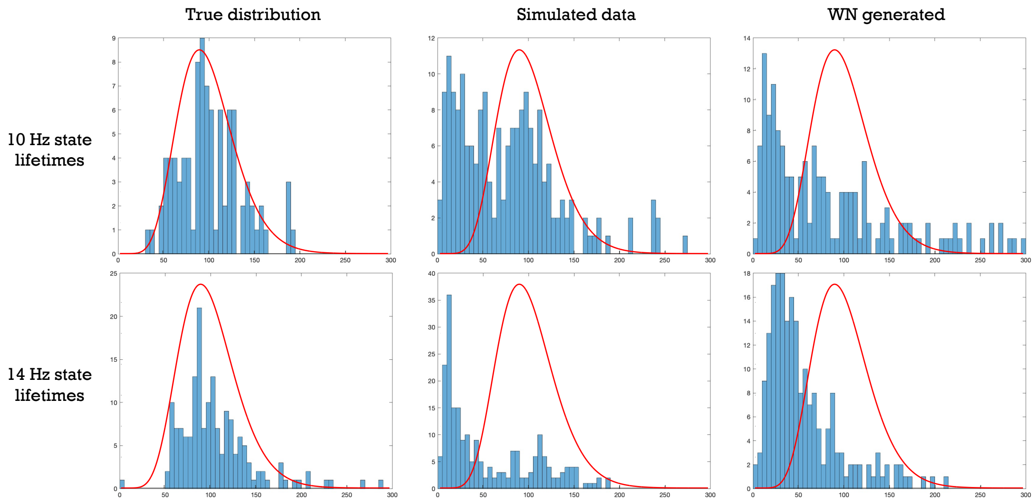


Figure C.7: Lifetime distributions (in milliseconds) for the 10 and 14 Hz states. The first column is the true distribution originally sampled to generate the simulated data. The second and third columns are the state lifetime distributions based on the HMM state time courses inferred from the simulated and `SimpleWavenet` (WN) generated time series, respectively. The red curve shows the true gamma probability density function from which the state lifetimes were sampled for the simulated data.

C.2.2 Quantised simulated data

From the previous section, we can conclude that a simplified version of Wavenet applied to continuous simulated data performs well. As our baseline model is a linear autoregressive (AR) model, this showed that using a multi-layer nonlinear architecture can better capture the dynamics of the data such as switching between events characterised by different oscillations. Next, we wanted to test whether the full Wavenet model is also able to produce this switching behaviour on the simulated data when it is quantised. This is primarily aimed at validating the quantised Wavenet approach, and whether training through cross-entropy loss and generating data through sampling from the output probability distribution works as well as the Wavenet trained on continuous data. Validating the GPT2 approach on simulated data and drawing comparisons with Wavenet is left for future work.

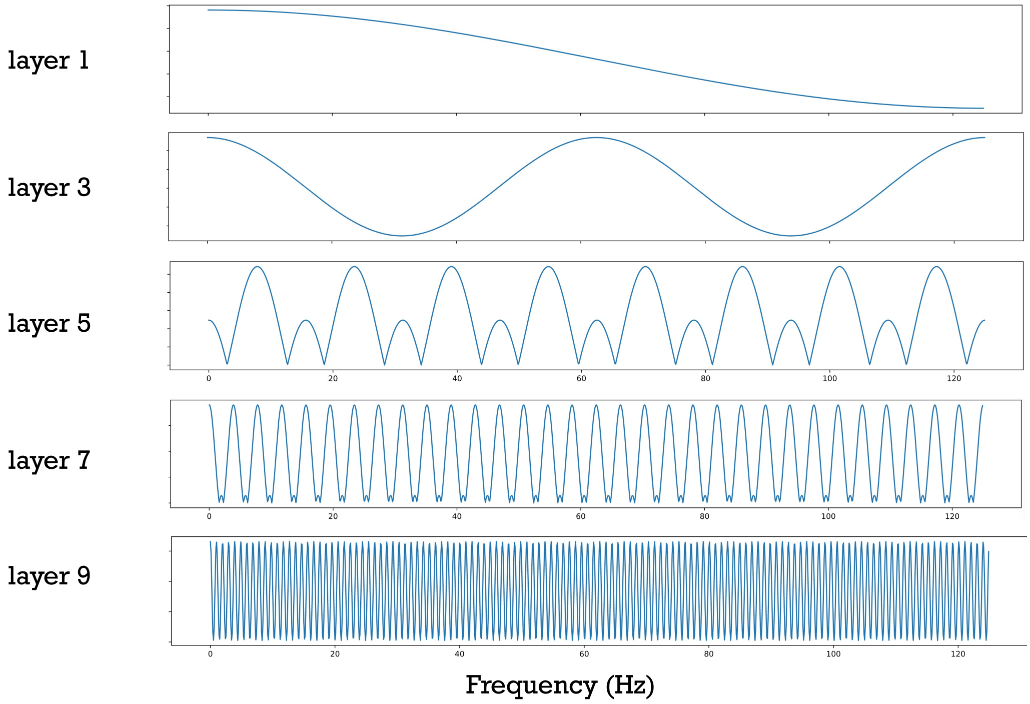


Figure C.8: The power spectra of 5 random kernels (as FIR filters) across layers of `SimpleWavenet`.

