2023

Computational Modeling of Temporal EEG Responses to Cyclic Binary Visual Stimulus Patterns

Connor M. Delaney
Virginia Commonwealth University

COMPUTATIONAL MODELING OF TEMPORAL EEG RESPONSES TO CYCLIC BINARY VISUAL STIMULUS PATTERNS

A submitted in partial fulfillment of the requirements for the degree of Master of Science at Virginia Commonwealth University.

by

CONNOR DELANEY

Bachelor's of Science with Virginia Commonwealth University - August 2018 to May 2022

Director: Dean Krusienski, PhD.,

Professor and Graduate Program Director, Department of Biomedical Engineering

Virginia Commonwewalth University

Richmond, Virginia

December, 2023

Acknowledgements

I would like to thank my advisor Dr. Dean Krusienski for his guidance throughout my development as a researcher. His willingness to share his experience and knowledge with me has meant the world to me. I would also like to thank the other members of my committee, Dr. Paul Wetzel and Dr. Cheng Ly, for their valuable time and constructive comments in regards to this thesis, as well as their support throughout my other academic ambitions. I would also like to acknowledge my lab group, Michael Dexheimer, Sam Cole, and Josephine Wallner, Tina Naik, and Pedram Zanganeh Soroush for their continued support throughout my research endeavors.

Finally, I am most grateful for all of my friends and family. I would not be the person I am today without their constant support and love through the best and worst of times. This thesis was completed during a very turbulent time in my life and without them, this project could not have been completed.

This work was supported, in part, by the National Science Foundation (NSF) Award Number 1944389.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract

COMPUTATIONAL MODELING OF TEMPORAL EEG RESPONSES TO CYCLIC BINARY VISUAL STIMULUS PATTERNS

By Connor Delaney

A submitted in partial fulfillment of the requirements for the degree of Master of Science at Virginia Commonwealth University.

Virginia Commonwealth University, 2023.

Director: Dean Krusienski, PhD.,
Professor and Graduate Program Director, Department of Biomedical Engineering

The human visual system serves as the basis for many modern computer vision and machine learning approaches. While detailed biophysical models of certain aspects of the visual system exist, little work has been done to develop an end-to-end model from the visual stimulus to the signals generated at the visual cortex measured via the scalp electroencephalogram (EEG). The creation of such a model would not only provide a better understanding of the visual processing pathways but would also facilitate the design and evaluation of more robust visual stimuli for brain-computer interfaces (BCIs). A novel experiment was designed and conducted where 15 participants viewed stereotyped visual stimuli while their EEG was recorded simultaneously. The resulting EEG responses were characterized across participants. Furthermore, a Residual Connection Feed Forward system identification Neural Network (ReCon FFNN) was implemented as a preliminary end-to-end model of the visual system that uses the temporal characteristics of the visual stimulus as the model input and the corresponding EEG time series as the model output. This preliminary model was

able to reproduce temporal and spectral characteristics of the EEG and serves as a proof of concept for the development of future artificial neural network or biophysical models that incorporate spatio-temporal information.

CHAPTER 1

INTRODUCTION

1.1 Introduction

Humans heavily rely on their senses to function in everyday life. One of the major goals of neuroscience is to understand how the information from sensory pathways is manipulated by the brain leading to conscious or unconscious interpretation of these senses by the central nervous system. One of the most critical and dominant human senses is vision. Visual perception of our environment allows us to plan and execute tasks much more efficiently and underpins many common interactions we have with our environment. Accordingly, the visual system has received much attention from the field of neuroscience. Retinotopic mapping and studies aimed at better understanding pathways involved in visual processing have been able to improve our understanding of the layout of the visual system [1, 2, 3]. Physiological experiments have been performed to reveal the biological mechanisms underpinning the visual system, and computational models have been developed to give deeper insight into how these underlying mechanisms react to changes [4]. Brain-computer interfaces (BCIs) have been developed to interpret neural signals generated by the visual system to offer practical utility by introducing new ways for people to interact with computers [5]. However, the visual system is far from perfectly understood. There remains a need to continue to investigate the visual system and develop a model which is not only capable of rendering new insights into the behavior of the visual system but also able to improve the performance of BCIs. Accurate biophysical or computational models of the visual system can be utilized to improve stimulus design and detection

compared to conventional, time-consuming empirical approaches, while also offering new insights into physiological mechanisms and extending existing data sets.

The proposed approach for probing the mechanisms of the visual system is to (1) collect data from a novel physiological experiment, (2) characterize the resulting physiological responses, (3) develop a computational model capable of reproducing realistic physiological responses, and (4) validate the model by demonstrating its outputs have the same characteristics as the recorded physiological data.

The physiological experiment underpins the other parts of the approach to a large extent. This study aims to unveil the biological basis for how sensory information is transmitted and processed. This is much easier said than done, however, as effectively measuring human neural processes directly *in vivo* presents many associated risks. Indirect measures such as electroencephalography (EEG), functional near-infrared imaging (fNIR), magnetoencephalography (MEG), and functional magnetic resonance imaging (fMRI) have all been used as well, but each technique has its limitations (e.g. temporal and spatial resolution, portability, safety, cost, etc.) [4, 6, 7, 2]. Furthermore, understanding how visual information is manipulated within the visual cortex is nearly impossible by indirectly recording physiological data alone. For this reason, biophysical and computational models have been developed to attempt to explain the behavior observed by these physiological experiments at various scales.

Some models operate at the level of individual neurons while others operate at the level of neuronal populations or regions consisting of larger divisions of the brain [8, 9]. Typically, models with finer resolution (i.e. neuronal level) are desired over those with larger resolutions (neuron population level), as those with finer resolution are ideally able to more closely resemble the organization of cells within the visual system. However, models with larger resolutions are often able to capture the overall behavior of the visual system using simpler, more efficient methods. The ideal goal of

these models would be to extrapolate our understanding of the visual system by applying them to new inputs. However, often due to limitations in recording equipment, computational hardware, and/or physiological understanding, models are developed for specific regions within the brain rather than modeling from visual input to completely processed output. This problem, and the large list of assumptions linked to many models, leads to models that have limited practical utility. On the other hand, BCIs are used to study the brain in a way that typically focuses on a practical application [5]. These systems are designed to interpret and classify the physiological activity observed within the brain and use that information to achieve a practical goal. BCIs based on visual processing, referred to as visual BCIs, are also useful for improving our understanding of visual processing as the components of neural behavior related to stimulus classification are likely to contain pertinent information. In this way, visual BCIs are useful tools for finding where and how important visual information is transmitted in the brain. However, BCI research often neglects to relate classification features to physiological mechanisms. Therefore, a logical approach is to merge aspects of modeling and visual BCIs to offer new insights into how the brain processes visual information.

1.2 Motivation

The visual system is critical for performing many everyday tasks. Understanding the visual processing performed by the brain is a key area of neuroscience, with potential impacts in a wide array of applications. Modeling physiological systems is a classic approach to garnering deeper insight into the mechanisms of how the body regulates itself and processes external stimuli. Accordingly, this thesis proposes a novel visual experiment aimed at developing an end-to-end model visual system, from visual stimulus to EEG response. A novel data set from 15 participants was collected,

characterized, and made publicly available for others to utilize for developing such models. Using this data, a preliminary model of the visual system from visual input to the visual cortex is implemented as a proof-of-concept. This preliminary model a launching point for future work, and aims to inspire other researchers to pursue developing original end-to-end models of the visual system or iterate on the one presented in this thesis.

While there have been many attempts to computationally model the visual system using a variety of methods, the majority of these models lack practical outputs and have severe limitations in terms of scope [9, 10, 11, 12, 8, 13]. On the other hand, visual BCI research is primarily focused on practical applications, and therefore most often leaves something to be desired in regards to connecting the features used for classification to the visual processing being performed [5]. Furthermore, the recent advances in computational resources and the development of powerful neural networks as models for real nervous system circuits have given way to the possibility of new modeling techniques originally inspired by the structure of the visual cortex.

Artificial neural networks (ANNs) are a type of computational system inspired by the biology of the visual cortex, suggesting that they may be a strong modeling tool for the visual system [9]. This preliminary modeling attempt aims to demonstrate that developing a model incorporating the strengths of neural networks and analysis techniques used for visual BCIs can produce a more robust and practical model of the visual system. The preliminary model implemented is a Residual Connection Feed Forward System Identification Neural Network (ReCon FFNN), which models EEG data based on the corresponding input sequence of visual stimulation.

1.3 Objective and Approach

There are two primary objectives of this work. The first is to collect a novel data set, which can be utilized to model the visual system. A study was designed and performed on 15 participants. The participants viewed a set of cyclic binary visual stimulus patterns while their EEG was simultaneously recorded. This data set has been cleaned, annotated, characterized, and made publicly available.

The second objective inspires a new line of inquiry into modeling the visual system using the visual stimulus pattern as the model input. To this end, a preliminary model was developed representing the processing of the visual system from visual input to the visual cortex. Models were developed using a ReCon FFNN for each participant's EEG data separately. The models developed in this work were evaluated in terms of their ability to produce realistic synthetic EEG data based on the stimulus input sequence which shared characteristics with the recorded EEG. This thesis aims to establish a proof-of-concept for further development of future models with the same goal.

It should be noted that the ReCon FFNN model of the visual system belongs to a family of models that are not intended to contain components that directly model the biophysics of the visual system. The decision to utilize a model that is agnostic of the biophysics of the visual system was motivated by the desire to blend practicality with a reliable computational model bearing biomimetic features (e.g. residual connections, layer connectivity inspired by simple and complex cells, etc.).

To make the model more tractable, it is limited to modeling the temporal characteristics of EEG. Thus, the visual stimuli were intentionally designed to be highly spatially correlated. To remove spatial information from the multiple EEG channels used in recording participant responses, a canonical correlation analysis (CCA) spa-

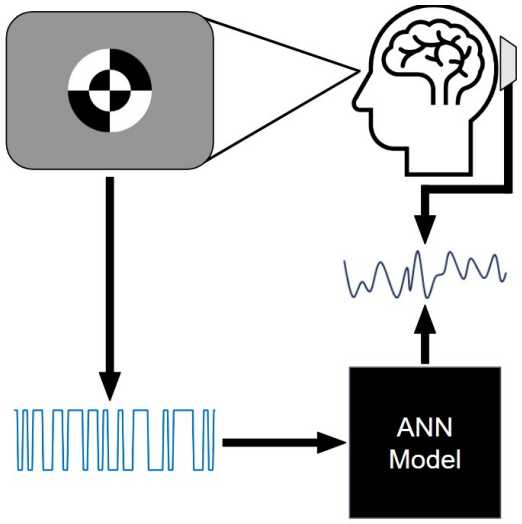


Fig. 1.: The proposed approach for the preliminary model the visual system from visual input to visual cortex. The visual stimulus is highly spatially correlated, thus only the temporal information from the stimulus was used to train the model. The ANN architecture used in this study is a novel structure called a ReCon FFNN.

tial filtering procedure was employed. After the data were collected and preprocessed, the ReCon FFNN model was trained using the sequences of flashes as the model input and the associated EEG response as the target output. Analysis of the results of the modeling process was then performed to determine the strength of the model. Additionally, there is currently no gold standard technique for evaluating the performance of models generating EEG. Habashi *et al.* (2023) state this in a clear, concise manner: "To date, there is no direct metric that could be considered a real assessment of the generated EEG signals' quality and directly relate it to the performance of the model" [14]. For this reason, it remains a challenge to definitively validate such models. This work represents an initial attempt at modeling and defining potential evaluation metrics in this context. This work hopes to inspire future research into this under-explored end-to-end modeling of the visual system.

1.4 Summary of Chapters

The subsequent chapters of this thesis are organized as follows. Chapter 2 presents the background of the human visual system. Chapter 3 presents the background of visual BCIs as it relates to this work. Chapter 4 presents the state-of-the-art of neural networks, how they function, and how they are used with time series data such as the physiological recordings used in this study. Chapter 5 describes the experimental design and the tools utilized to execute the experiment. Chapter 6 details the preparation of the data recorded from participants including using CCA as a spatial filtering technique. Chapter 7 discusses the desired properties of an ANN model of the visual system, and presents the architecture of the ReCon FFNN developed to model the visual system. An evaluation of the performance of the model is presented in Chapter 8. Chapter 9 concludes the dissertation with a discussion of the main contributions of this work, limitations, and possible future directions for continued research in this area.

CHAPTER 2

THE VISUAL SYSTEM AND COMPUTATIONAL MODELS OF THE VISUAL SYSTEM

This chapter presents background on the human visual system. The chapter is organized in a way that follows the flow of sensory information from the eye through the visual cortex. While not an exhaustive presentation of all known qualities of the visual system, this chapter aims to give the reader sufficient understanding to see parallels between the anatomical and physiological function of the visual system and the preliminary computational model presented in this study (e.g. feed-forward and feedback connections within the visual system).

2.1 The Eye and Optic Nerve

The visual system is made up of a combination of light-sensitive photoreceptors and an immense number of neurons responsible for processing visual information within the brain. All visual information the body receives begins with the eyes, which have a structure designed for efficient collection of sensory information from light in the environment. The sclera is the connective tissue on the outermost layer of the eye. The next layer is the choroid which is a highly pigmented layer of epithelial tissue, which helps draw light onto the inside surface of the eye. The retina is the innermost layer and contains the photoreceptors and neurons responsible for transmitting information from photoreceptors to the optic nerve [15]. This structure is slightly altered at the front of the eye where light enters. The cornea is an oblong protrusion in the sclera which is transparent to allow light to enter the eye. The choroid in this

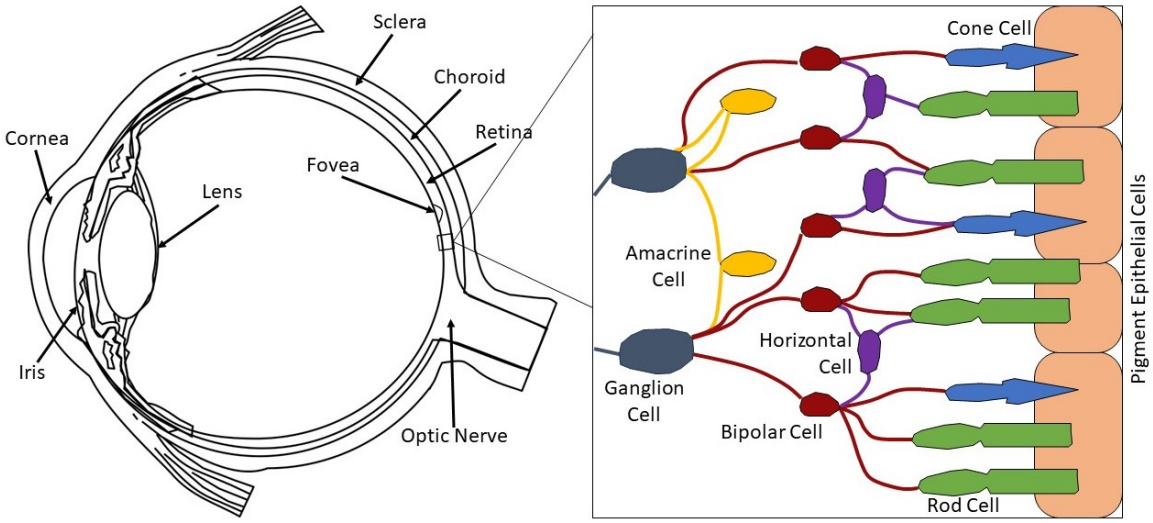


Fig. 2.: Diagram of structure of the eye (left) and diagram of the organization of photoreceptors and sensory neurons (right). Connections from Ganglion Cells synapse with the optic nerve. The anatomical illustration of the eye is public domain.

same area becomes the iris, a region with colored pigment similarly responsible for helping draw light into the eye. On the inside of the eye, positioned behind the iris, is the lens. This transparent disk of proteins can be stretched to specific thicknesses allowing for visual depth adjustments. A diagram of the eye's structure is shown in Figure 2.

The retina itself is where visual processing begins. The photoreceptors can be divided into two classes: rods and cones. Rods detect the presence or absence of light without color specificity, while cones are responsible for color detection. Furthermore, the distribution of rods and cones is organized throughout the retina. The fovea is the portion of the retina linked to the center of the visual field and has the densest number of photoreceptors. Light activates photoreceptors within the retina causing the release of chemical signals to bipolar cells, initiating an electrochemical cascade of intercellular communication. Bipolar cells synapse with one or more photoreceptors and interspersed among them are horizontal cells. These horizontal cells can activate

or inhibit bipolar cells in a process called lateral inhibition. Bipolar cells transmit signals to ganglion cells. Mirroring the design of bipolar cells and horizontal cells, ganglion cells are assisted by amacrine cells that also perform lateral inhibition. These layers of integration ultimately generate the receptive fields of the visual system. Each ganglion cell transmits information through the optic nerve, the axons of which terminate within the brain. Figure 2 includes an illustration of the organization of these cells.

2.2 Visual Processing within the Brain

The first structure in the brain that receives and manipulates the signal from the optic nerve is the lateral geniculate nucleus (LGN) within the thalamus. This junction is responsible for diverting the signals from the optic nerve to specific regions of the visual cortex [16]. The visual cortex takes up a large portion of the occipital lobe and is divided into six regions: V1 through V6 [17]. Each of these regions exists in both hemispheres for processing information from the contralateral half of the field of vision, has its specific role in visual processing, and is further divided into layers. The region that receives the vast majority of visual information from the LGN is V1, which is also known as the primary visual cortex or striate cortex. Layer 4 of V1 receives the information directly from the LGN and then distributes information throughout the rest of the layers [17]. One of the most notable features of the primary visual cortex is the presence and structure of simple and complex cells. Simple cells respond to the orientation of lines and edges in a given portion of the field of vision. Complex cells receive and sum inputs from multiple simple cells to process information from a larger area within the field of vision. Furthermore, these complex cells have a scaled response relative to motion in a specific direction. This implies that simple cells process a lower order component of visual information, lines or edges, and the

downstream complex cells process a higher order component of visual information, namely the motion of lines or edges. This notion of visual processing beginning with simple features of the visual information and gradually increasing the complexity of the features being identified can be applied to the areas of the visual cortex as a whole [18]. Furthermore, it has also been suggested that the visual cortex operates on a "middle-out" basis, where lower-order features making up a certain size of the visual field are identified before the extraction of smaller and larger features [19]. This theory dictates that simple cells are tuned to only extract features of a certain resolution within their portion of the visual field. Complex cells extract features spanning more than one simple cell's portion of the visual field. V1 is responsible for extracting the first features from the visual input, and the organization of simple and complex cells within it is a direct inspiration for the design of modern neural networks (see section 4.1).