Specifically, we generated 1000 seconds of simulated data with 8 events, using the frequencies described in the previous section. The data was generated at 1000 Hz, then a 1-100 Hz infinite impulse response (IIR) bandpass filter was applied, and then downsampled to 200 Hz. This approach was used to match preprocessing steps of real data. Finally, the data was quantised using the mu-law companding transform to 256 bins. Data was split into train and validation sets with a 4:1 ratio.

We set the model order (receptive field) of the linear AR model to 255. As our Wavenet model we used `WavenetFullChannel` with a matched receptive field. We used two identical dilation blocks stacked on top of each other. A single block contained 7 layers with doubling of the dilation factors in successive layers. Dropout rate between layers was set to 0.2. The embedding for the quantised inputs was set to size 64, and the hidden channel size of the convolutions was 128. The

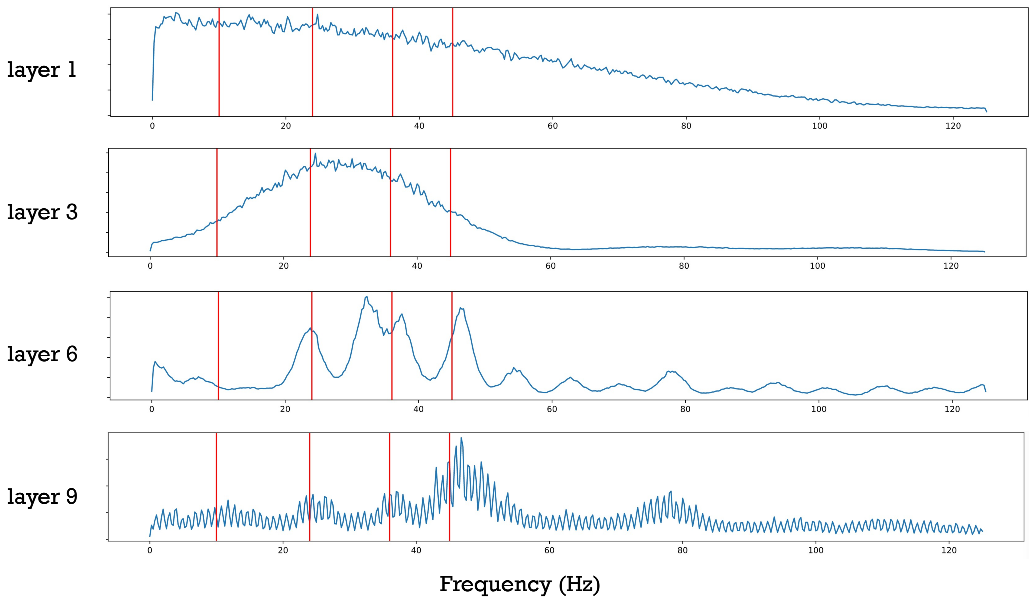


Figure C.9: The power spectra of 5 random kernels from kernel FIR analysis of `SimpleWavenet`. Vertical lines indicate the 4 ground truth frequencies.

channel dimension of the skip convolutions was set to 512. The linear AR model is applied directly to the quantised values and produces continuous outputs.

The accuracy and mean squared error (MSE) of predictions on the validation set is presented in Table C.2. Accuracy is computed over the 256 bins. For the linear AR model, the closest bin of the prediction is used to compute accuracy. For `WavenetFullChannel`, MSE is obtained by reconstructing the original signal from the quantised output, by applying the inverse of the mu-law transform, and comparing to the target values. The results clearly show that `WavenetFullChannel` is somewhat better at predicting future timesteps compared to the AR(255) model. We also tried an AR(10) model which showed the same performance as AR(255). This demonstrates that the linear AR model is not able to leverage longer receptive fields effectively. Interestingly, the next-timestep prediction accuracy of the AR model is worse than the repeat baseline, however the MSE is much lower. Because the AR model was optimised with MSE, it makes

	Accuracy	MSE
`WavenetFullChannel`	4.3%	0.038
AR(255)	2.1%	0.049
Repeat baseline	2.3%	0.092

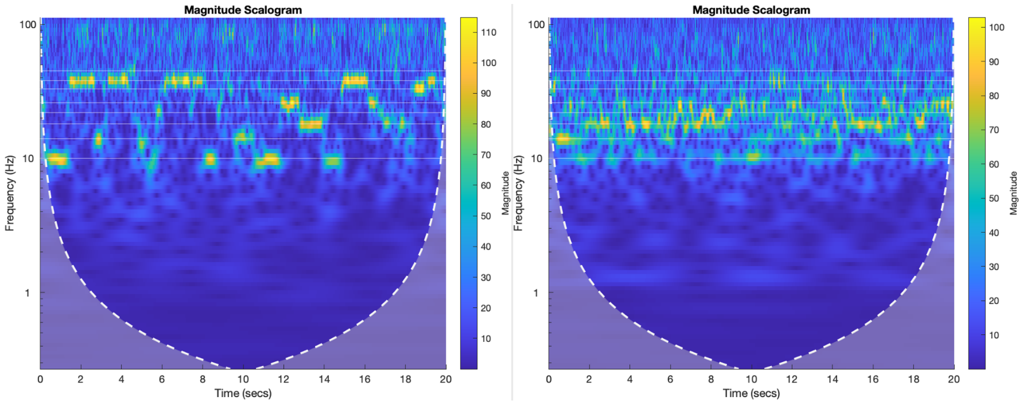
Table C.2: Comparing the accuracy and MSE of a linear AR model with order 255 and `WavenetFullChannel`. The last row shows the performance achieved by a baseline model which always repeats the last timestep. Chance level is $100/256\%$.

sense that accuracy might not reflect its performance as well.

Next, we tested whether `WavenetFullChannel` can generate the event-switching dynamics of the simulated data. We sampled from the full output distribution during generation. As shown in Figure C.10a, it has similarly capabilities to `WavenetSimple`. Thus, the simulation now becomes too simplistic for these models, and we move to applying them to real data in the next section.

We also performed an ablation of the nonlinearity of `WavenetFullChannel`. In the ablated model, we replaced all activation functions with the identity function $y = x$. This ensures that the model is essentially linear, other than the softmax function at the output, and the nonlinear training dynamics caused by dropout and successive dilated convolutions. A comparison of generated data between the standard `WavenetFullChannel` and the ablated (linear) version is shown in Figure C.10. This clearly shows that the linear version is much worse at capturing the event-switching structure of the simulated data, and its generation spectrum looks closer to the linear AR model.

In summary, we have demonstrated that the full Wavenet model with quantised inputs can successfully model the switching dynamics in simulated time series data, outperforming linear AR models. The nonlinear nature of Wavenet was shown to be critical through an ablation study.



(a) `WavenetFullChannel` (b) Linear `WavenetFullChannel`

Figure C.10: Wavelet transform of the generated data from `WavenetFullChannel` (left) and the ablated (linear) version (right). White horizontal lines show the true frequencies used to create the simulated data.

C.2.3 Next time-step prediction performance

For `FlatGPT2` we set the (temporal) receptive field to be between 120 and 240 because of memory constraints. Note that the total (actual) receptive field of the model is the temporal receptive field multiplied by the number of buckets + 1. All embedding sizes were set to 96, and we used 8 GPT2 layers, with 8 attention heads. Dropout was set to 0 and we used early stopping on the validation set. The quantisation parameters are given in Table C.1.

Next-timestep forecasting accuracy for different models on a sample subject is shown in Table C.3. Beyond standard accuracy (the number of true positives divided by the number of all examples), we also evaluated top-5 accuracy, counting a prediction as correct if the true bin was within the 5 most probable bins. Surprisingly, all models performed only slightly better than a naive baseline of repeating the previous timestep's value. This suggests next-timestep metrics do not effectively capture model performance.

As expected, the linear AR model had lower MSE but worse accuracy than the nonlinear models. This can be because MSE measures the distance of the prediction to the target, while accuracy is only 1 if the prediction is in the target bin. Thus it can be that the AR model always predicts values that are slightly closer to the target, but never quite falling in the target bin. While `WavenetFullChannel` appears to be worse, `WavenetFullChannelMix` and `ChannelGPT2` have nearly identical performance.

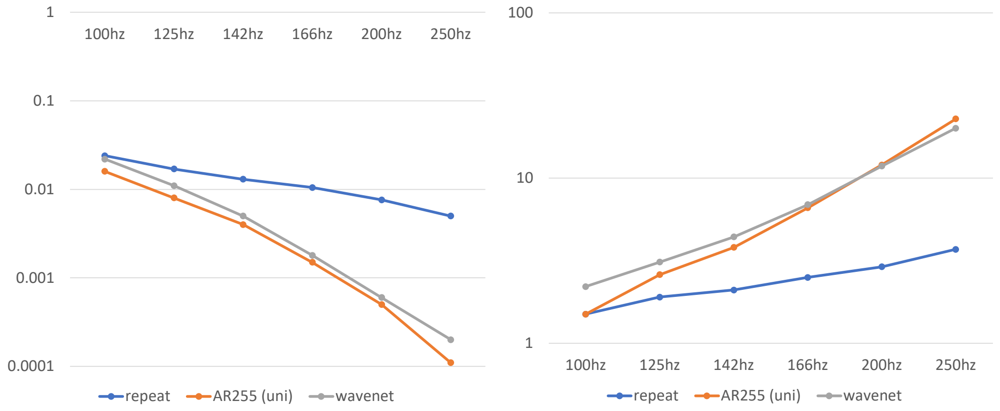
All these observations are very likely a consequence of these metrics not capturing actual goodness of modelling the data. Specifically MSE and accuracy measure only how well models predict the next timestep. As we have seen in Chapter 5, these models have very different dynamics when generating data over longer temporal horizons. Perhaps looking at these metrics when recursively generating multiple timesteps in the future might be more informative.