The next region of the visual cortex, V2, receives inputs from V1, and further increases the complexity of features being identified. It is responsible for differentiating between colors, spatial frequency of lines, patterns, and object orientation. V2 has feed-forward outputs to V3, V4, and V5 as well as feedback outputs to V1. The feedback outputs from V2 are part of a larger number of feedback outputs from higher-order areas of visual processing to V1. These connections seem to play a role in adaptation over time which occurs in V1 [17]. This feedback causes what is called contextual receptive fields, which is the ability to weigh the contributions from parts of the V1 network to its ultimate output. Practically what this leads to is the ability for the visual system to self-regulate the information it deems important within the visual field. These types of feedback connections are a direct inspiration for residual connections in ANNs (see section 7.1.3). Visual information continues to flow through to the higher-order regions of the visual cortex: V3, V4, V5, and V6. These higher-

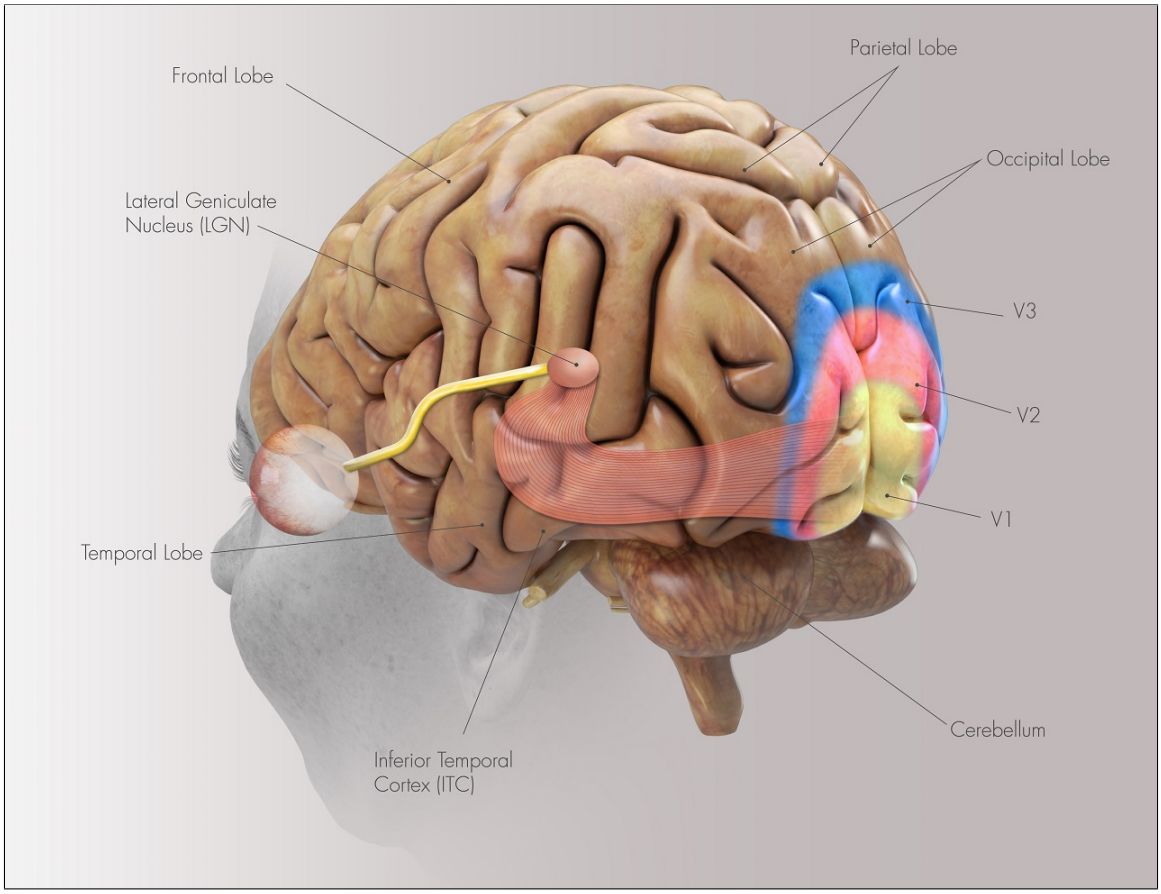


Fig. 3.: Diagram of the Visual System [21]. Visual information is sensed in the eye and transmits information to the LGN, which in turn transmits information to the visual cortex. The visual cortex contains feed forward and feedback connections throughout its regions and layers.

order regions are responsible for object identification and linking motion to body movements [20]. After the visual cortex, visual information is diffused throughout many different areas of the brain where visual information is identified and utilized to execute tasks, ultimately resulting in what we refer to as vision.

Overall, the visual cortex contains highly specialized regions that process lower-order features before higher-order features of visual information. This structure has inspired biophysical models, but due to the complexity of the system and many unknown qualities of visual processing, many if not all, models opt for simplified

structures or a reduction in scope by only including a subset of the visual pathway. Ultimately this restricts the practical implementation of these models, which has given way to the approach presented in this thesis of modeling the visual system in an end-to-end fashion.

CHAPTER 3

VISUAL BRAIN-COMPUTER INTERFACES

This chapter discusses the background of visual BCIs that primarily utilize the electrophysiological information from the visual cortex. The chapter begins with an overview of visual evoked potentials (VEP) which are the basis for nearly all visual BCIs. Specific approaches to visual BCIs, including Steady-State VEPs (SSVEPs) and coded VEPs (cVEPs), are then discussed. Additionally, a focus is given to certain relevant techniques for visual BCIs including canonical correlation analysis and the generation of synthetic EEG as visual BCI data. It should be noted that this section focuses almost exclusively on EEG-based visual BCIs and, while this includes a large amount of research into visual BCIs, it is not intended to be an exhaustive review of the state-of-the-art.

3.1 Visual Evoked Potentials

The basis of modern visual BCIs is an electrophysiological phenomenon called visual evoked potentials (VEP), which consists of a summation of the activity of a set of neurons within the visual cortex in response to a visual stimulus. Traditionally, VEPs are recorded using scalp EEG and are considered to be solely a sensory response (i.e., separate from consciousness and attention) [22]. The characteristics of VEPs are directly and strongly related to the characteristics of the stimulus being perceived and, accordingly, their numerous spatio-temporal stimulus parameters must be considered, as minor changes can result in drastic differences in the recorded VEP. Flashing or flickering patterns are perhaps the most widely used types of stimuli in VEP

applications and experiments. Reversal-Pattern VEP is the use of light and dark elements within a stimulus which are contrast reversed to create the flashing pattern, commonly referred to as a flicker pattern. This technique allows for constant variation in the stimulus while maintaining the average luminance [23], produces relatively consistent results in participants, and is the technique used for the present study.

The spatial, temporal, chromatic, and contrast properties of the stimulus can be modified throughout the delivery of the stimulus to evoke different responses within the visual cortex. Traditionally, the amplitude of the VEP is considered to increase linearly with contrast and disappears with light and dark contrasts within 10% of each other [23]. The visual stimuli used in VEP experiments and clinical applications are typically referred to as checkerboard visual stimuli, which consist of a spatial arrangement of squares with binary luminance with a distinct contrast difference (e.g., black and white). For clarification, the term retinotopic stimuli describes spatially warped checkerboard stimuli used in retinotopic mapping studies.

Examples of binary visual stimuli are shown in Figure 4. These images have variations in contrast over their area, which is referred to as the spatial frequency of the stimulus. More specifically, the spatial frequency is the number of times the image changes from the minimum to maximum value of contrast in a degree of visual angle (the unit commonly used to describe spatial frequency is cycles/degree) [24]. Furthermore, for most applications, a maximal contrast between the light and dark elements of the stimulus is used to maximize the amplitude of the VEP. Another property of the stimulus that is commonly modified, in particular for visual BCIs, is the temporal frequency of the stimulus. Temporal frequency is defined as the rate at which a pattern of changes (light/dark elements reverse) in the stimulus. Low temporal frequencies are defined as 4 Hz or less and are considered to produce transient responses. High temporal frequencies are defined as more than 4 Hz and

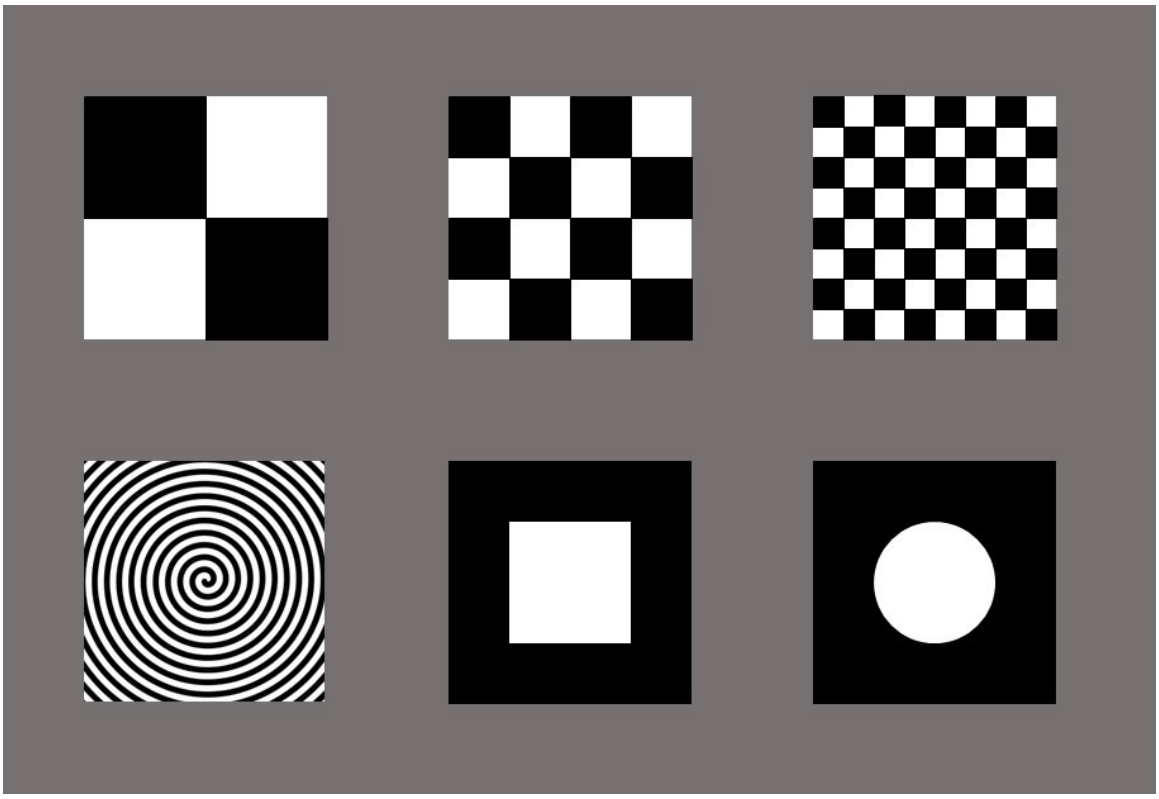


Fig. 4.: Examples of binary stimuli for VEP experiments. The top row of stimuli shows a checkerboard pattern commonly used in BCI and clinical studies. The spatial frequency of the stimuli increases from the left stimulus to the right stimulus. The bottom row illustrates sample stimuli having different spatial patterns. Such patterns ultimately produce different responses in the VEP based on the spatial frequency and geometry of the stimulus.

are considered to produce steady-state responses [25]. These steady-state responses to flashes at a fixed frequency are known to produce oscillations within the visual cortex at the frequency of the flashes and harmonics of that frequency. Both low- and high-frequency stimuli can be in clinical and experimental applications based on the goals of the application, up to the flicker fusion frequency of the visual system at approximately 70 Hz [25].

3.2 Visual BCIs

BCIs are systems that allow users to communicate with external devices without the use of peripheral nerves or muscles. Brain activity for BCIs can be acquired from various modalities, but it is most common to use electrophysiological signals from the surface of the scalp or cortex, as such, these will be the focus of the remainder of this chapter.

Historically, scalp EEG is used in non-invasive BCIs, while stereotactic EEG, electrocorticography (ECoG), or intracortical microarrays are used for invasive systems. Invasive designs are generally considered to produce superior data as recording occurs closer to the source of the electrophysiological activity, increasing the signal-to-noise ratio. Additionally, the spatial is improved as there is little to no interceding tissue between the desired brain region and the electrode. However, non-invasive systems are generally more widely employed as there is no need for a surgical procedure, and well-designed non-invasive systems can be quite effective.

BCIs can be further categorized based on the type of control scheme: active, reactive, and passive. Active BCIs require users to consciously, and directly control the system without an external stimulus by using endogenous mental activity. Reactive BCIs have the user perceive some external, exogenous stimulus and the brain's response to the stimulus is used by the system as an input. Passive BCIs record the user's physiological state and infer aspects of their cognitive state without active participation from the user. The field of visual BCI research has been dominated by reactive BCIs using VEPs due to the reliability of the detection and classification [26].

Some of the first BCIs were visual BCIs, and most prominent designs for non-invasive BCIs today remain visual BCIs. In 1984, Sutter [26] proposed a Speller BCI

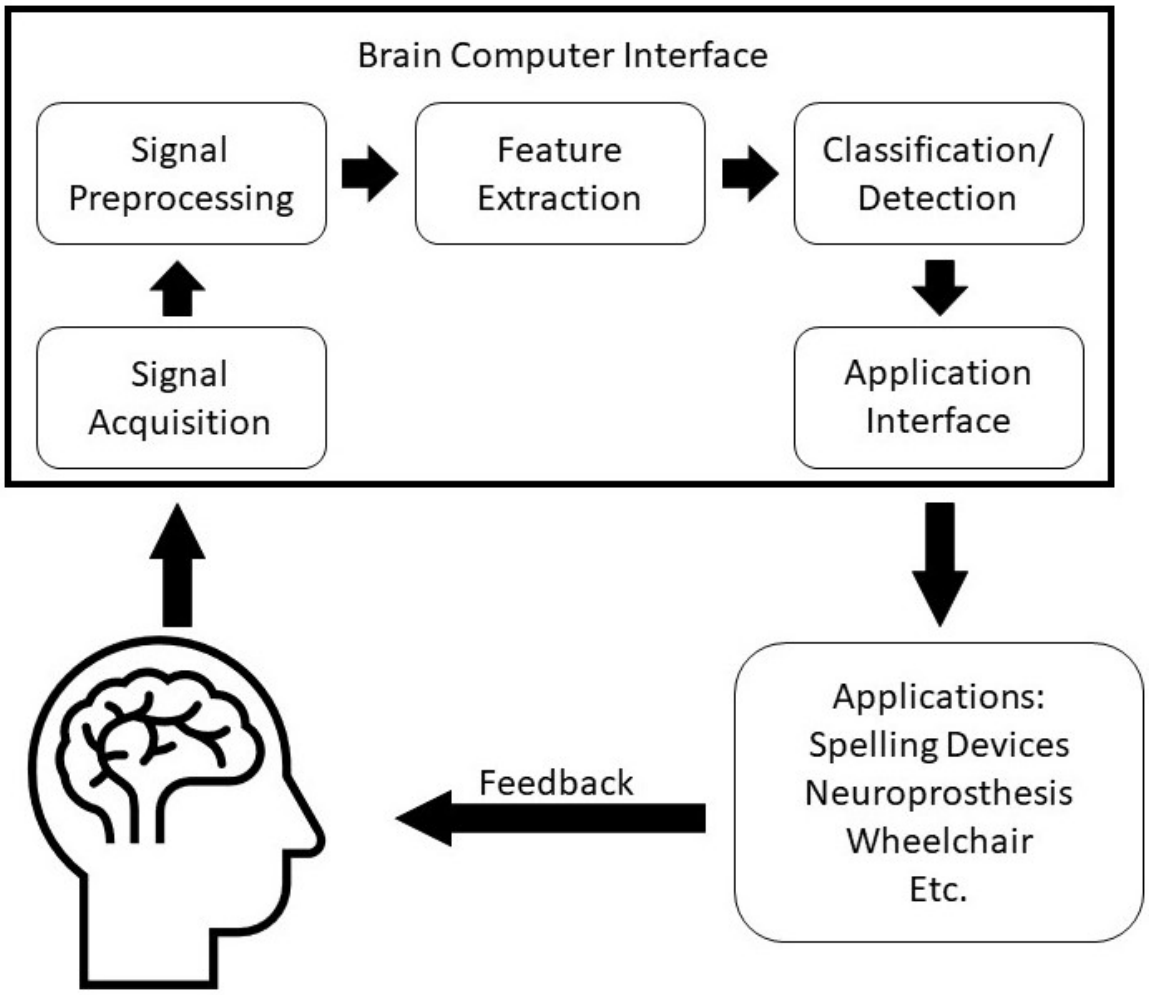


Fig. 5.: Diagram of the components of a Brain-Computer Interface (BCI). Signal acquisition hardware collects physiological data relating to the current state of the brain. These signals are processed to remove noise and enhance the relevant components of the signal. Features related to the application are identified and then classified. The classifier sends commands to the application interface, which can inform the actions to be taken by a device. The user receives feedback (e.g. visual feedback) from the system for BCIs operating in real-time (often called online BCIs) [27].

based on VEPs, and by 1992 he was able to present a functional system with an ALS patient [28]. This design presents a selection of different characters or numbers on a screen that have unique flashing patterns from each other. The response to these flashing stimuli can be distinguished from one another by a classifier and then used as a unique input to a computer. In the case of Sutter's device, and many other Speller BCIs, they present a keyboard to the user with letters as the stimuli. Since Sutter's presentation of his system, an entire subfield within the BCI research community has focused on VEP-based systems for a wide variety of applications and utilizing many different modifications of the basic VEP BCI system.

The two broad categories of VEP-based BCIs are defined by the temporal frequencies of the flashing patterns used in their stimuli [22]. Stimuli presented at low frequencies (less than 4 Hz) are considered to produce transient responses, while stimuli presented at high frequencies (more than 4 Hz) produce steady-state responses. BCIs that use higher frequency stimuli are called Steady-State VEP (SSVEP) BCIs and often use multiple target stimuli simultaneously that flash at different frequencies. This allows SSVEP systems to include as many target stimuli as frequencies can be differentiated from one another during classification, hence providing a greater number of control options for the user. Furthermore, these systems have been shown to not require classifier pre-training by using a CCA-based classification approach (see section 6.2) [29].

Building upon the design of SSVEPs, cVEPs use pseudorandom sequences instead of fixed frequencies for the flashing pattern of the stimuli [30]. This technique does require a training period to develop the classifier, but clever designs such as using the same sequence for all stimuli but with different time shifts, allow for minimal training requirements. The advantage of cVEP systems is that there is less bias across the stimuli compared to the potential EEG frequency specificity of SSVEPs.

CHAPTER 4

ARTIFICIAL NEURAL NETWORKS AS TIME SERIES MODELS

This chapter discusses the types of artificial neural networks (ANN) used to model time series. It begins with a brief overview of what constitutes an ANN. There are a wide variety of approaches to working with ANNs designed for time series based on the goals of the application and the characteristics of the time series in question. Therefore, selecting the proper neural network for a given problem and data set can become quite a challenge. For this chapter, a few general problems will be discussed along with the typical ANN architectures used to solve them. A focus will be given to techniques that are most widely used and those that pertain to the present study.