`FlatGPT2` performance is not comparable to other models, since the output distribution is over 16384 tokens, and the prediction is done sequentially for channels as well. This latter point is probably the reason why we observe such high accuracies, since it is much easier to predict the same timestep of one channel at a time, while some others (possibly with high correlation) have already been predicted. In addition it is possible to have a skewed distribution of tokens and thus true chance level may be higher than $1/16384$.

To allow for true next-timestep prediction, where the model can not access information from other channels in the same timestep, we also computed a recursive loss. This is done by recursively generating all of the channels/buckets within a timestep t. Thus the model has to rely on its own predictions and can not "cheat" by using the true value of buckets $1, ..., b$, when predicting bucket $b + 1$ within timestep t. Importantly this metric is computed after the model is trained, so there is a discrepancy in how the model was trained (autoregressively over buckets) and

Model	MSE	Top-1 Accuracy	Top-5 Accuracy
Repeat baseline	0.024	1.5	7.6
AR(255)	0.016	1.5	7.5
WavenetFullChannel	0.026	2.0	9.8
WavenetFullChannelMix	0.022	2.2	10.8
ChannelGPT2	0.023	2.2	10.9
FlatGPT2	-	3.0	10.8
FlatGPT2 recursive	-	0.0015	0.0069

Table C.3: Test data next-timestep prediction performance across various models. Accuracy values are given in percentages. Note that FlatGPT2 is not comparable to other models as the prediction is done for buckets with much larger vocabularies. Chance-level for FlatGPT2 is $1/16384$, while for other models it is $1/256$. FlatGPT2 *recursive* refers to recursive prediction of all buckets within the same timestep.



(a) Test MSE

(b) Test accuracy (%)

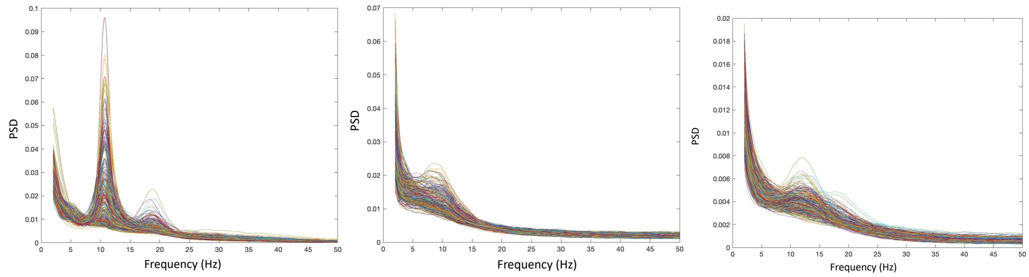
Figure C.11: Comparing AR(255) and WavenetFullChannelMix (wavenet) across increasing sampling rates of the data. *repeat* refers to the repeat baseline. Accuracy is given in percentages.

tested. As we can see in Table C.3 the accuracy of recursive predictions is much lower than in the normal operating mode. This kind of analysis ties into losses that optimise for multi-timestep future horizons. We believe this may be important to consider as future research for improving FlatGPT2 and ChannelGPT2.

We further analysed sampling rate effects on forecasting performance in Figure C.11. We trained the AR(255) and `WavenetFullChannelMix` models on increasing sampling rates of the data from 100 Hz to 250 Hz. The lowpass filter was kept the same at 50 Hz. The receptive fields were kept the same in terms of timesteps, thus they decreased accordingly in terms of actual time in seconds. As expected, both AR and Wavenet models improved markedly with higher sampling rates, as the prediction task became easier when timesteps were closer together. The performance gap between models and the repeating baseline also grew with sampling rate. However, these trends are likely influenced by both the changing prediction interval and filtering artefacts. It is unlikely that such marked improvement would be caused by better modelling of higher-frequency content. Varying the low-pass cutoff with sampling rate reduced performance, suggesting filtering effects dominate. Removal of noise with lower lowpass filters is also a possible explanation. Overall, next-timestep prediction remains a problematic metric. Not only does it not differentiate between models of very different characteristics and dynamics it is also heavily dependent on arbitrary factors like sampling rate.

C.2.4 FlatGPT2 on group data

Unfortunately, even scaling `FlatGPT2` did not improve evoked generation. However, we did find that the spectral content of the generated data matched the real data much better than the single-subject version of `FlatGPT2` (Figure C.12). `FlatGPT2-group` seemed to scale particularly well with model size as larger models achieved lower and lower loss, improving test accuracy by multiple folds (16.1% top-1 and 40.1% top-5 accuracy) over single-subject `FlatGPT2` (3% top-1 and 10.8% top-5 accuracy). This is interesting behaviour compared to `ChannelGPT2-group` which did not improve much on our forecasting metrics. It remains to be seen whether even more data and larger models are needed to make



(a) Data (b) `FlatGPT2` (c) `FlatGPT2-group`

Figure C.12: Comparison of generated data PSD across data single-subject `FlatGPT2` and `FlatGPT2-group`. Each line represents a different MEG channel.

this type of architecture viable.

C.2.5 Ablations

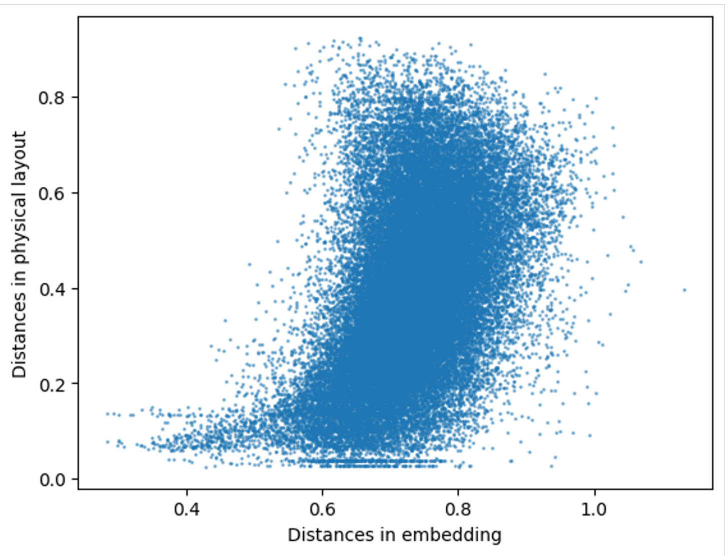


Figure C.13: Plotting pairwise Euclidean distances of channels in real, physical space versus embedding space. Sensors that are near to each other in the real sensor montage tend to have more similar embeddings. Each point represents a different pair of channels. Correlation is 0.45.

D | Decoding thoughts

D.1 Results

D.1.1 Evoked analysis

We computed evoked responses jointly across the two inner speech types without any baseline correction. We visualise these for each session for electrode PO7 (visual area) in Figure D.1. It is evident that evoked responses across sessions are very similar in the visual area, except for session 2 which appears to be an outlier. The plot also shows the expected response to visual stimulus, with the first peak as early as 100 ms post-stimulus (P100), followed by several peaks and troughs. This demonstrates the oscillatory nature of the evoked response in the visual area, likely due to the cross cue used in the inner speech task.

While we were also interested to compare other channels across sessions, non-visual channels exhibit more noise. Thus, we plot evoked responses in separate plots per-session. Figure D.2 shows this for the T7 electrode, which is above the temporal lobe. Evoked responses in the temporal lobe are much more variable across sessions, but all display peak activity around 400 ms post-stimulus. However, it is questionable whether this reflects language-related activity (due to inner speech), or merely spreading/propagation of the visual response. The latter is more probable.

It is important to note that since we utilise the Cz electrode for referencing, our evoked results are influenced by this. Any evoked response present at the reference is subtracted from all other channels. To better elucidate the spatiotemporal evolution of the evoked response, we plot responses averaged across sessions for all channels concurrently (Figure D.3). This demonstrates that after the initial two

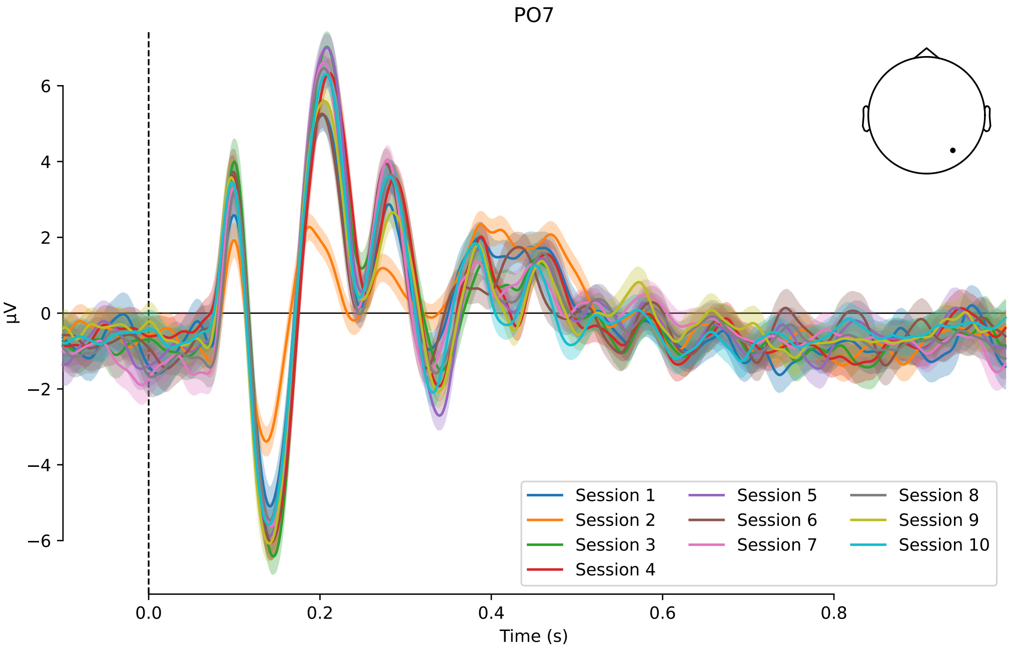


Figure D.1: Evoked responses across the 10 EEG sessions of P4 for 1 electrode (PO7) in the visual area. Shading indicates 95% confidence interval across trials. Timepoint 0 indicates stimulus (cross) onset.