4.1 Overview of Artificial Neural Networks

ANNs are computational systems inspired by the structure and function of biological neural networks. Biological neural networks consist of neurons that synapse with other neurons in specific pairings and arrangements, which ultimately serve to relay and/or process information. A classic example of a biological neural network that inspired early artificial neural networks [31] are simple and complex cells within the V1 layer of the visual cortex (see section 2.2). Simple cells receive a visual input from the optic nerve and then several simple cells synapse with a complex cell. Each cell in the network triggers (or does not trigger) an action potential based on the excitatory or inhibitory inputs received, and in this configuration, complex cells can synthesize the information generated at the outputs of simple cells. This type of system is mimicked in ANNs by using artificial neurons, which receive inputs and

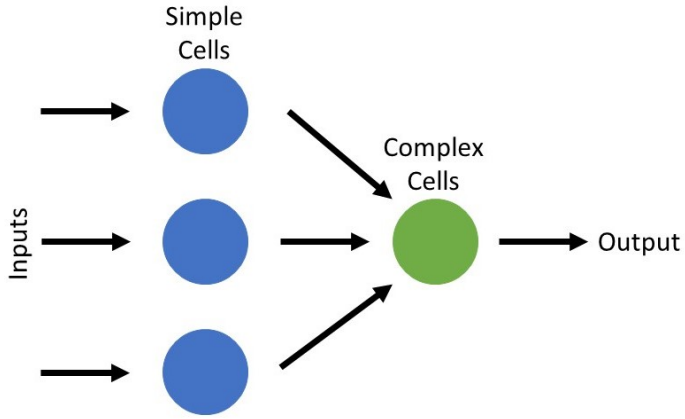


Fig. 6.: Diagram of Simple and Complex Cells. This arrangement was the inspiration for modern artificial neural networks. In the case of biological neural networks, the inputs are sensory information and the output is processed visual information such as object recognition. For ANNs, the input and output are determined by the type of problem aiming to be solved.

transmit a weighted, thresholded version of that input to all neurons it is connected to. The thresholding of weighted values connecting artificial neurons in a network is what ultimately gives ANNs their non-linear behavior. A diagram of this layout is shown in Figure 6, and a diagram of a typical fully connected ANN is presented in 7. Many of the early ANNs were directly inspired by the structure of the visual cortex [32], and many more ANNs aim to solve the same problems as the human visual cortex (this task is referred to as computer vision and is a core area of application for ANN research).

In practice, ANNs can be used as a universal approximator to solve difficult high-dimensional, non-linear problems. In a feedforward ANN, data in the network are represented numerically and pass through the layers of neurons until the final layer of the network produces the output. This output is then compared to a target output using a loss function. The target output is the desired response that is associated with the specified input, and the loss function is typically some measure of the quality of the estimate of network output compared to the target output. The result from

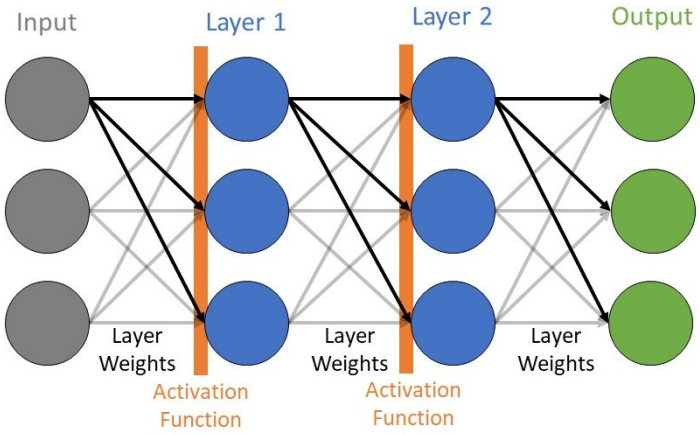


Fig. 7.: Diagram of a typical fully connected ANN consisting of two layers. Circles represent nodes containing values. Arrows represent the weights between layers, and all products of node values and weights are summed before passing through activation functions. Activation functions are responsible for contributing to the nonlinear behavior of ANNs.

the loss function is used by an optimization technique called backpropagation [33], to minimize the loss function by modifying the weights between neurons in the network based on an optimization algorithm such as stochastic gradient descent [34], adaptive moment estimation (ADAM) [35], or root mean square propagation (RMSprop) [36].

4.2 Problem Domains for Artificial Neural Networks

ANNs have become very powerful, prominent tools in a large number of domains due to their capability to learn solutions to a problem. While ANNs are considered universal approximators, assuming proper and sufficient training data is provided, there are many ways to configure the network architecture and the selected architecture should be appropriate for the application. Determining what types of ANN architectures perform best on a given problem has become a research field itself, and has led to the refinement of how problems and ANN architectures are described. ANNs can be used to solve both classification and regression problems, which respectively require different design considerations.

Classification problems aim to classify the data into two or more groups. The simplest case of classification is the binary classification of two groups. While this is the simplest conceptualization of classification, binary classification problems may have highly complicated decision boundaries. Furthermore, binary classification is often used within more sophisticated networks to augment the output of the network based on the classification that is made, such as in Generative Adversarial Networks (GAN) [14]. Classification can also be used for multiclass problems in which there are three or more groups for the data to be sorted into. There are several techniques for how to handle multiclass classification, but most often the network output is scaled to represent the probability that the input is a member of each of the classes. Then the class with the highest probability is selected as the one associated with the input.

For regression problems, ANNs have the goal of producing specific values (possibly over a continuum) based on the input given to the network. In theory, the size, dimensionality, and organization of the output are independent of the size, dimensionality, and organization of the input, which is only possible due to the inherent flexibility of ANNs arising from the ability to change connections between nodes. However, ANNs targeted at solving regression problems are most commonly configured to have the number of points in the input greater than or equal to the number of points in the output (this particular matter is relevant to the network employed in this study; see section 7.1.1).

Regression problems are most often in the form of system identification or prediction problems. System identification problems attempt to generate the input-output mapping of an unknown system. Prediction problems aim to use past values of a time series as the input to predict future values of the time series as the output. This configuration of a prediction problem is referred to as forecasting.

A third problem domain for ANNs is generation problems. These problems task

the network to generate a new example of a desired output. This can be based on a given input to the network or essentially from scratch if the network is designed to accept random noise as an input. These differ from regression in the fact that the model output is of variable length and nearly always uses the underlying concepts in the input to determine the output. Generation problems also include the creation of synthetic data. This type of problem aims to generate a larger data set off of an original small data set when the option to record more data is not feasible [14, 37]. This application of generation usually uses noise as the input which the network is then able to mold into an output in-line with the characteristics of the original data set.

4.3 Network Architectures for Time Series Problems

ANNs have been used to solve a wide array of problems including those relating to time series. One of the key properties of time series data is that the order of data points contains information. This has a drastic impact on how ANNs are designed to work with time series data, and accordingly informs the most prevalent types of neural networks for working with time series data.

4.3.1 Time Series Prediction and Forecasting

The most common problem ANNs are used to solve regarding time series data is forecasting. It is possible to design networks capable of predicting future events of the same time series such as stock market trends, weather patterns, or medical events [38]. The black box nature of ANNs is rarely a concern for these types of issues so long as the network is capable of accurately predicting future events. A common architecture for time series forecasting is recurrent neural networks (RNNs). These networks allow for feedback from the output of network nodes back to the

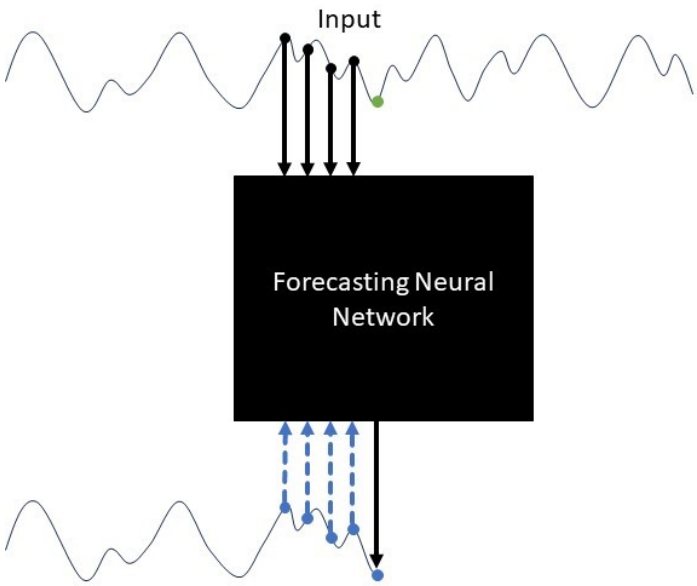


Fig. 8.: Example of a typical Forecasting Neural Network for Time Series. Several of the past points of the time series along with the current point act as the input to the next. Additionally, a common feature among many forecasting neural networks is utilizing several of the past output points as another input into the network (indicated by the blue dashed lines). The network produces an output with the value of the next predicted point in the time series (the green point indicates this target value).

same nodes or prior nodes in the network. This type of layout produces a system that acts in a way somewhat analogous to an infinite impulse response filter [39]. The unique strength of RNNs is that they use previous outputs as a feedback mechanism to adjust the weighting of neurons in the network. While RNNs can be more difficult to train compared to feedforward networks, they can provide more robust estimates. Common types of RNNs include Long Short-Term Memory (LSTM) networks, Gated Recurrent Unit (GRU) networks, and Fully Recurrent Neural Networks (FRNN).

An ANN architecture that has become increasingly popular for time series prediction and forecasting is a transformer [40]. These ANNs are primarily known for their significantly improved performance with natural language processing tasks but have been modified to work for time series prediction as well. The performance of these networks may not be drastically better than RNNs for time series prediction,

and they require a significantly larger number of stored weights. Thus, RNNs are still considered are still commonly utilized for time series prediction.

4.3.2 Classification of Time Series

Convolutional Neural Networks (CNNs) are the most prevalent architecture used for classification due to their straightforward underlying structure, low number of stored weights, and high performance [41]. These networks use a series of convolutional layers as filters on the input to identify the most critical components of the time series for classification. The final layer of the network can be a reduction of the sequence to a single value representing the output classification or a series of probabilities that the input belongs to each of the classes. CNNs are set apart from other architectures in the fact that they intrinsically maintain the short-range temporal relationship between points in a time series. This is due to their property of simply being a series of convolutional filters, referred to as kernels, which pass along the length of a time series. It should be noted that long-range temporal relationships (those spanning distances greater than the kernel length) are not captured by this intrinsic structure of CNNs. Despite this limitation, CNNs are commonly used for time series classification.

4.3.3 Regression of Time Series

The most prevalent ANN architecture for system identification of time series is feed-forward fully connected linear networks (FFNs) because the network is designed able to learn the time dependence between points. These networks receive a sequence of lagged time points, which pass through several fully connected linear layers until the output layer, which transforms the output to the size of the regression output [42]. The fully connected nature of the layers in the network means that each neuron

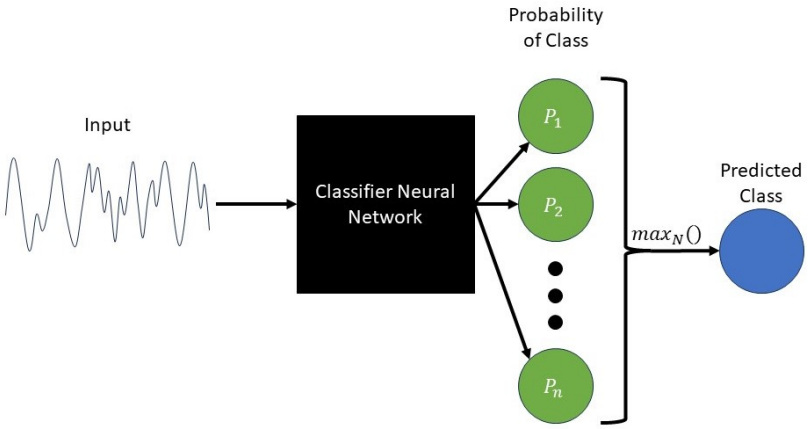
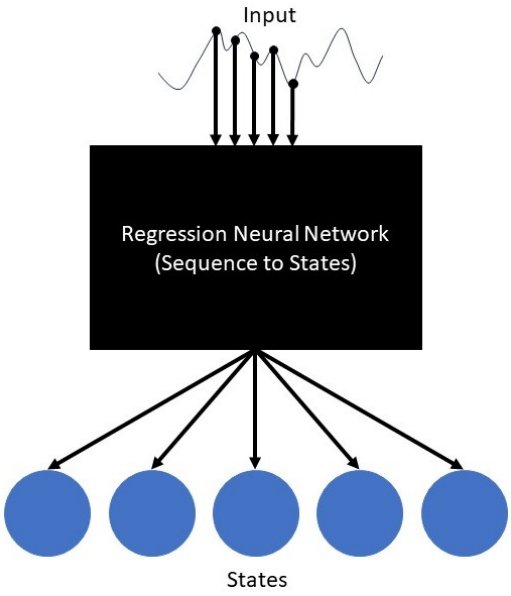
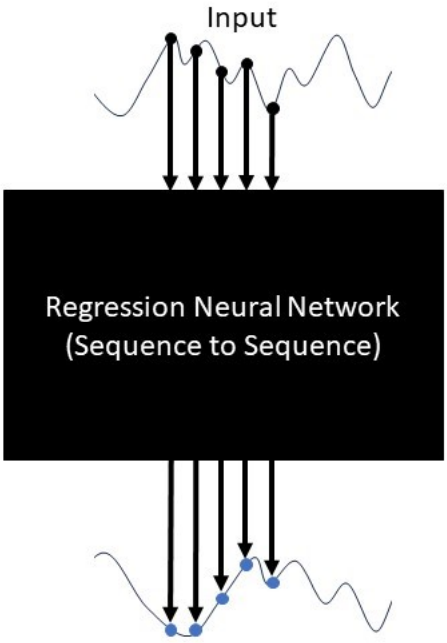


Fig. 9.: Example of a typical Classifier Neural Network for Time Series. The lagged time series is the input into the network. The network produces a probability value for how likely the input belongs to that class. The class with the maximum probability is selected as the output class for the network.

is connected to all neurons in the subsequent layer, making the network weights challenging to interpret. There are many different subtypes of FFNs based on additional or removed connections as well as the dimensions of each layer in the network, such as Encoder-Decoder architectures or Residual Deep Neural Networks, but the underlying premise of FFNs remains widely the same across these subtypes. Differences typically relate to relatively minor improvements in performance or are supposed to aid in making the network learn in a specific pattern. The architecture used in the model developed in the present work is an FFN with residual connections to the input added to improve performance (see section 7.2).



(a) Example of a typical Sequence-to-States Neural Network.



(b) Example of a typical Sequence-to-Sequence Neural Network.

Fig. 10.: Examples of typical Regression Neural Networks for Time Series. Figure 10a depicts a sequence-to-states network, which takes a lagged time series and produces outputs containing information about some state of the system the time series represents. Figure 10b depicts a sequence-to-sequence network, which takes a lagged time series and produces an output time series based on the information within the input. Note: The ReCon FFNN designed in the present work is a sequence-to-sequence regression network.

CHAPTER 5

EXPERIMENTAL DESIGN AND PROTOCOL

This chapter presents the details of the design and execution of the visual response experiment performed to generate new data for modeling the visual system. Details on the human participants, the equipment and software used, the format of the experimental session, and the visual stimuli used are provided in this chapter. This includes the relevant considerations to ensure the aims of the investigation were thoroughly addressed by the resulting data and analysis.

5.1 Participants

Data were collected from 15 able-bodied participants (7 female, 8 male, ages 22-59, mean 31±12). 7 participants had 20/20 vision, and 8 had corrected to normal vision. The participants with normal vision, and the single participant with corrected vision who wore contact lenses daily, used the head-mounted display normally. Interestingly, the remaining participants with corrected vision (7 total) elected to not wear their glasses while using the head-mounted during during the experimental session due to a comfort issue. However, each participant in this group stated they were readily able to clearly see the spatial stimulus features and contrast changes without glasses.

5.2 Experimental Task

During the experiment, the participants were asked to perform a simple visual task while their electrophysiological responses to the task were recorded by an EEG

system (see section 5.5). The resulting EEG data was used to inform the construction of the computational model of the visual system. For the task, the participants were asked to focus their gaze on a pattern-reversal visual stimulus shown in Figure 11, presented in the form of a video. This stimulus was designed to maximally excite the visual system, based on those frequently used in experiments concerning the visual system [24] (see section 3.1). A feature present in nearly all checkerboard or retinotopic stimuli is the high contrast between the spatial components of the stimulus. Typically, pure white and black are chosen as they produce the highest contrast differential between any two colors, and, due to the physiology of the retina, this high contrast maximally excites the visual system. The visual field around the stimulus was presented as a 50/50 gray tone (exactly halfway between the white and black of the stimulus). This was done to ensure that the peripheral visual field would not distract the participant or influence the response.

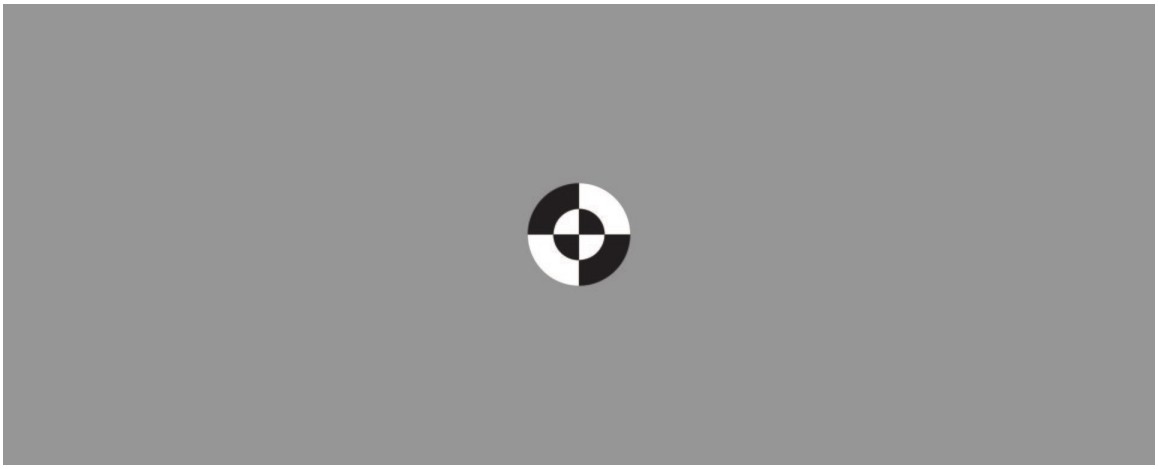


Fig. 11.: The Checkerboard Visual Stimulus used for the experiment. The white and black regions would alternate contrast according to a predetermined binary sequence to create a flicker pattern. The stimulus was positioned in the virtual space to appear to be 60 centimeters away from the participant's head, and scaled to be 10 centimeters in diameter. This resulted in the stimulus taking up 7.3^o of visual angle.

The size and dimensions of stimuli were chosen based on feedback from early

pilot testing of the system. Previous work suggests that a balanced, moderate spatial frequency of the subdivisions of the stimulus may result in better responses from the participant [24]. During pilot testing, it was found that roughly 1 cycle per degree of visual angle (just less than that determined to be optimal in [24]) both elicited stronger responses and was more comfortable for the participants. During testing, as more divisions were added to the stimulus, the resolution at the center of the stimuli became less clear. Since the center of the stimulus is where participants were asked to focus their gaze, this resulted in participants reporting greater visual fatigue and lower overall signal quality of the EEG visual responses. Differences between the hardware used to deliver the stimuli in this study in comparison to previous work, such as using a head-mounted display rather than a computer monitor, are a likely explanation for this difference in spatial frequencies.