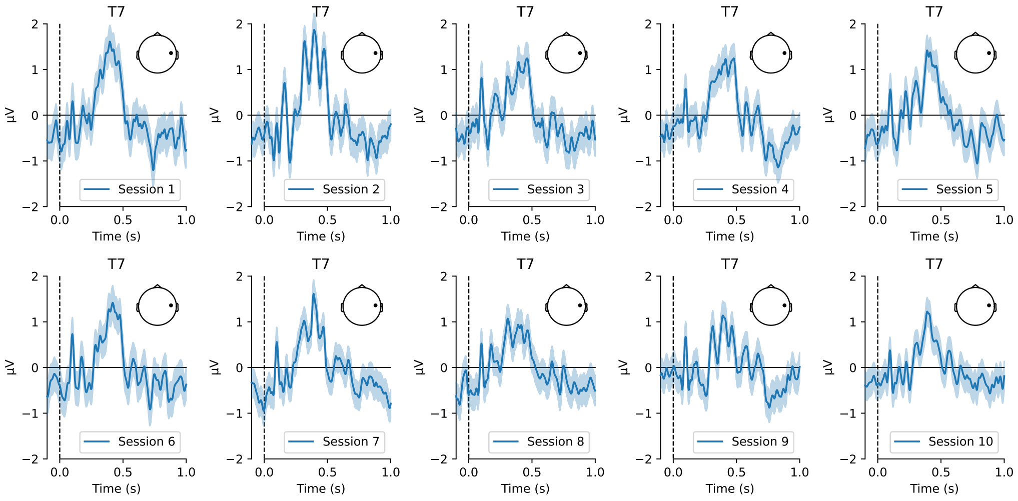


Figure D.2: Evoked responses across the 10 EEG sessions of P4 for 1 electrode (T7) above the temporal lobe. Shading indicates 95% confidence interval across trials. Timepoint 0 indicates stimulus (cross) onset.

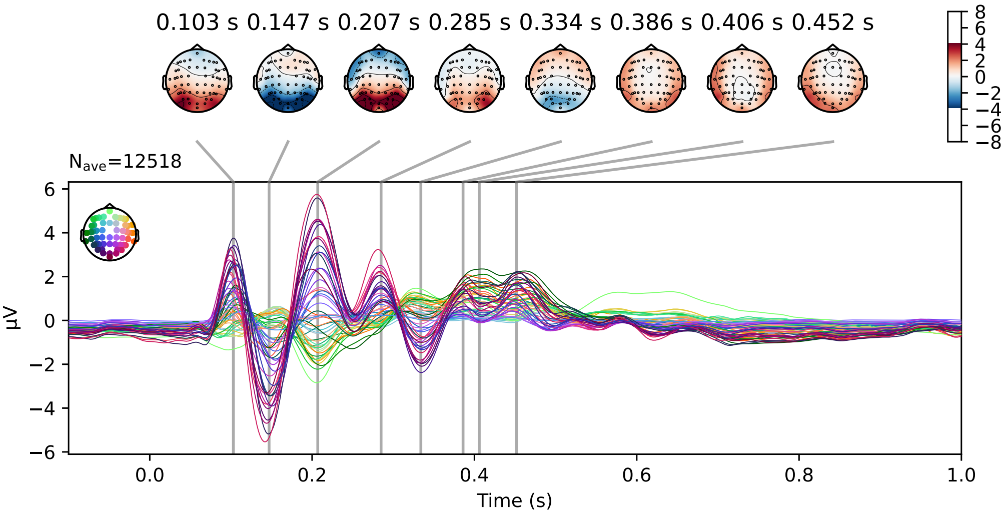
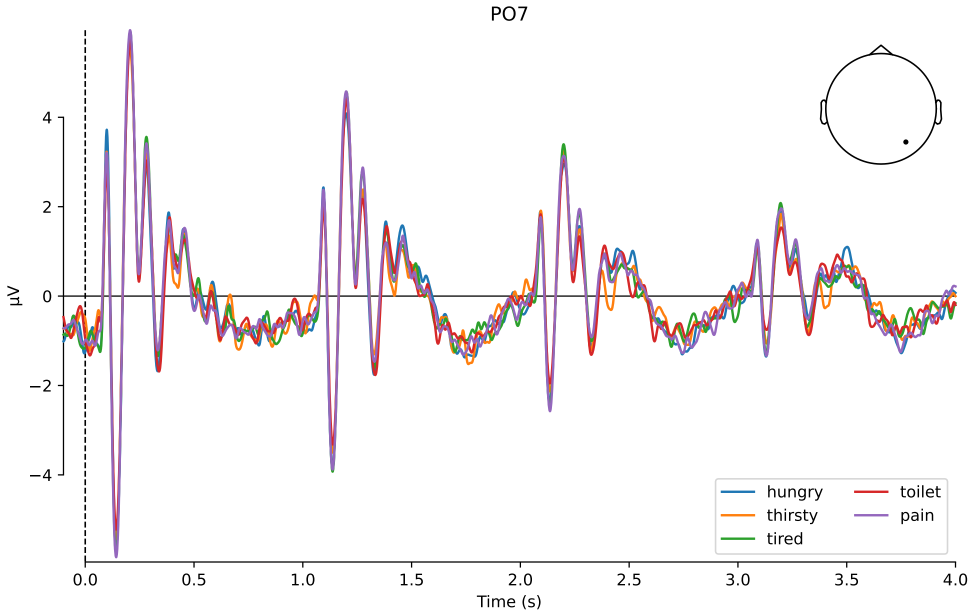


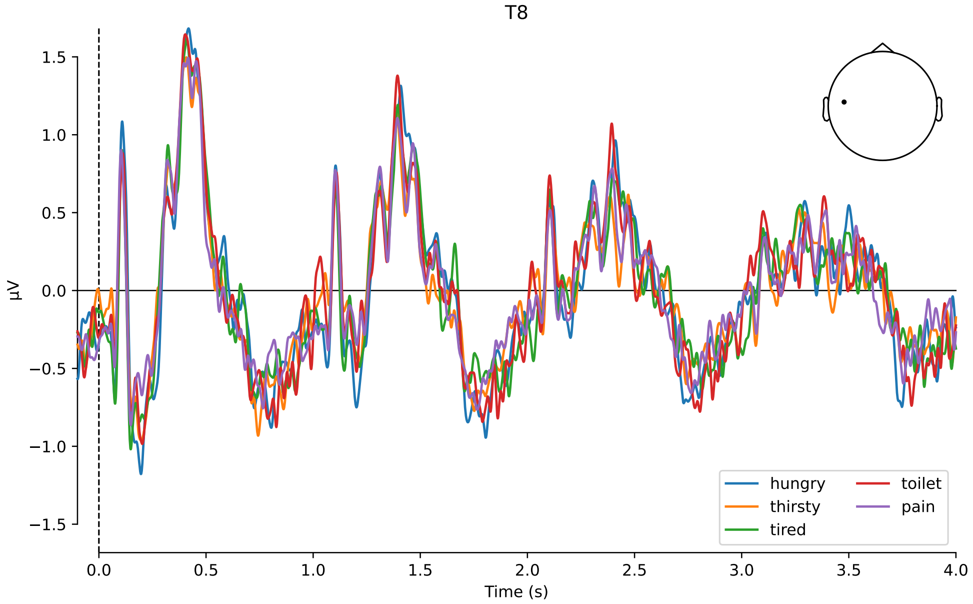
Figure D.3: Joint evoked responses for each channel averaged across all 10 EEG sessions of P4. The spatial topography and timestamp of notable peaks is shown in the upper part.

visual peaks, at 200 ms a third positive visual peak emerges, accompanied by a smaller negative activity in the frontal area. Then, at 285 ms another smaller visual peak occurs, followed by a negative peak at 334 ms. At this time, some positive activity also arises in the frontal area. Finally, more visual positivity is observed around 386 ms, which shifts slightly to temporal/lateral areas at 406 ms, returning to the visual area at 452 ms. This plot provides a robust characterisation of the evoked response to inner speech across a substantial number of trials and sessions. However, we suspect the described activity remains principally due to the cross-cue presentation.

Figure D.4 displays the evoked response for each word across all sessions. Again, we only examine the two inner speech types here. We also opted to plot the entire 4-second trial with the 4 consecutive crosses, rather than averaging over these. We can discern that as the trial continues, the response to subsequent cross cues diminishes. This could reflect genuine activity changes and/or more noise across