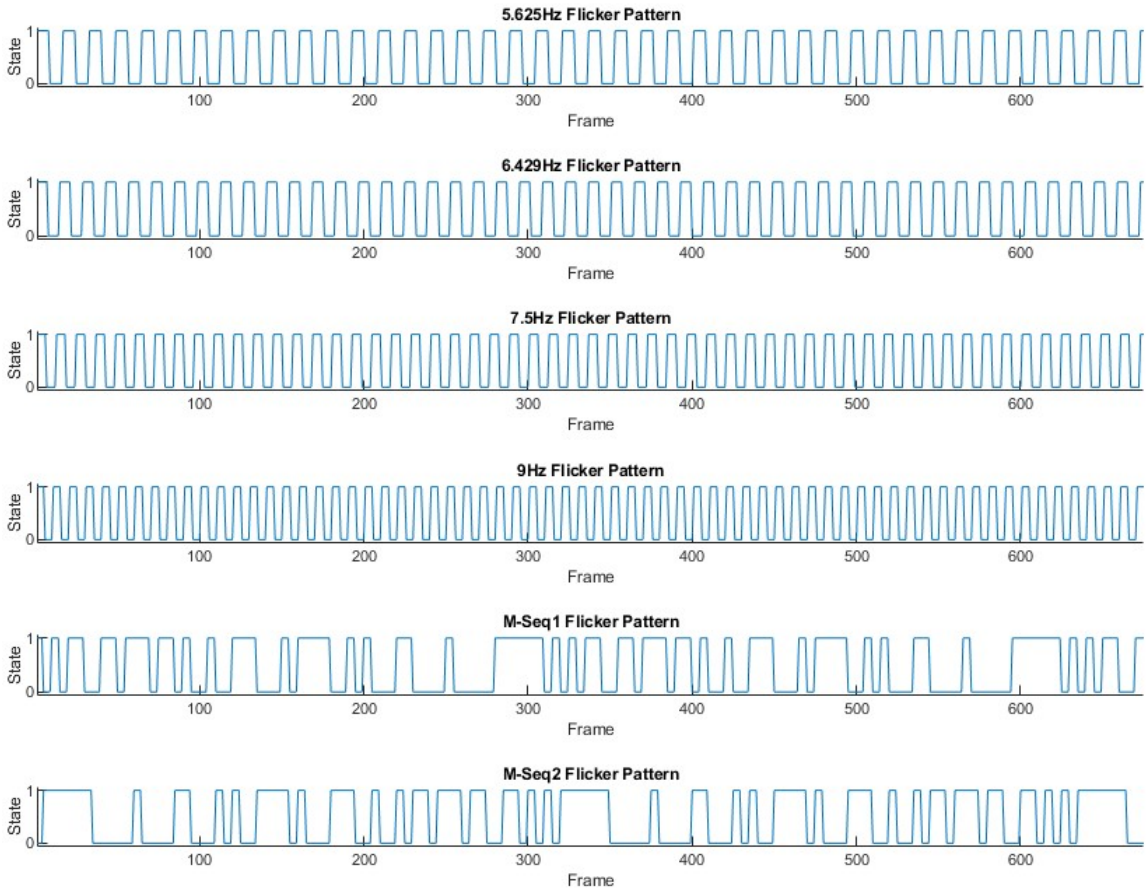
Another feature of the stimulus is its circular shape. This shape was chosen to reflect the roughly circular shape of the fovea within the retina. Due to the foveal area containing the highest concentration of photoreceptors within the retina, it is responsible for producing the highest resolution area of the visual field. When fixating gaze at a point in space, the fovea is aligned with that region to produce the highest resolution image of that point, and a proportional amount of visual processing within the brain occurs to manage this large amount of information. The stimulus was displayed at the center of the video and at a visual angle (7.3 degrees of visual angle) slightly larger than that of the fovea (approximately 5 degrees of visual angle [43]).

Another property of the visual stimuli is the rate at which the divisions alternated between the binary colors (i.e., changed from black to white), or in other words the rate at which the stimulus "flickered" or "flashed", henceforth referred to as the flicker frequency. Similarly, the temporal patterns at which the stimuli flickered are referred to as the flicker pattern. There were two distinct types of flicker patterns

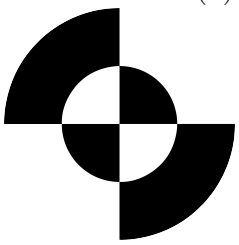
implemented: fixed frequencies, being similar to those used in SSVEP-BCIs, and pseudorandom, being similar to those in cVEP-BCIs.

Fixed frequency patterns consisted of the stimulus division alternating between colors at a fixed time interval. There were four unique frequencies used in these fixed frequency patterns: 5.625 Hz, 6.429 Hz, 7.5 Hz, and 9 Hz. These frequency values refer to the number of duty cycles the stimulus completed over one second. One duty cycle consists of the divisions of the stimulus appearing as their starting color for 50% of the time and then appearing as the alternate colors for the remaining 50% of the time and can be thought of as the period of a wave as the pattern repeats in the following duty cycle. These specific frequencies (5.625 Hz, 6.429 Hz, 7.5 Hz, and 9 Hz) were chosen due to hardware considerations relating to the 90Hz refresh rate of the display being used (see section 5.5), frequencies used in previous studies [44], and being safely below the frequencies typically linked to epileptic seizures (approx. 30 Hz). Specifically, concerning the stimulus presentation hardware, the refresh rate of the visual display limits the frequencies that can be used. This is primarily due to the duty cycle needing to have an equal number of frames presented for both the starting and alternate states of the stimulus. This results in the duty cycle needing to last for an even number of frames. Accordingly, the frequencies chosen follow this principle: 5.625 Hz = 16 frame duty cycle, 6.429 Hz = 14 frame duty cycle, 7.5 Hz = 12 frame duty cycle, and 9 Hz = 10 frame duty cycle.

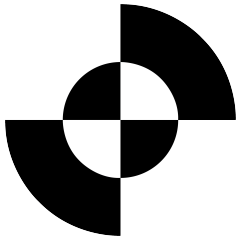
The second type of pattern used to present the stimulus was a pseudo-random binary sequence. An m-sequence was employed to produce these pseudo-random sequences as is common practice among cVEP BCIs [45]. M-sequences are pseudo-random binary sequences generated by a maximal linear feedback shift register of a given period length. The length of the shift register determines the length of the resulting output sequence's period. A shift register with a length of X would result



(a) Sample Stimulus Sequence Segments.



(b) Stimulus State when Binary Pattern = 0



(c) Stimulus State when Binary Pattern = 1

Fig. 12.: Sample Stimulus Sequence Segments. The respective plots show the flicker patterns across the first 7.5 seconds of a trial for each stimulus type. A value of 0 denotes the checkerboard visual stimulus in the starting orientation, and a value of 1 denotes the stimulus with the contrast reversed. The x-axis is in units of frames of the video, which was presented at 90 frames per second.

in an output sequence period of 2^X bits long. The given output sequence could be repeated for an infinite number of periods. The specific sequence of bits is determined by the initial values chosen to use with the shift register. Any combination of binary values can be used, except for a sequence consisting of exclusively zeros, which would only produce a sequence of zeros regardless of the shift register.

M-sequences have a handful of properties that make them desirable for performing experiments with visual BCIs [46] and as such have seen widespread use in the field [47, 48, 49, 50, 51]. The first property is the balance property, which states that there will be approximately the same number of ones and zeros in a given sequence (more specifically in a 2^n - 1 length sequence there will be 2^{n-1} ones and $2^{n-1} - 1$ zeros). The second property is the run property. In a m-sequence, there are many sub-sequences of consecutive 1s or consecutive 0s, and these are referred to as "runs." Within a given sequence, one-half of the runs are of length 1, one-quarter of the runs are of length 2, one-eighth of the runs are of length 3, and so on. These first two properties result in the m-sequences being able to stimulate across a wide band of frequencies when used as the flicker pattern for visual stimuli. The third property is the autocorrelation property. When a m-sequence is given a time delay it is effectively producing a circular shift within the given sequence. Taking the correlation between the original and the time-delayed sequence produces a minimal correlation. In practice, the utility of this property is that the same sequence can be time-shifted and used for a separate target of a visual BCI. This is the primary reason m-sequences are commonly used in cVEP BCI studies.

Due to these desirable properties, an m-sequence was used for several stimuli in the present study. Specifically an m-sequence with a shift register of length six resulting in a sequence of length 63 ($2^6 - 1$). The input sequence to the shift register was [101011]. MATLAB Code from Buracas and Boynton (2002) [52] was utilized to

generate the sequence. The exact sequence used in the experiment in hexadecimal notation was **566ED2717946107E**.

This sequence was presented with a one corresponding to the starting state of the stimulus and a zero corresponding to the alternate state of the stimulus, at a bit rate of 18 Hz. This bit rate was chosen as it allowed the maximum flicker frequency to be 9 Hz, which would occur for the sub-sequence 010 or 101 in the sequence. Furthermore, the time-reverse of the above sequence was presented as a separate stimulus pattern (the forward direction is referred to as *m-Sequence 1* and the reverse direction is referred to as *m-Sequence 2*).

Ultimately, the new sequence generated this way has the same frequency spectrum as the other pseudo-random sequence, which was expected to yield interesting relationships when used to generate the model of the visual system. Furthermore, these pseudo-random sequences can produce wide-band stimulation of the visual cortex, and thus the model generated from the data collected from this experiment would be substantially more robust if trained on both random and fixed frequency flicker patterns. A summary of the various patterns used for the primary task can be found in Table 1.

5.3 Fixation Task

A secondary task was used to ensure that the participants kept their gaze focused on the center of visual stimuli to maximize the visual response. This fixation task consisted of the participants clicking a button on a handheld remote each time a red dot appeared at the center of the visual stimulus. These red dots appeared at two random times during each trial and lasted for 0.5 seconds. Specifically, a red dot would appear randomly between 5 and 24 seconds into the trial, and again between 25 and 29 seconds into the trial. These specific timings were selected to ensure that

Stimulus Type	Description
5.625 Hz	Fixed frequency stimulus with a duty cycle being completed at a rate of 5.625 Hz.
6.429 Hz	Fixed frequency stimulus with a duty cycle being completed at a rate of 6.429 Hz.
7.5 Hz	Fixed frequency stimulus with a duty cycle being completed at a rate of 7.5 Hz.
9 Hz	Fixed frequency stimulus with a duty cycle being completed at a rate of 9 Hz.
M-Seq1	Pseudorandom stimulus following a binary m-sequence pattern presented at 18 Hz, which results in the fastest duty cycle of the sequence being completed at a rate of 9 Hz.
M-Seq2	The same sequence as M-Seq1, presented in the reverse order.

Table 1.: Description of Stimulus Types. The various flicker patterns used as different stimulus types within the experiment. Each was presented twice to each participant: once in the first 6 trials and once in the last 6 trials. Fixed Frequency rates are related to hardware considerations, see Section 5.2

the participant was paying attention for the entire length of the trial. The EEG data from the trial were synchronously recorded with the time stamps of the dot appearances and participant button presses. This information would be used during preprocessing to remove segments of the data where the participant was moving to ensure there were as few motion artifacts as possible within the data.

5.4 Trial and Experimental Session Structure

The experimental session was divided into 12 trials. Each trial was assigned a stimulus type from the different types discussed in the previous section. There were a total of 6 different stimulus patterns used in the experiment, which are described in Table 1. These patterns were randomly assigned to the first 6 trials and the last 6 trials independently. This made it such that each stimulus would appear once in the first half of the trials and again in the second half.

Each trial consisted of 3 components: the pre-trial countdown/baseline period, the task period, and the rest period. Preceding each task period, the user would watch the grey screen for two seconds and then a three-second countdown would appear on the screen to inform them that a task period was about to begin. The visual stimulus appeared while the participant's EEG was recorded. Within each task period, the participant would simultaneously perform the primary task and the secondary fixation task. The participants were asked to minimize their blinking and keep their gaze fixated on the center of the stimuli during the trials. After the trials, the participant was allowed to take a self-paced break which would end if they pressed the button they used for the fixation task. After this break, the next trial began with the pre-trial countdown for the remainder of the trials. A diagram of the trial structure is presented in Figure 13.

The data collected from each participant included the EEG recording and trial-

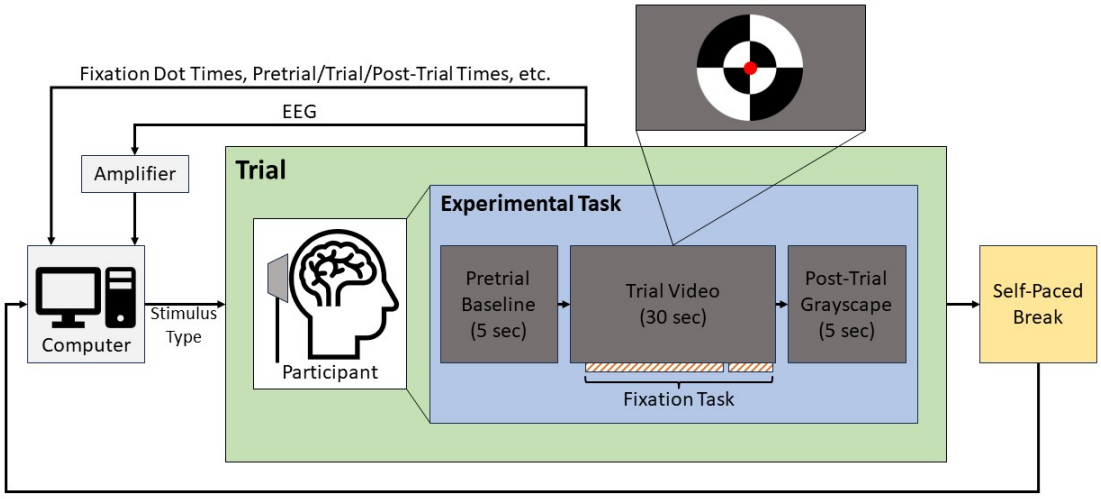


Fig. 13.: Diagram of the experimental design and trial structure. The computer managed controlled the visual stimuli via Unity and recorded EEG and relevant task information via BCI2000. The fixation task blocks denote possible times when the fixation dot could appear. The fixation dot was presented for a half second duration.

specific data. This trial-specific data included the participant's head position and rotation, the start time of the stimulus, the timing of each component of the trial, the times the dot for the fixation task was on screen, and the timing of button presses for the fixation task.

5.5 Data Collection Equipment and Software

The EEG recordings used 8 passive cup electrodes connected to a g.USBamp amplifier from g.tec (Guger Technologies, Austria) [53] with a sampling rate of 256 Hz. Signals were recorded using the BCI2000 software platform [54], which also was used to synchronize and record the timing of all the events within a trial. The positions of the recording electrodes were POz, PO1, PO2, POO3, POO4, Oz, O1, and O2 according to the International 10-20 system. The ground and reference were placed on the mastoid on opposite sides of the head. Figure 14 contains the electrode montage. This organization of the electrodes is a modified version of the layout used in [44].

Unity and C# were used to implement the experimental design. Playback of the stimulus videos and the timing of the different components of each trial were managed by Unity. The trial information such as the stimulus pattern and current trial component were recorded by exporting this information from Unity to BCI2000. The display used for the videos was a HTC Vive headset operating at a 90 Hz refresh rate. This headset was used to completely isolate the user's field of vision from potential distractions that may affect the EEG. The 90 Hz refresh rate used by the HTC Vive is common for such virtual reality headsets, as frames displayed at a rate lower than this typically cause simulator sickness. This particular feature of the hardware resulted in the specific values selected for the stimulus flicker frequencies discussed previously (see section 5.2), as the flicker frequencies are determined ultimately by the number

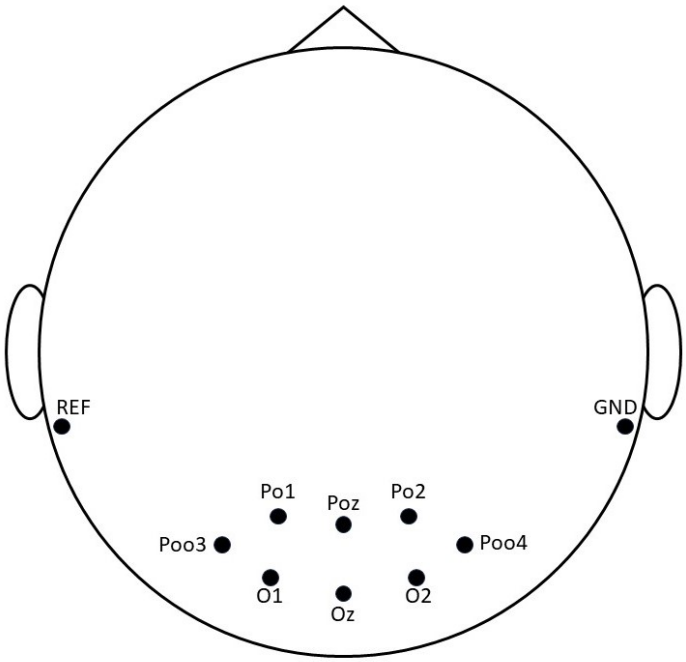


Fig. 14.: Electrode Montage. The electrode montage used to record EEG from the visual cortex for the experiment presented in this thesis. Electrodes are positioned and labeled according to the International 10-20 system.

of frames spent in each state by the stimulus.

CHAPTER 6

DATA PREPROCESSING AND MULTICHANNEL SPATIAL FILTERING

This chapter presents the details of the preprocessing of the EEG data before use in the visual model. The filtering and segmentation of the data to remove the portions of the trials containing the button presses are covered. Details on the techniques used to convert the recorded EEG time series from the set of 8 electrodes to the single channel time series ultimately used for the model are also discussed. The chapter concludes with the windowing procedure used to properly format the data for use with the model.