(a) PO7



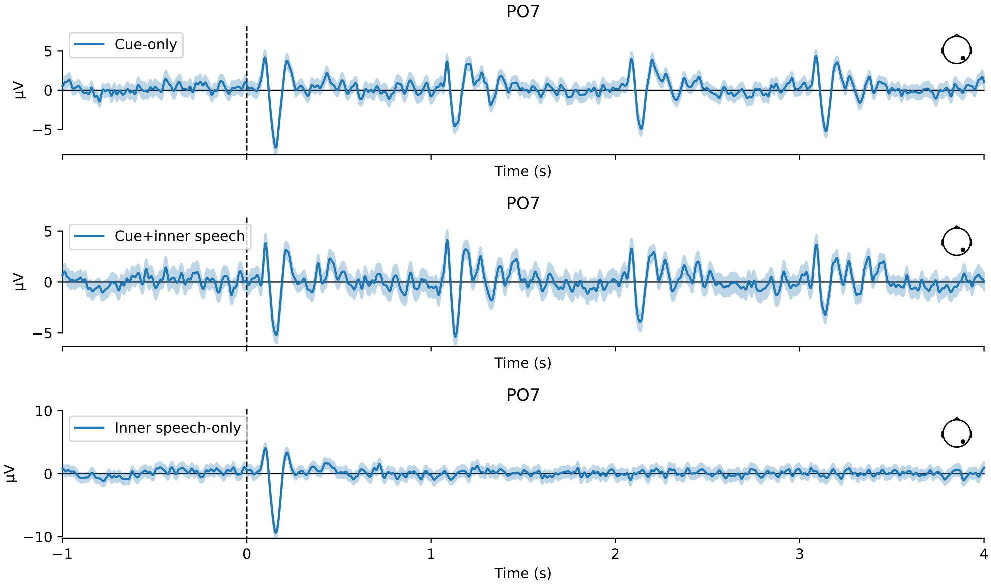
(b) T8

Figure D.4: 4-second evoked responses in 2 channels (PO7 and T8) across the 5 words averaged across all 10 EEG sessions of P4. Each line represents a different word.

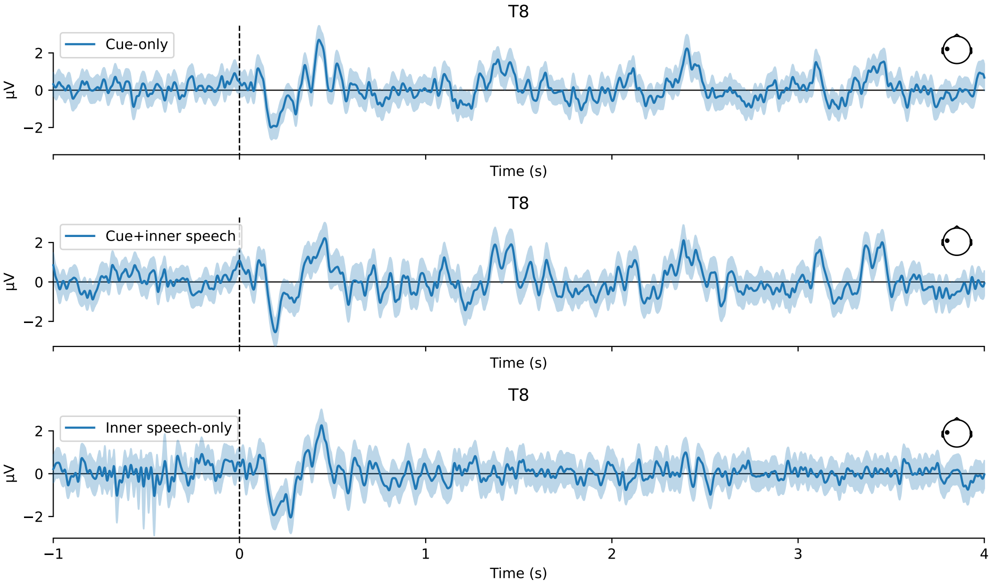
sessions later in the 4-second trial. There are no apparent differences between the evoked responses of words. This is anticipated since inner speech should elicit very subtle distinctions in EEG that would be nullified when averaging over many trials and sessions.

We wanted to verify that the evoked responses are purely visual in nature. However, there is no straightforward way to separate the visual and inner speech-related activity. The cross cue is essential to provide consistent timing for inner speech production; otherwise, variability in timing would impede decoding. Still, we conducted 1 EEG session with P4, where we implemented 3 tasks. First, we utilised the standard repetitive inner speech task with the 4 consecutive cues (*cue+inner speech*). Then, we included a task where the visual stimuli were identical, but the participant was instructed not to think/internally vocalise the words, simply observe the crosses (*cue-only* task). Finally, we incorporated a task with only 1 cross cue at the beginning of the 4-second trial, after which the participant attempted to repeat the inner speech 4 times as in the original task, but without timing alignment (*inner speech-only* task).

Evoked responses across the 4-second trials for the three tasks from this single session are depicted in Figure D.5. This provides unambiguous evidence that the previously observed evoked responses are elicited by the cross cue. There are no discernible differences between the brain activity of solely observing the cues, compared to also engaging inner speech. The task where inner speech had to be repeated without visual cues shows that after the evoked response to the initial cue, there are no subsequent evoked responses attributable to inner speech. This could also stem from variability in timing. However, it is more likely that utilising inner speech is too subtle to generate brain signals exceeding baseline noise. These findings imply that decoding inner speech may be an equally challenging endeavour.



(a) PO7



(b) T8

Figure D.5: 4-second evoked responses in 2 channels, PO7 and T8, for the EEG session with 3 tasks. The evoked response across the 4-second trial is shown for the cue-only (top), cue+inner speech (middle), and inner speech-only (bottom) tasks, in both (a) and (b). Shading indicates 95% confidence interval across trials.