6.1 Data Preprocessing

Data from each trial was recorded by BCI2000 and saved after the experimental session for each participant was completed. The files were subsequently preprocessed using MATLAB [55]. The signals for all trials were bandpass filtered from 3.25 Hz to 55 Hz using a minimum-order, zero-phase FIR filter with a stopband attenuation of 60 dB. To remove any potential artifacts from the button press during the fixation task, the time spanning from the first frame of the dot appearance until a second-and-a-half later was removed from the EEG data for both dot appearances for each trial. This resulted in a usable trial length of 27 seconds. A bar plot of the button press reaction times to the dot appearance of the fixation task for all participants is shown in Figure 15. It should be noted that the average reaction times were all well within the second-and-a-half window. This, along with the EEG responses from

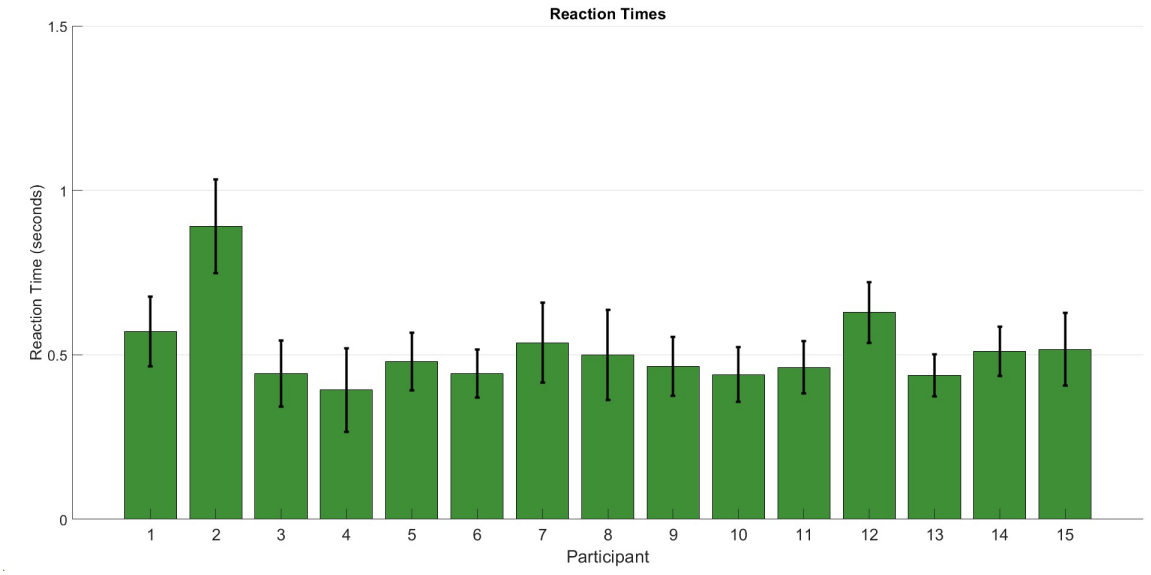


Fig. 15.: A bar plot of the button press reaction times for all participants for the fixation task. The fixation task was intended to confirm that the participants foveated on the center of the visual stimulus to maximize their response in the EEG. The average reaction times all fell well within the bounds of the portion of the recording excluded to eliminate potential motion artifacts of the associated button presses. This indicates the fixation stimuli were reliably detected across participants and trials, and is confirmed by the EEG responses (see Figure 19).

all participants (see section 8.1), indicates that the participants reliably maintained focus on the center of the visual stimulus for the duration of the trials. All trials were visually inspected to ensure the recording and preprocessing had been properly performed.

The data from each EEG channel for each trial were subsequently normalized by subtracting the mean and dividing by the standard deviation of the average of the given channel's baseline periods before each trial was recorded. This baseline normalization ensures that a single channel does not bias the result if there is any impedance mismatch across the channels.

As the proposed computational model discussed in the subsequent chapter was designed to accept the stimulus flicker pattern as an input (see section 7.1.1), the

flicker sequences of each trial's pattern and fixation task dot appearances were generated at the same sampling rate (256 Hz) as the EEG data. This was achieved by taking the stimulus videos from each trial and selecting a single pixel (the same pixel located in the outer top right region of the stimulus was used for all trials and all participants) from the visual stimulus. The color value of this pixel was defined as 1 for white and 0 for black. The video was upsampled from 90 Hz to 256 Hz to match the sampling rate of the EEG. A single pixel was used because the other pixels in the stimulus represent perfectly redundant temporal information and the proposed model was focused on reproducing the temporal, not spatial, patterns. This resulted in a single-channel binary sequence of ones and zeros representing the states of the checkerboard visual stimulus throughout the entire 30-second trial. The time spans of the fixation task dot were also excluded from the analysis in the same fashion as described for the EEG data to maintain data alignment.

6.2 Multichannel Spatial Filtering

The present study was designed as a starting point for a larger line of research aimed at being able to model the spatio-temporal dynamics of the human visual system. Due to the complexities of jointly modeling spatial and temporal information in the visual system, this work exclusively focuses on modeling temporal dynamics. This is accomplished by relating the single pixel input (representing the temporal flash pattern of the visual stimulus) and a single, transformed EEG time series.

To derive a single EEG time series that is most representative of the visual stimulus-response across the 8 recorded channels, a linear combination of the channels (i.e., a spatial filter) was created using the canonical coefficients produced by canonical correlation analysis (CCA) for each trial [56].

CCA is a multidimensional statistical analysis technique that is applied to find

underlying correlations between two data sets, and ultimately produce canonical coefficients and canonical variables [29]. Given two multidimensional datasets, X and Y, linear combinations $x = X^T A$ and $y = Y^T B$ can be found which maximize the correlation between x and y. A and B are referred to as canonical coefficients while x and y are called canonical variables. A and B contain multiple sets of coefficients depending on the dimensions of the two datasets of interest. The vectors of coefficients within A and B can be found by solving the optimization problem:

$$max_{W_x,W_y} \frac{E[W_x^T X Y^T W_y]}{\sqrt{E[W_x^T X X^T W_x] E[W_y^T Y Y^T W_y]}} \tag{6.1}$$

Practically, this is solved using a singular-value decomposition method to diagonalize the covariance matrices as the maximum canonical correlation corresponds to the square root of the largest eigenvalue of the solution.

In the context of BCIs, CCA is used to generate spatial filters for multichannel EEG data. The most common application of these spatial filters is the classification of SSVEP or cVEP data. Classification is performed by generating 'templates' of characteristic responses to specific visual stimuli which can then be used as the second dataset for the CCA. The output canonical coefficients for the EEG and all possible stimuli templates are compared, and the pair with the largest coefficients is selected as the corresponding class.

The templates used for this application are critical as the accuracy of the classification depends on the template's ability to represent the actual response expected to be observed in the EEG. These templates can be generated in several ways, but the primary factor determining how templates are generated is the flicker pattern used to generate the VEP. In the case of SSVEPs, templates are traditionally generated using sine and cosine waves at the fundamental flicker frequency and a selected num-

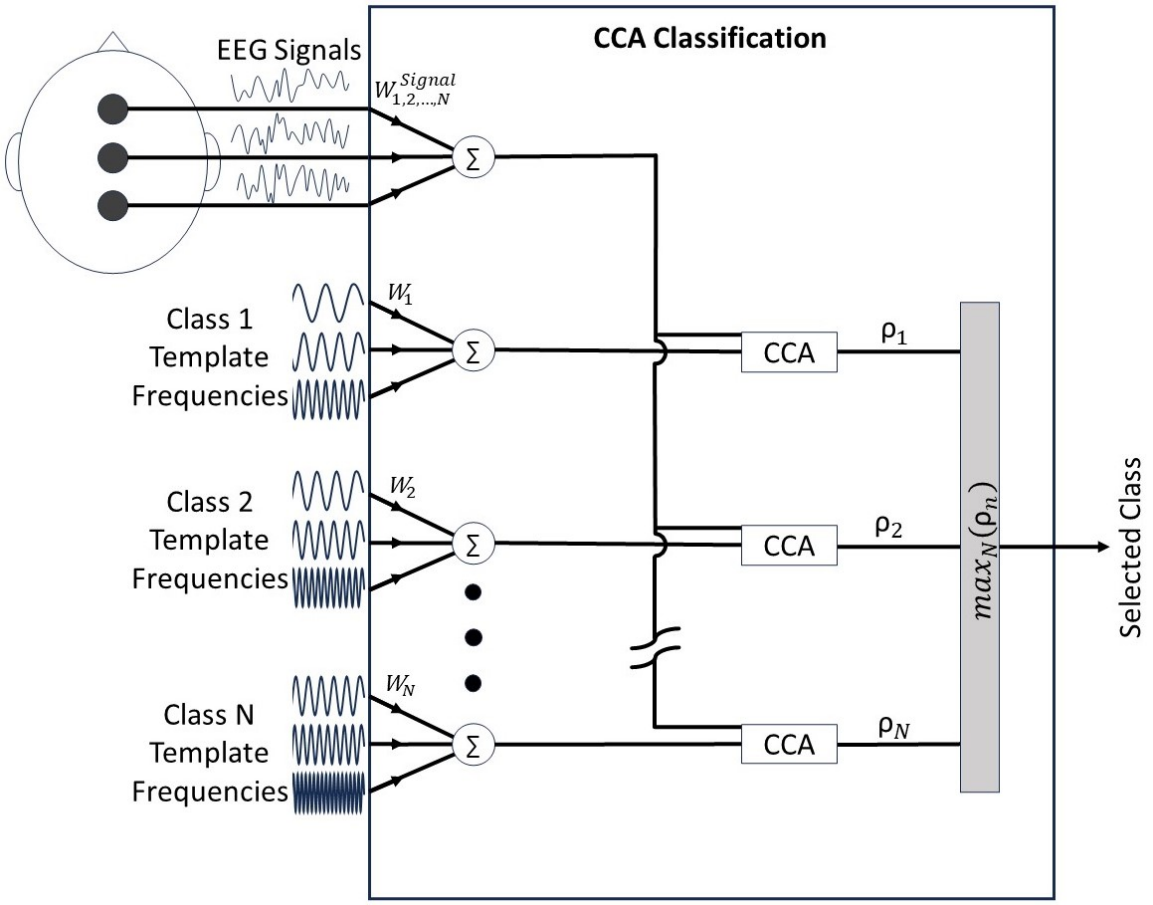


Fig. 16.: Diagram of CCA Classification for an n-class paradigm adapted from [29]. For all N targets, the weighted sum of EEG signals is correlated with a weighted sum of the template signals. Templates are drawn as sinusoids but can be any waveform. The optimal weights are computed with CCA and produce a maximal Pearson correlation coefficient. The class corresponding to the templates with the maximum correlation is selected as the class of the input EEG signals.

ber of harmonics of this frequency. The justification governing this approach is that the strong response seen in SSVEP EEG at the fundamental flicker frequency and its harmonics will result in the correlation between the templates at the same frequencies being higher than that of the other templates at other frequencies. Typically, only the second and third harmonics are due to the 1/f attenuation of the EEG spectrum.

The use of non-fixed frequency flicker patterns in cVEP paradigms results in a different approach to generating templates being necessary. A calibration period of the experiment is performed to collect data from the response to the coded flicker pattern(s) being used [30]. The coded flicker pattern is repeated many times during this training period. After the training period concludes, the EEG responses to each of the repetitions of the coded pattern are averaged. This average response is then used as the template for the actual testing portion of the experiment to classify which stimulus the participant is attending. While this approach to generating templates is relatively crude and relies heavily on generating training data, it has proven to be effective at generating strong classification results from CCA [30].

In the context of the present experiment, CCA was not used for classification, but rather for dimensionality reduction. As previously mentioned, the goal of the experiment presented in this study was to model the temporal behavior observed in SSVEP and c-VEP EEG data using the stimulus flicker pattern as an input. This required that the channels of the EEG data be combined to consolidate the spatial information from the data. Using the canonical coefficients generated from CCA as weights for a spatial filter produces a single channel that best captures the aggregate response to the stimulus. Further differences from the typical VEP classification study format include the absence of the calibration period, the use of both steady state and coded flicker patterns, and the exclusion of timespans relating to the fixation task. These differences led to the resulting implementation of CCA discussed below.

The templates for each trial were generated based on the flicker pattern used in the given trial. For the fixed frequency trials (5.625 Hz, 6.429 Hz, 7.5 Hz, or 9 Hz), the templates used were sine and cosine waves at the fundamental flicker frequency and the respective 2nd and 3rd harmonics. For m-sequence trials, the templates were generated using a modified averaging approach compared to the conventional method typically used for cVEP experiments. As there were two trials for each of the flicker patterns used, one trial of the same m-sequence was used to generate the template. The 8 repetitions across the trial were averaged together and concatenated with itself until it reached the length of the EEG data recorded for each trial. Regardless of which stimulus was used in a trial, the times corresponding to button presses for the given trial were then removed from these templates to align them with the given EEG data. CCA was then performed using the 8 channels from the EEG and the corresponding template.

For the fixed frequency trials, the first vector of weights from the CCA was applied to the channels to produce the composite signal that would ultimately be used for the model. This composite signal using the weights from the first set of canonical coefficients is called the first canonical variable. In the case of the m-sequence trials, only one set of weights was generated due to the template being one-dimensional. This set of weights was applied to the EEG data to generate the composite signal for the model. A block diagram of this spatial filtering procedure is presented in Figure 17

6.3 Downsampling and Windowing

The spatially filtered EEG and flicker pattern were decimated by a factor of 2 to give a new sampling rate of 128 Hz, as the data had already been bandpass filtered up to 55 Hz. The data were segmented into 2-second (256-point) windows with the

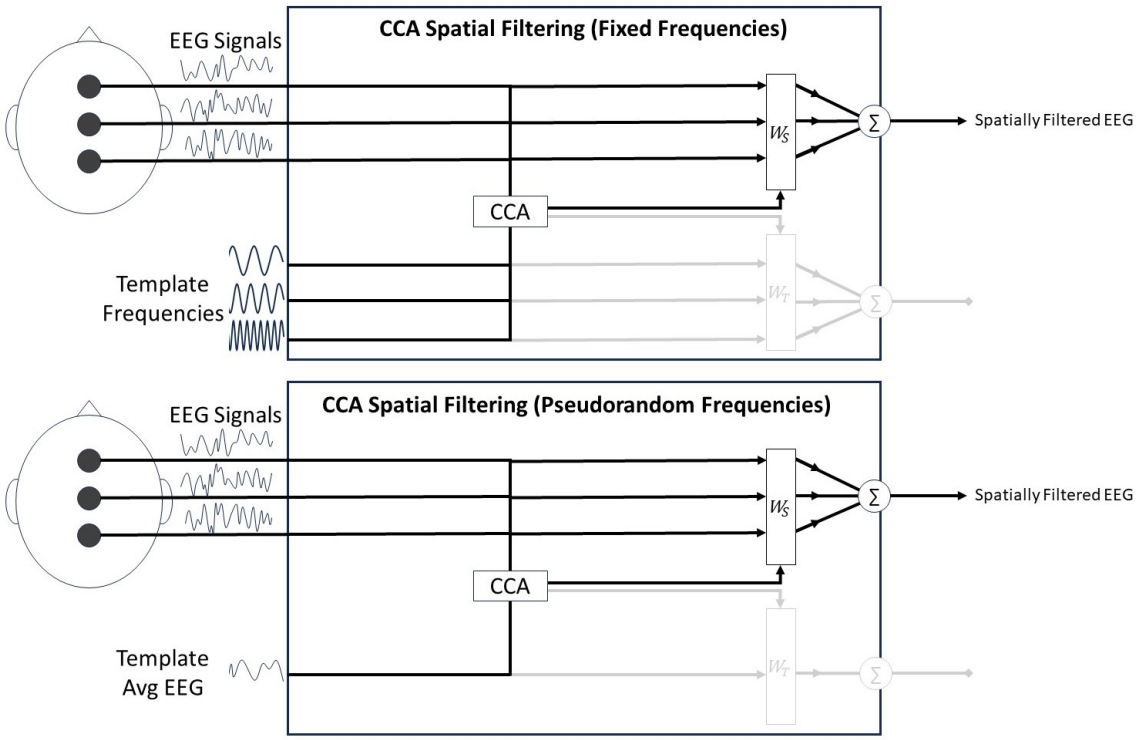


Fig. 17.: Block diagram of CCA spatial filtering process. The top panel of the figure depicts the procedure for fixed-frequency trials, while the bottom panel depicts the procedure for pseudorandom trials. Note CCA produces weights (referred to as canonical coefficients) that when applied to the two sets of data maximally correlate them. In the spatial filtering procedure, only the weights produced for the EEG signals were utilized.

flicker pattern used as the model input and the corresponding EEG signal as the target model output. The first window for each trial began at the start of the trial with a duration of 2 seconds. The corresponding EEG signal was aligned using a delay of 70 ms, as this is the approximate time needed for the sensory information to reach the visual cortex ([4]), also having a duration of 2 seconds. The next window was started 62 ms after the start of the previous window to give an overlap of 93%. These values were chosen as a form of data augmentation due to the large amount of data needed to train models of the type used in this study. To maintain consistency for model training, only full windows were used for each trial, thus data from the ends of the trials were not included. These input-output window pairs were saved to the same file along with a label of the stimulus type for training the models. Additionally, this windowing procedure was utilized for calculating Welch's power spectral density estimate for the signals (see section 8.1).

CHAPTER 7

PRELIMINARY VISION MODEL NETWORK

This chapter discusses the design and implementation of the preliminary visual system model that is trained using the experimental data described in prior chapters. The chapter begins with a detailed meta-analysis of the qualities needed for the network to be successful in the task of producing realistic EEG from a binary flicker pattern. Then the specifics of the selected network architecture are discussed. The chapter concludes with an outline of the training procedure used to generate models of the visual system for each participant.

7.1 Meta Analysis of Desirable Qualities for a Neural Network Model of the Visual System

Building an ANN that is capable of synthesizing accurate EEG based on a representation of the sensory information perceived requires careful conception. Even for VEP responses that have well-understood properties, there are still many obstacles to constructing a network that arises from various aspects of the structure of the data, data preprocessing, and the inherent properties of different ANN architectures. Despite this, ANNs are still one of the most promising tools for modeling how sensory information modulates EEG due to their power to learn relationships within the data. With proper attention given to the challenges of working with this type of system, an accurate model can be developed.

The model developed for this work aims to reconstruct the EEG time series directly from the binary visual stimulus patterns. Several different ANN architec-

tures were considered, and ultimately the Residual Connection Feed Forward System Identification Neural Network (ReCon FFNN) provided strong performance given the evaluation metrics discussed in the subsequent chapter and was thus selected as the model architecture for this preliminary analysis. It should be noted that the selected model is an initial attempt to model the end-to-end behavior of the visual system, and there may be other candidate models that perform just as well or better. The model presented in this thesis is primarily intended to provide an initial proof-of-concept for this modeling scenario. Limitations and other considerations for the modeling and evaluation approaches are discussed in the concluding chapter.

7.1.1 Network Input and Output Structure

The structure and representation of the data within the input and output for an ANN determines how the network must be organized. As previously described (see section 6.2), the model only receives the flicker pattern sequence from the visual stimulus presented to the participant. Thus, the model only considers the temporal information of the stimulus and is not informed of any spatial information. This simplification was deemed reasonable since the foveated stimulus was presented with fixed contrast changes from bordering regions, and the stimulus background was static. Disregarding the spatial information greatly simplifies the modeling process by removing dependencies between the spatial and temporal dynamics, which are more challenging to accurately model. Additionally, this simplification greatly reduces the number of parameters within the network as the entire spatial stimulus is not used as the input and instead, a single vector representing the spatial redundancies of the stimulus can be used. It is envisioned that the proposed temporal model can be extended in the future to include the spatial dynamics of the stimulus.

The flicker pattern of the stimulus is represented as a sequence of binary values

(i.e., 0 and 1), with each value corresponding to one of the states of the stimulus. Additionally, this is how the flicker sequence from each trial was originally coded and recorded (see Section 6.1). To improve the convergence properties of the network based on the sigmoid activation function utilized within the network (discussed in section 7.2), the binary values for white and black were converted to +3 and -3, respectively. This specific ±3 scaling of the input was intentionally selected to have the input operate within the entire unsaturated region of the sigmoid function. This had a desirable effect on the size of the weights within the network by keeping them small and accelerating the learning process.

The flicker patterns for the stimulus can be thought of as periodic square waves (in the case of the m-sequence stimuli, the period is much longer than the fixed frequency stimuli), and these periods are repeated throughout a trial. The data windowing procedure (see section 6.3) results in sets of identical inputs due to the periodic nature of the square waves. This can cause the model to simply learn the average of all the trials with the matching input windows. To mitigate this problem, random noise (normally distributed, mean=0, and variance=0.1) was added to diversify the input characteristics, which is a common approach for adaptive learning [57]. The 0.1 variance of the noise added was empirically found to be the minimum noise to have the intended effect, and as such kept any blurring of the input to a minimum.

Furthermore, it should be noted that this scaled variance is selected in proportion to the ±3 scale of the input. This random noise added to the model input is ultimately representative of the intratrial and intertrial differences between the EEG windows with identical inputs. Additionally, this noise was independent for each input window in a training batch to aid in mitigating overfitting (the residual connections within the network also aided in reducing how much overfitting occurs within the network; see Section 7.1.3). Nonetheless, with such repetitive lagged inputs, overfitting was a

primary concern while designing the network. SSVEP response modeling using the stimulus as the input to the model will always need to address this problem as the stimuli are intentionally repetitive yet produce non-identical responses. Thus, it is advised for future studies to consider the use of alternate visual stimulus patterns, such as cVEP stimuli patterns, to build preliminary models, or carefully implement even more robust ways to address this matter.

7.1.2 Data in Multiple Units

As outlined in section 6.2, the recorded EEG from each participant was spatially filtered using CCA to yield a single channel. This was performed to aid in making the problem more approachable for the ANN model. ANNs are commonly used for tasks for which the dimensionality of the input is greater than that of the output. However, for the present modeling problem, the model input square waves are much simpler (only contains fundamental frequency and harmonics) in comparison to the target model output EEG (dynamic, broadband spectral content). For this reason, the CCA spatial filtering technique was employed to reduce the model's task modeling 8 channels of EEG to only modeling a single channel.

However, this spatial filtering does create a problem that complicates the network's task. The process of spatially filtering the data using the canonical coefficients must be performed on a trial-by-trial basis because the templates for the CCA are trial-specific due to the different stimulus types and random times of the button presses for the fixation task. This means that the data may be scaled differently for each trial, posing a problem for network training. The traditional approach to such scaling issues is to normalize or z-score the data, but this technique does not necessarily solve the problem as the statistics of the stimulus-induced EEG may be non-Gaussian. To help mitigate this issue, the signals were normalized with respect

to the average baseline for each channel of EEG before performing the CCA spatial filter as described in Section 6.1. This normalization to the signal without the stimulus present proved effective for accounting for potential scaling discrepancies across the stimulus types.

7.1.3 Residual (Skip) Connections

Fully connected linear feed-forward networks are well motivated for the type of system identification regression task presented in this study. However, there are several key differences between the vast majority of system identification tasks and the present task, primarily, the relative length of the output to that of the input. Most regression tasks receive inputs with larger dimensions than the output. For example, state-of-the-art techniques for natural language processing utilize a sequence generation approach by selecting a single word from a dictionary with the highest probability of being the next in the output sequence, which allows outputs to be one word rather than an entire body of text. The task of the present model is structured differently as the output is an entirely new sequence consisting of the same number of points as the model input. One of the main inspirations for the proposed preliminary model's design came from prior work on ANNs for image generation.

Image generation tasks often present the model with random noise as the input, and require the model to learn a general set of desired qualities of the output images. Nonetheless, there are some examples of ANNs for image generation that receive an input besides random noise. Isola et al. (2018) [58] constructed an image generation network based on a GAN, which can take simple line art or blocks of colors as inputs to produce photorealistic images of objects and buildings. The task of generating realistic EEG based on sensory input has many similarities with the task of producing images from blocks of colors.

Since VEPs contain the fundamental frequency and harmonics of the stimulus' flicker pattern, the binary sequence given to the model as an input acts in much the same way as the blocks of color in the input image for the Isola et al. Image-to-Image network. The general notion of the desired output is captured, but the input lacks specific contextual details. Much of the strength of the Isola et al. Image-to-Image network, as well as many of the highest-performing networks in other fields, comes from an understated property that they share: the use of residual or "skip" connections.

Residual connections consist of merging the information of the model's input or a previous layer's output with the input to a downstream layer. What this means practically is that the model can perform more drastic transformations on the data flowing within the network as it gets "reminded" of the content of the input or previous outputs. This type of connectivity is extremely important as models add more layers and become deeper. Each layer of an ANN can cause the relationship between its output and the input to become less distinct, which can lead to convergence issues and/or overfitting. Residual connections are how this problem is mitigated, and this approach has greatly contributed to the success of many types of networks. For example, RNNs, U-Nets, and Transformers all utilize residual connections [59, 58, 60]. However, these architectures are not well suited for the regression task of producing EEG from sensory input as presented in this thesis.

The architecture selected for the present visual cortex EEG model is a Residual Connection Feed Forward System Identification Neural Network (ReCon FFNN). Figure 18 shows a block diagram of the ReCon FFNN used in this thesis. The network architecture was in part inspired by the work of Sobreira and Tremmel (2018) [6], as well as Lundby *et al.* (2023) [61]. Sobreira and Tremmel were able to show that a system identification style fully connected neural network was able to accurately

reproduce EEG for VEP presented at a slower rate than those used in this study. Lundby *et al.* were able to improve their system identification model by adding residual connections to the input at each layer.

The ReCon FFNN offers several features targeted at aiding it to produce a meaningful output. An aspect of this architecture that proves extremely valuable in the given modeling situation is the ability to self-learn relationships between time points rather than prescribing them within the architecture itself. While learning the relationship between time points is not unique to the ReCon FFNN, what is unique is the ability to allow residual connections to each time point in each layer to be learned. Most other architectures require residual connections to be made by adding or multiplying the input or previous layer output with the current input due to constraints on layer input sizes. The ReCon FFNN can learn how to weight residual connections by simply concatenating the current layer's input with the original input or previous layer output. This change in how residual connections are made contributes greatly to the success of the architecture. Finally, FFNNs are considered the standard for system identification regression questions, which nicely align with the present modeling scenario.

7.1.4 Composite Loss Function

While iterating on the architecture and training of the network, it was found that it required a very carefully constructed loss function to achieve convergence. Most regression networks utilize point-wise mean absolute error (MAE) or mean square error (MSE) (also respectively known as L1 and L2 loss). These metrics are a logical choice for training many regression networks as they can inform the network how off its output is from the target output in relatively simple terms. However, the highly oscillatory nature of EEG complicates how point-wise MAE and MSE should

be interpreted. If the network output is exactly the same as the target output except for being slightly out of phase, then both metrics would likely produce large errors. For this reason, the MSE between the magnitudes of the Fourier Transform of the network output and target output is calculated and included within the loss function. Furthermore, despite the complications from the oscillatory data, both MAE and MSE have additional strengths that justify their inclusion in the loss function. MAE is commonly considered to be robust against outliers, which allows the model to learn the lower-order statistics of the output more efficiently. On the other hand, MSE causes larger errors to contribute significantly more than small errors to the overall loss. To have the model benefit from both these respective strengths, the point-wise MAE and MSE were both calculated and averaged. This averaged point-wise loss was then empirically scaled before being combined with the MSE of the Fourier magnitudes to control its relative contribution to the loss overall. This empirically developed loss function equation is provided in Equation 7.1:

$$Loss = 0.55 * \frac{MAE(RealEEG, ModelOutput) + MSE(RealEEG, ModelOutput)}{2} + MSE(FreqMagnitudesRealEEG, FreqMagnitudesModelOutput) \quad (7.1)$$

7.2 Vision Model Network Architecture

The selected architecture of the visual cortex EEG model network is relatively straightforward in its structure. The model consists of four layers and uses sigmoid activation functions. The sigmoid function has traditionally been used as an activation function as it mimics biological neuronal behavior [62]. The first layer in the network receives the input consisting of a binary sequence as described in section 7.1.1. The output of this layer is a sequence of equal length. The input to the second

layer uses the output of the first layer concatenated with the original input and outputs a sequence equal in length to the original input. This design repeats through the final (fourth) layer. The output from each layer passes through the sigmoid activation function before continuing to the next layer. For a diagram of the architecture of the network refer to Figure 18.

7.3 Model Training

The network was trained and validated on each participant's data separately. The data for each participant was divided into two independent portions: one consisting of 80% of the total number of window pairs for training the model, and one consisting of the remaining 20% for testing.

The participant-specific models were trained using all pairs of windowed flicker sequences and corresponding lagged EEG signals for a given participant from the training data. Each epoch of training began with a batch of randomly selected window pairs. The batch size was equal to one-fifth of the total training data. The mean of each EEG window within the target outputs for the batch was subtracted from that window to remove any offset. Flicker sequences were scaled and combined with normally distributed random noise as described in Section 7.1.1. These flicker patterns were applied as inputs to the model and the model output was computed and recorded. The model outputs were compared with the desired EEG signal via the loss function (see section 7.1.4) for each window individually. The average of the losses across all windows in an epoch was used to perform the optimization calculation. The Adam optimization algorithm [35] built into PyTorch was used to optimize the model weights based on modifications to the backpropagation algorithm.

After the optimization step, the previous losses were checked to determine if the model's performance had started to plateau. When the model's loss had ceased to

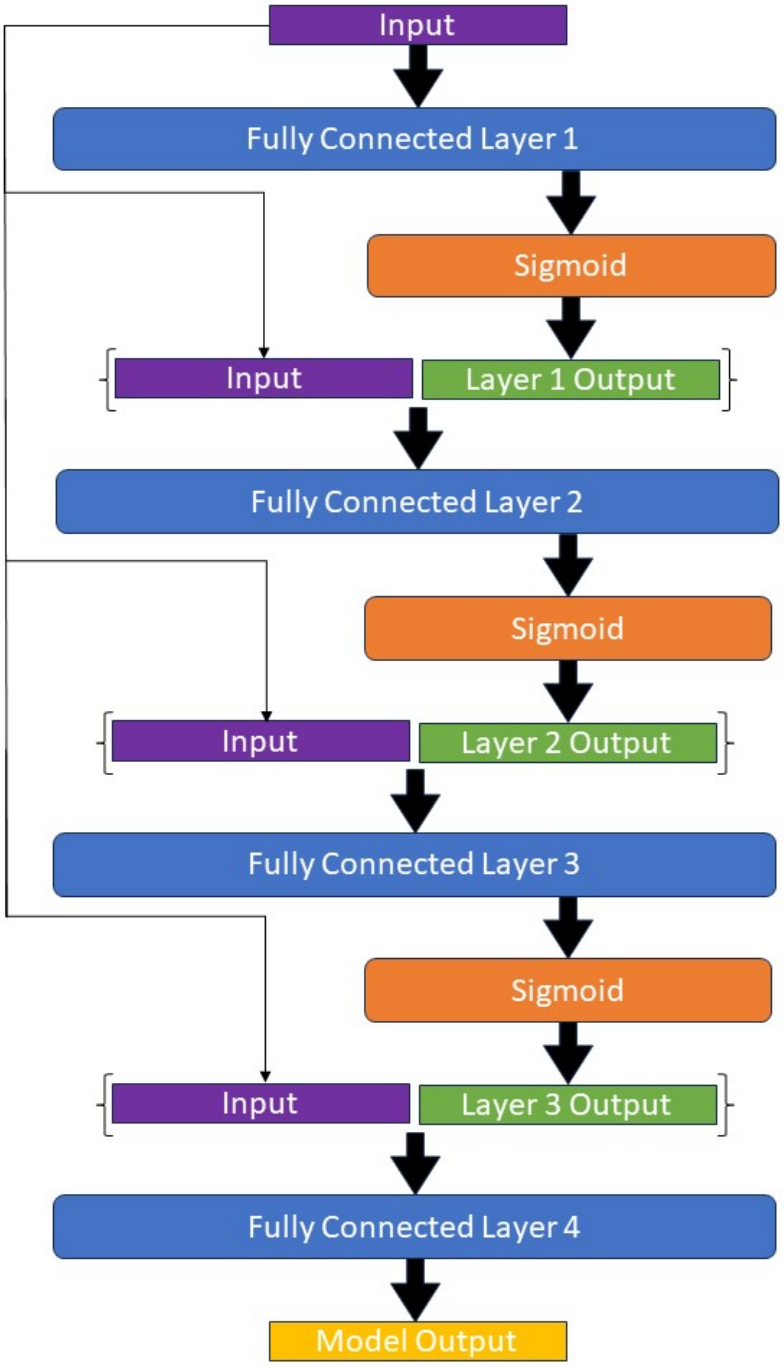


Fig. 18.: ReCon FFNN Model Architecture. This diagram shows the flow of information throughout the network. The brackets indicate concatenation.

Hyperparameter	Value	Description
Training Epochs	2000	The number of times the model would produce outputs for a batch of inputs and then calculate the average loss across all outputs from the batch. The model weights were updated each epoch.
Initial Learning Rate	0.001	The initial learning rate used to scale the adjustments made to the model weights after each loss calculation. This value is adjusted by the learning rate factor.
Learning Rate Factor	0.2	When the loss value of the model ceased to decrease by at least 0.00001 for several epochs equal to the Learning Rate Patience, the learning rate would be multiplied by this value.
Learning Rate Patience	80	The number of epochs that would have to pass before the Learning Rate Factor could be used.
Beta 1	0.7	The Beta 1 parameter for the Adam Optimization function used to find the optimal weights for the model.
Beta 2	0.999	The Beta 2 parameter for the Adam Optimization function used to find the optimal weights for the model.

Table 2.: Model Hyperparameters. This table presents the hyperparameters used with the model presented in this thesis.

decrease by at least a factor of 0.00001 across the last 80 epochs (Learning Rate Patience), the learning rate for the model would be multiplied by a factor of 0.2 (Learning Rate Factor). The hyperparameters for model training can be seen in Table 2. After the model had been trained, the model was evaluated by computing and recording the model output for all windows in the training and testing data, respectively. These outputs were saved with their corresponding inputs and the actual EEG signals for evaluation of the model's performance, as described in the next chapter.

CHAPTER 8

DATA CHARACTERIZATION AND MODEL EVALUATION

This chapter covers the characterization of the EEG responses from the experiment and the evaluation of the preliminary model for estimating the responses. The power spectra are compared across stimulus types and the challenges of properly evaluating models that generate simulated EEG are presented. Three types of model evaluation are considered: (1) performance based on specific loss functions, (2) discrimination of the model output compared to actual EEG, and (3) comparison of the autocorrelations of the model output and actual EEG.

8.1 EEG Response Characterization

To characterize the data recorded for each of the visual stimulus types, Welch's power spectral density estimate [63] of the spatially filtered EEG was calculated for each participant and trial. The windowing procedure for this calculation was the same as that for the model inputs (i.e., 2 second, or 256-point, windows with 93% overlap) with FFTs zeropadded to a length of 512 points. The responses across all trials of the same stimulus for all participants, along with the average of all responses for the respective stimulus, are displayed in Figure 19.

The EEG responses for the fixed-frequency trials are as expected, with consistent steady-state responses manifesting as spectral peaks at the frequency of the stimulus and its harmonics. Harmonic peaks are observed through the fourth harmonic for all fixed frequency responses, and even the sixth harmonic in certain cases. The m-sequence stimuli, by design, generate both a broadband and steady-state response.

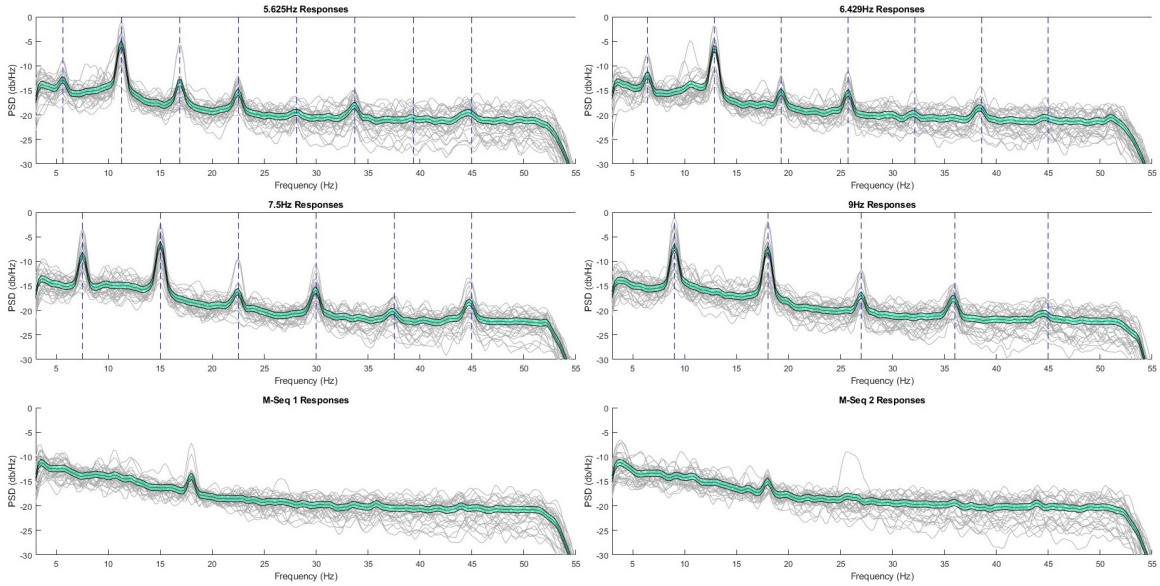


Fig. 19.: Welch's Power Spectral Density Estimates for all EEG responses to each stimulus. The gray lines represent each trial and the green shaded regions show the 95% confidence intervals. The vertical dashed lines indicate the locations of the harmonics of the stimuli for the fixed-frequency trials.

This elevated broadband power is observed in frequencies below 18 Hz. The value of 18 Hz is significant as that is the second harmonic of the highest frequency of the m-sequence stimuli based on the bit intervals. Thus, participants produced higher activity across the band of frequencies associated with the fundamental frequencies within the m-sequence stimuli and their second harmonics. Note that this increase in power was spread across all of the frequencies in this band, resulting in no prominent spectral peaks except for at exactly 18 Hz. This peak centered at 18 Hz, which is not as prominent as the peaks in the fixed-frequency responses, is presumed to be due to the average response of the stimulus alternating at the prescribed bit interval.

8.2 Evaluation Techniques for Vision Modeling Network

Because this specific type of stimulus-to-EEG modeling has not been previously explored in the literature, conventional evaluation metrics do not conveniently apply.

Therefore, the proposed evaluation borrows techniques established to evaluate synthetic EEG with forecasting ANNs (see section 9.2), and the re-purposing of other existing techniques to evaluate networks in the context of time series regression. Overall, the guiding principle for the present model evaluation is to quantify the similarity of the relevant qualities of the model output to the associated actual EEG signal.

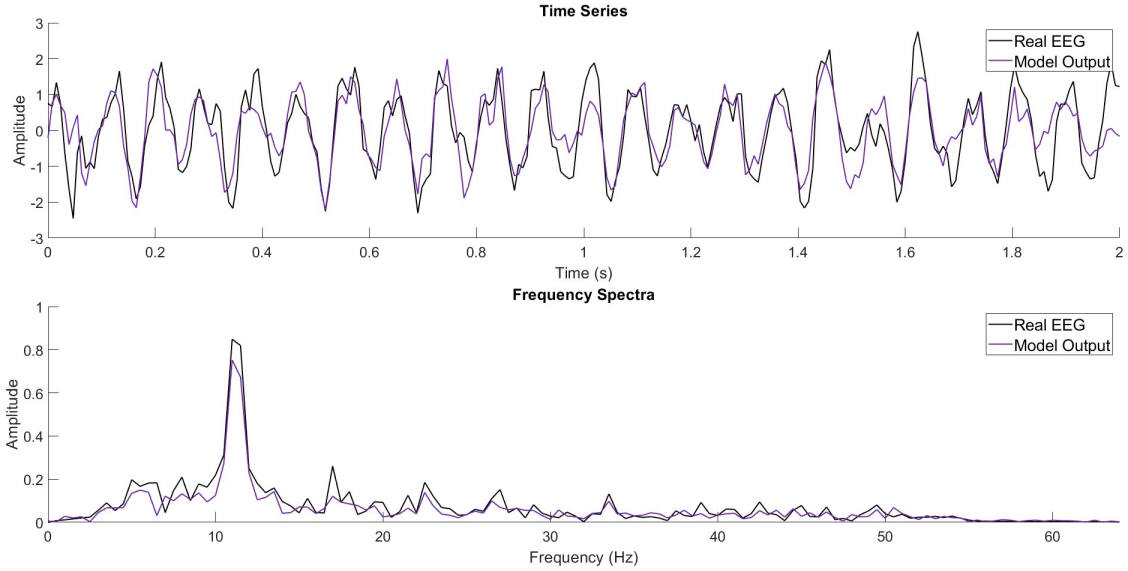
Example model outputs are presented in Figure 20 and Figure 21 for Participants 1 and 4, respectively. While the time series and associated frequency magnitude spectra for the recorded EEG and model output visually align with the training data, the recorded EEG and model output for the testing data largely match in terms of magnitude spectra but appear to have phase differences from inspection of the time series. This is generally an acceptable result for such cyclic stimuli and, thus, non-traditional methods of quantifying such properties are proposed.

After model training was completed, all training and testing data inputs were passed through the model to produce final outputs for evaluation. These outputs were saved with the corresponding inputs and the actual EEG from the same input sequence. These model outputs are evaluated using several distinct approaches, which are subsequently described.

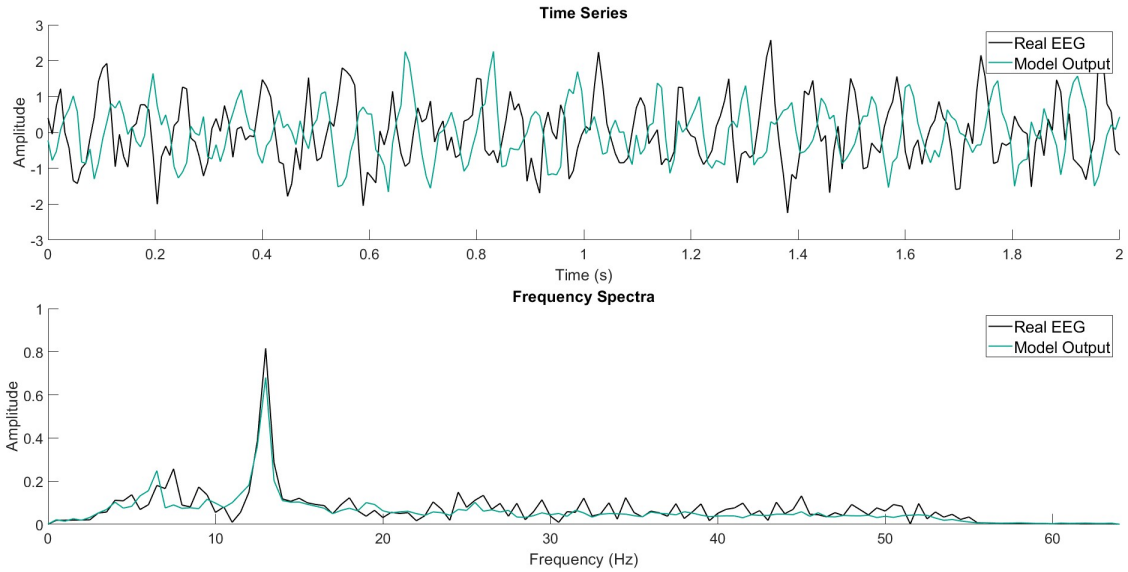
8.3 Measurements of Loss for Vision Model Neural Network

8.3.1 Results

Several loss terms that compare the actual EEG and model output were calculated for each participant over the training and testing data, respectively. Some of these terms directly contribute to the loss function used for network optimization, while others are included as alternate metrics for reference. The specific loss terms used are listed in Table 3. Figure 22 shows the averages of these loss terms across

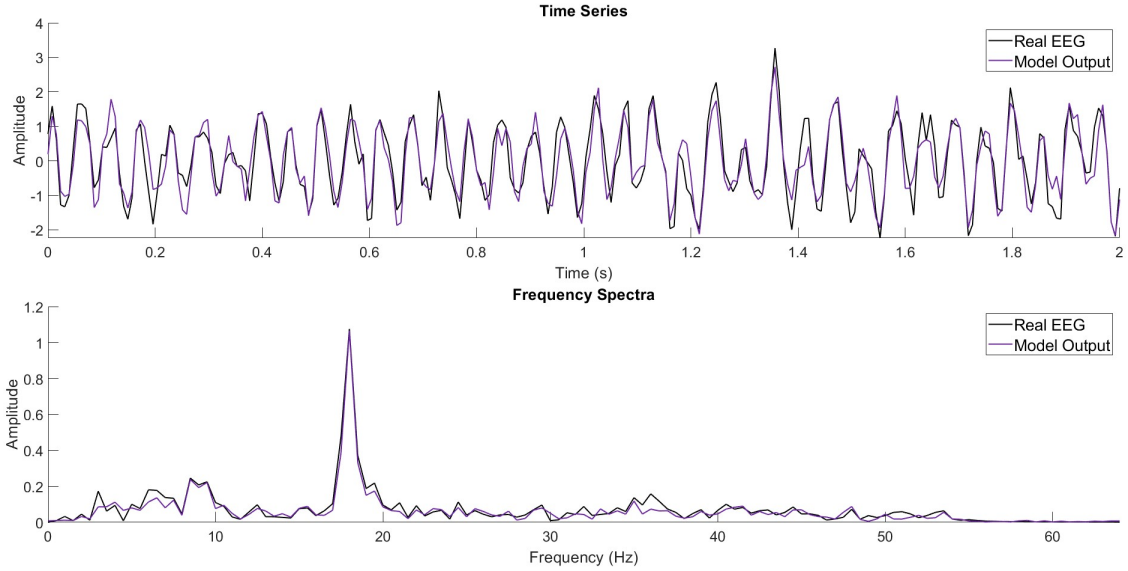


(a) Example from Training Data (5.625 Hz Trial).

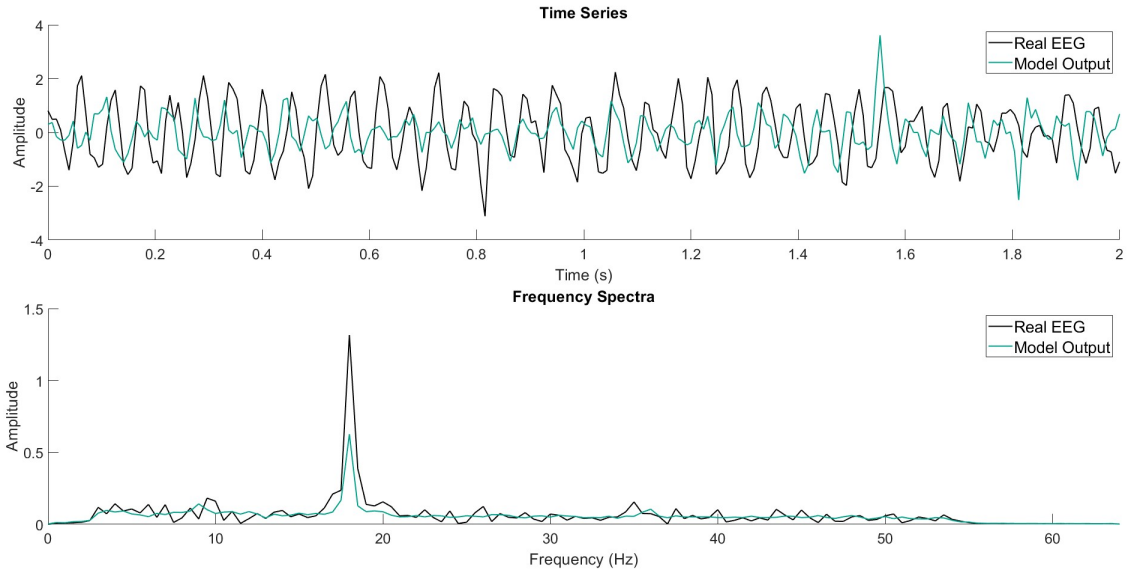


(b) Example from Testing Data (6.429 Hz Trial).

Fig. 20.: Example Model Outputs and corresponding Actual EEG from Participant 1. Corresponding frequency magnitude spectra appear beneath the time series.



(a) Example from Training Data (9 Hz Trial).



(b) Example from Testing Data (9 Hz Trial).

Fig. 21.: Example Model Outputs and corresponding Actual EEG from Participant 4. Corresponding frequency magnitude spectra appear beneath the time series.

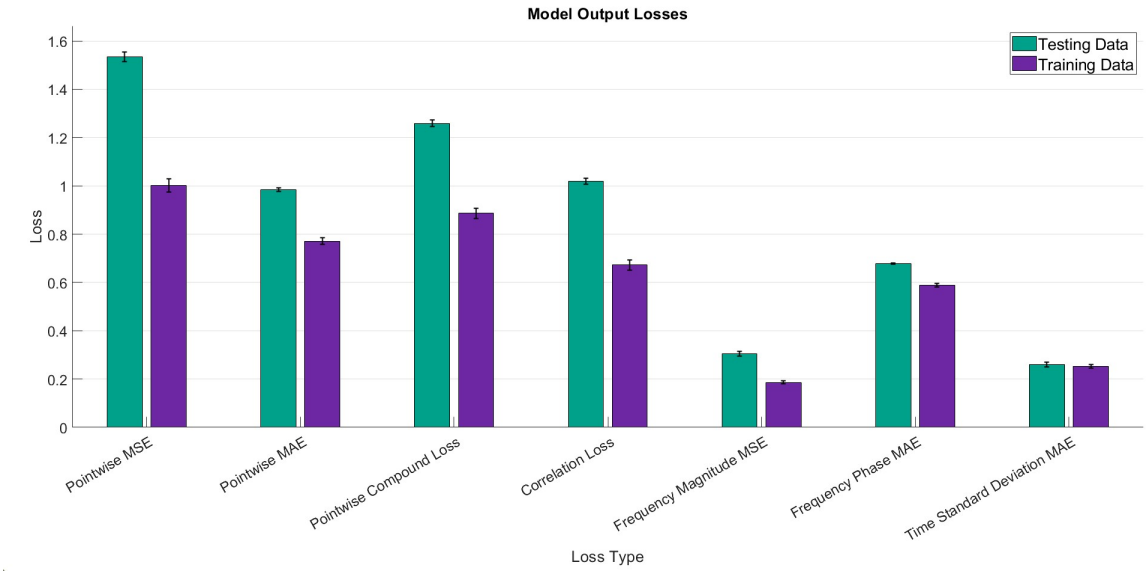


Fig. 22.: Average Losses across participants for both testing and training data. The mean of each of the loss metrics (described in Table 3).

participants.

8.3.2 Discussion

While loss terms cannot be directly compared due to each having unique units, what can be inferred from the losses are the characteristics of the EEG data the model was able to learn and how performance shifts between the training and testing datasets. As expected in model training, the loss is generally higher for the testing data compared to training - indicating that there is some degree of overfitting. However, this is a typical and reasonable behavior of ANN training.

The Time Standard Deviation MAE is relatively low and constant across the training and testing datasets, which indicates that the model was able to learn to produce signals with similar distributions of amplitudes as the actual EEG data. Similarly, the Frequency Magnitude MSE is relatively low for both the training and testing data. This means the model was fairly apt at capturing the magnitude spectral

Loss	Description
Pointwise Time MSE	The Mean Square Error between each point in the recorded EEG and the model output for the same input sequence.
Pointwise Time MAE	The Mean Absolute Error between each point in the recorded EEG and the model output for the same input sequence.
Pointwise Time Compound Loss	The average of the Mean Absolute Error and Mean Square Error between each point in the recorded EEG and the model output for the same input sequence.
Correlation Loss	One minus the correlation between the recorded EEG and the model output for the same input sequence.
Frequency Magnitude MSE	The Mean Square Error between the frequency magnitude coefficients of the recorded EEG and the model output for the same input sequence.
Frequency Phase MAE	The Mean Absolute Error between the frequency phase coefficients of the recorded EEG and the model output for the same input sequence.
Time Standard Deviation MAE	The Mean Absolute Error between each standard deviation of the recorded EEG and the model output for the same input sequence.

Table 3.: A description of the losses used to calculate the differences between the model output and the recorded. These losses computed for the network are shown in Figure 22.

content of the actual EEG.

The Correlation Loss (Note the correlation loss definition in Table 3) and Frequency Phase MAE suggest that the phase information of the actual EEG was not effectively captured by the model. Both metrics are related to the temporal alignment of the model output and actual EEG. These phase metrics were not directly included in the composite loss function for optimization due to convergence issues introduced by creating a more difficult gradient to estimate during optimization. Therefore, while the selected composite loss function facilitated the reproduction of the magnitude spectra, a different model and/or loss function is likely necessary to reliably capture the phase.

8.4 Classification of Actual EEG and Model Output

8.4.1 Results

A commonly used method to evaluate synthetic EEG generation is to build a classifier based on actual EEG to determine if the correct class label (e.g., stimulus type) can be predicted for synthetic EEG. The type and organization of these classifiers can vary drastically depending on the specific application. Most synthetic EEG studies employ ANN classifiers, or traditional SSVEP classifiers such as a CCA (see section 6.2). However, for this study, a classifier architecture with a straightforward design was selected for better interpretability.

Two different participant-specific classification scenarios were examined using k-nearest neighbor (KNN) classifiers [64]. The first scenario used actual EEG time series windows (including both the training and testing data) as the reference data. Using $k=3$ (empirically selected) with Pearson's correlation as the similarity metric, classification of the associated stimulus type was performed on the modeled EEG

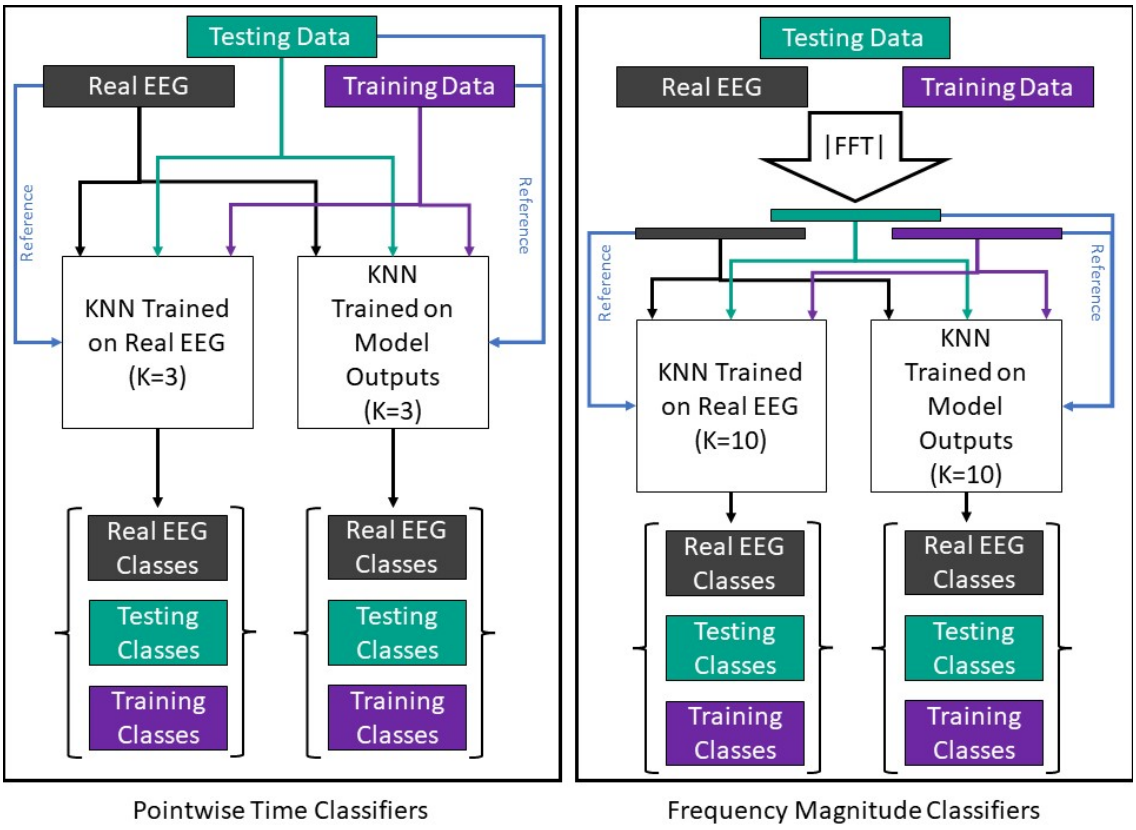


Fig. 23.: Diagrams of the KNN Classifiers. Blue arrows indicate the data is being used as the reference for the classifier. |FFT| is shorthand indicating that a Fast Fourier Transform was performed and then the absolute value was taken to produce the frequency magnitude coefficients.

corresponding to the training data partition, testing data partition, and actual EEG, respectively. This process was repeated by using the modeled EEG as the reference, to compare how the actual EEG would classify in this case [37]. The results of this comparison are shown in Figure 24.

The second classification scenario was conceptually identical to the time series classifiers except the magnitude spectra of the respective time series windows were used and a value of k=*10* was empirically selected. Block diagrams of the two classification scenarios are shown in Figure 23. Results for the second classification scenario are presented in Figure 25.

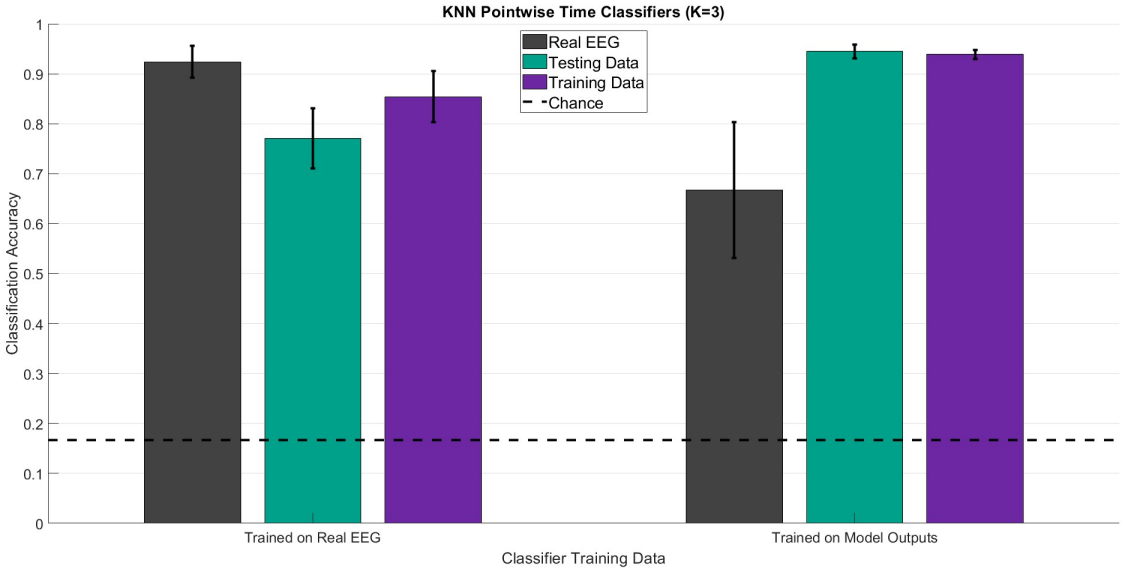


Fig. 24.: Average Accuracy of KNN Pointwise Time Classifiers (k=3) across all participants. One classifier was built using the recorded EEG from a participant as the reference data (left bars), while the other used the model outputs as the reference data (right bars). The classifiers assigned the predicted stimulus type according to the reference data. Self-classification results are displayed for reference.

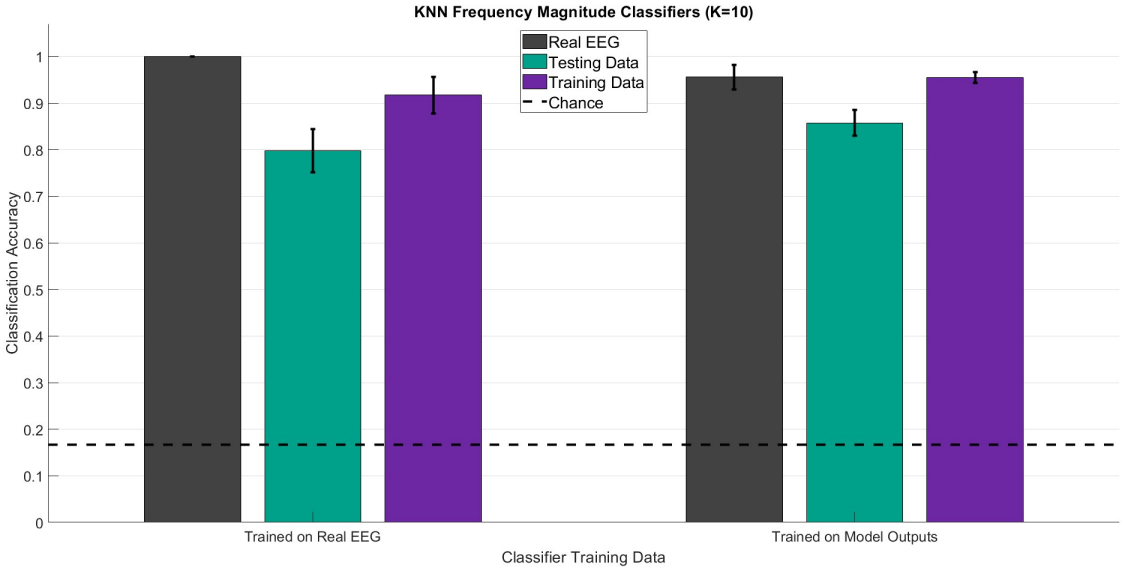
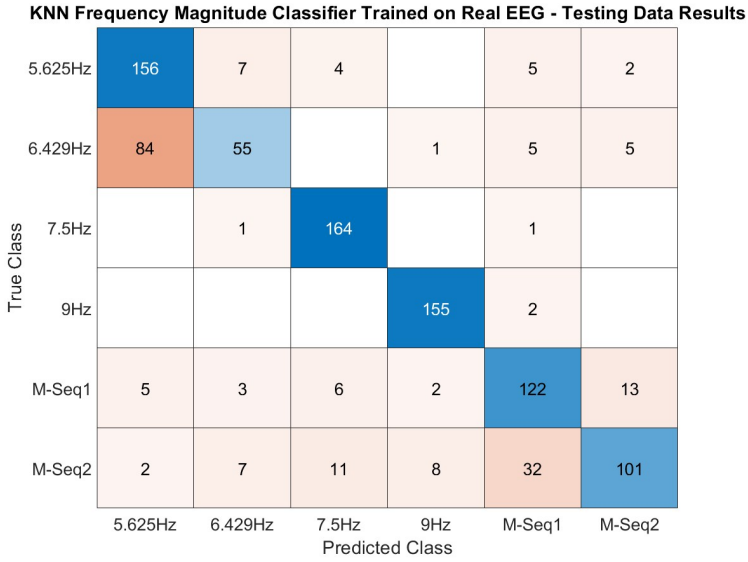
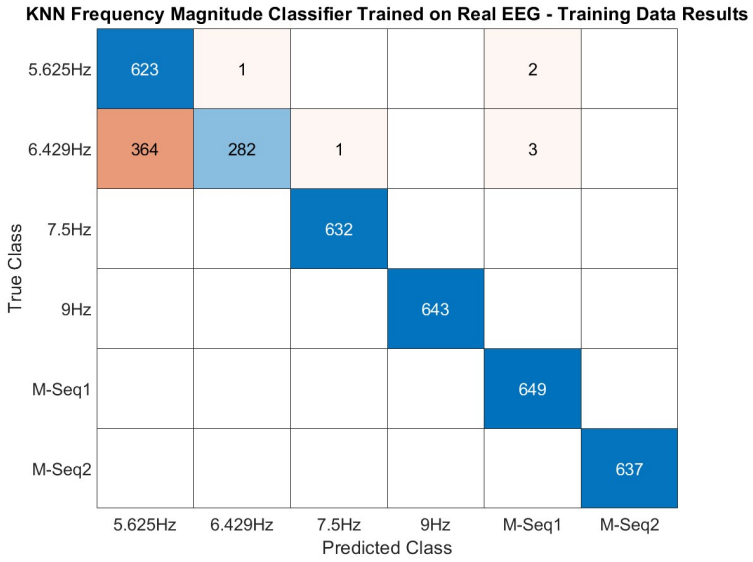


Fig. 25.: Average Accuracy of KNN Frequency Magnitude Classifiers (k=10) across all participants. One classifier was built using the recorded EEG from a participant as the reference data (left bars), while the other used the model outputs as the reference data (right bars). The classifiers assigned the predicted stimulus type according to the reference data. Self-classification results are displayed for reference.



(a) Example Confusion Matrix for Testing Data.



(b) Example Confusion Matrix for Training Data.

Fig. 26.: Example Confusion Matrices for the KNN Frequency Magnitude Classifier Trained on Actual EEG from Participant 6. These confusion matrices are representative of the behavior seen throughout many of the classifiers across participants. This includes misclassification between the 5.625 Hz and 6.429 Hz classes and separately the MSeq1 and MSeq2 classes.

8.4.2 Discussion

The classifiers built to assess the similarity of the actual EEG and model output were relatively successful across all scenarios. The KNN Pointwise Time Classifier Trained on Actual EEG obtained relatively high average classifications on both the training and testing datasets (85% and 77% respectively), indicating that the model was able to produce time series similar to the actual EEG used to train it. These results are also strengthened by the notion that the maximum accuracy of the classifier (92%) is within the same range as the classification accuracy of the reference data used to build the classifier.

The KNN Pointwise Time Classifier Trained on Model Outputs has the lowest performance of all the classifiers as it is only able to accurately predict the class of the actual EEG 66% of the time on average and exhibits relatively high variability across participants (i.e., 42% to 90% accuracy). This is to be expected since the model was trained using the synthetic EEG, which is more likely to deviate from the characteristics of the actual EEG due to the nature of the modeling.

The example confusion matrices in Figure 26 show the same misclassification pattern observed for both Pointwise Time Classifiers. An explanation for the M-Seq1 and M-Seq2 classes being mistaken is that they are the same flicker pattern but reversed. This means that the frequency content of both stimuli is identical, and accordingly, the resulting EEG response for both conditions is expected to be very similar. The use of two distinct m-sequences would likely alleviate this issue. Some fixed-frequency trials were misclassified as an m-sequence trial. This is likely because the pseudorandom stimuli are wideband and include prominent frequencies that overlap with the fixed-frequency stimuli. The misclassification between the 5.625 Hz and 6.429 Hz classes is most likely due to spectral spreading and the spectral peaks

being within 1 Hz of each other. As correlation was used as the distance metric, the higher correlations in responses to these stimuli combined with noise could have led to the classifiers predicting incorrectly. Regardless, the KNN Pointwise Time Classifier Trained on Model Outputs still achieved a classification accuracy much higher than chance, indicating the model outputs and actual EEG were at least broadly similar.

The Frequency Magnitude Classifiers were superior in all regards compared to the Pointwise Time Classifiers. The KNN Frequency Magnitude Classifier Trained on Actual EEG was able to achieve 80% classification accuracy on the testing data model outputs and 91% classification accuracy on the training data model outputs. Furthermore, the KNN Frequency Magnitude Classifier Trained on Model Outputs was able to achieve a 95% accuracy on the actual EEG data. These results support the notion that the model was able to produce data with very similar spectral content as the actual EEG.

8.5 Correlation between Autocorrelations (ACC) of Actual EEG and Model Output Pairs

8.5.1 Results

For many VEP BCI applications, key features for classification are derived from the frequency magnitude response rather than the phase response. Furthermore, EEG responses can be variable, even for repetitive stimuli. Therefore, computing the autocorrelation function of the EEG time series can be a valuable representation that captures the magnitude information while suppressing uncorrelated noise. The autocorrelation of a signal effectively computes the correlation of a signal with itself at different time lags. This is useful for elucidating repeating patterns in a signal such as major frequency components in a periodic signal. Examples of autocorrelation

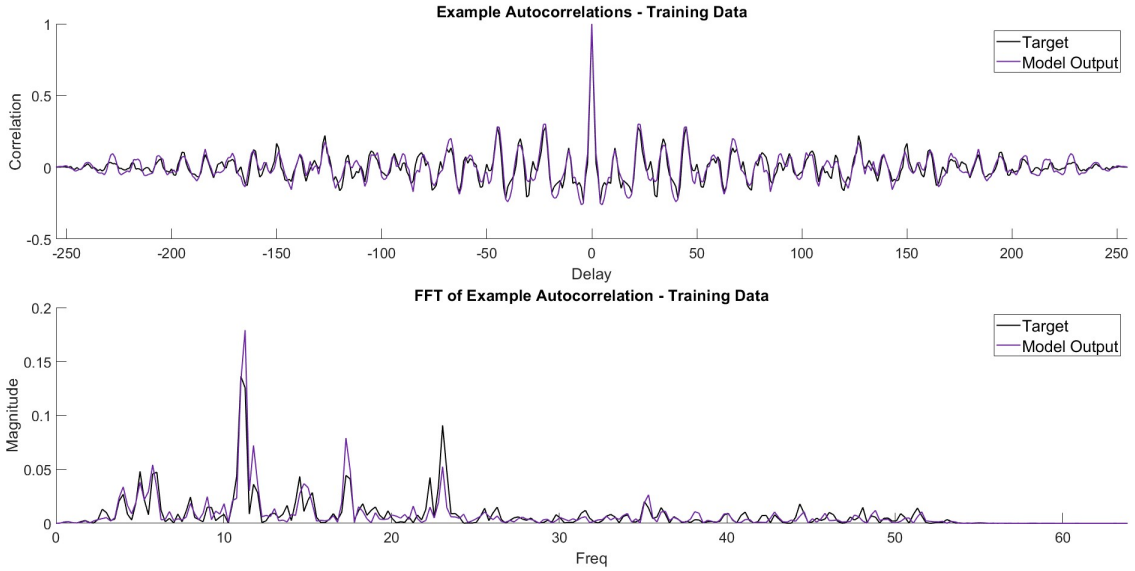


Fig. 27.: An example of the autocorrelation function of a successful model output (testing data) compared to the corresponding actual EEG for Participant 13. The Fourier Transform of both autocorrelation functions is shown below.

functions for the model outputs and the actual EEG are presented in Figure 27 and Figure 28, respectively. Viewing the autocorrelations from the frequency domain is another way to assess shared frequency characteristics.

Computing Pearson's correlation between two autocorrelation functions (ACC), can be used as a measure of the similarity of the major frequency components of the signals. This process was performed for both the testing and training model outputs and their respective actual EEG window. Additionally, a 2-second segment of each participant's average baseline data was collected, and the autocorrelation of this segment was calculated to serve as a reference in the absence of stimulation. The ACC between this segment and each window from the testing data and training data was computed. The bar graphs of the calculated ACCs are presented in Figure 29 including the ACCs with respect to the baseline for reference.

Fig. 28.: An example of the autocorrelation function of a less accurate model output (testing data) compared to the corresponding actual EEG for Participant 13. The Fourier Transform of both autocorrelation functions is shown below.

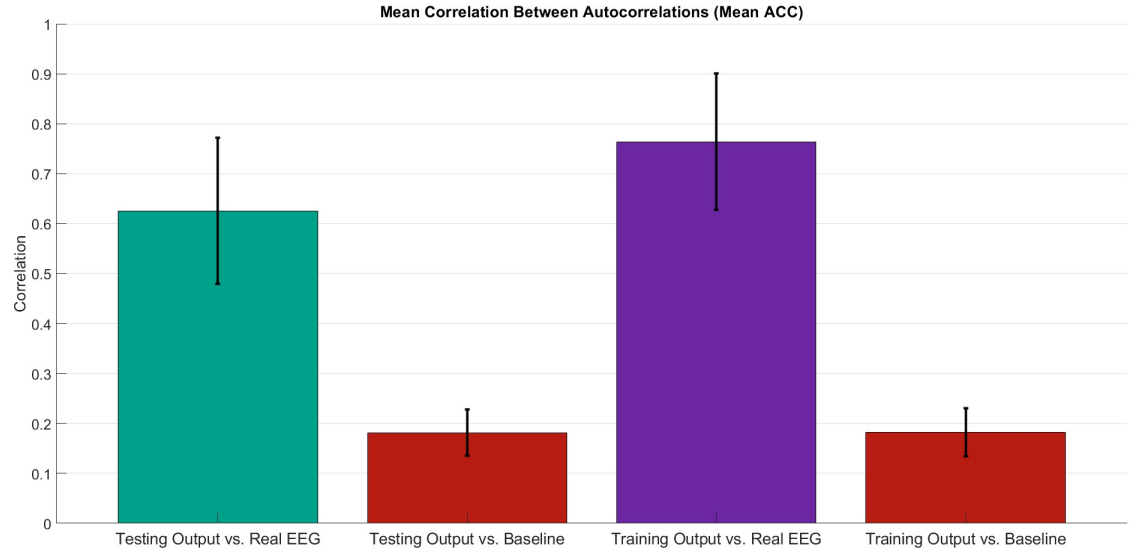


Fig. 29.: Mean ACC of Actual EEG and Model Outputs for both Training and Testing Data. The autocorrelation function of all model outputs and all segments of actual EEG were calculated, respectively. The correlation between the autocorrelation functions of corresponding input-output pairs was computed and averaged within and across participants. This same procedure was performed using an average of the baseline data from the participant in place of the actual EEG.

8.5.2 Discussion

The ACCs of the actual EEG and model outputs proved to be relatively high. Both the testing and training data had an average correlation above 0.6 with standard deviations slightly less than 0.15. This relatively high mean correlation is another indication that the model was able to successfully generate outputs with the same characteristics as the actual EEG by capturing the same harmonic frequencies in its outputs. However, the resulting ACCs are highly variable. A likely explanation for this is the presence of the m-sequence trials within the datasets. While the m-sequences are periodic signals, their periods are longer than an individual data window used in the analysis, they are broadband in nature and can produce transient responses from portions of the stimulus occupying frequencies less than 4 Hz. Therefore, it is expected for the responses to the m-sequence stimuli to be more variable than those of the fixed frequency stimuli. Despite the impact of the m-sequence trial data, the average correlation remained relatively high for all participants.

Inspecting the ACCs between the testing and training data compared to the average baseline, the baseline was relatively weakly correlated with the average baseline. This baseline measure captures the natural autocorrelation of EEG due to ubiquitous rhythmic activity and further highlights that the proposed models are not simply reproducing this background EEG activity.

CHAPTER 9

CONCLUSION

The main objectives of this thesis were to produce a novel data set for end-to-end modeling of the visual system, and then to develop a preliminary end-to-end computational model of the visual system from the visual stimulus light intensity waveform to the scalp EEG waveform using an artificial neural network. This chapter concludes the thesis with the main contributions, limitations, and future directions of this research.

9.1 Main Contribution

A novel experiment was designed and conducted on 15 participants to generate EEG data related to both fixed frequency and pseudorandom stimulus sequences, for use in the development of end-to-end system identification models of the visual system. The stimuli and experimental protocol were systematically designed to produce concordant EEG responses that broadly activate different regions of the EEG frequency spectrum. As such, the resulting input and output waveforms are intended to excite a variety of system modes for creating robust end-to-end models.

Using this data, a preliminary computational model of visual processing from visual input to the visual cortex was designed and implemented. A novel ANN with learnable residual connections to the input (ReCon FFNN) was developed and participant-specific networks were trained using the binary flicker patterns as the input and the corresponding EEG as the target output. Because this modeling scenario has not been previously reported in the literature, several novel metrics are proposed

to quantify the difference between the model output and the actual EEG. These metrics showed that the network was able to learn many of the key characteristics of the data including the frequency magnitude spectra and variability of the data.

The visual models were also evaluated using a novel implementation of a k-nearest neighbor classifier scheme to identify the type of stimulus used to generate the modeled and actual EEG data. The classifiers yielded accuracies that were significantly above chance levels, suggesting that the models captured the meaningful shared characteristics in the time domain. The equivalent classifiers using the frequency magnitude spectra yielded comparatively higher accuracies, indicating the network is more robust for capturing the relevant frequency content. The network was further evaluated by comparing the autocorrelation functions of model outputs and actual EEG windows. This process produced moderate to high correlations between the model inputs and corresponding outputs, which further indicates that the network was able to reliably capture the harmonic characteristics of the data. Overall, these results demonstrate that the ReCon FFNN can be a viable model of the pathway from the retina to scalp EEG.

9.2 Limitations and Future Work

There are several possible ways to extend this research based on the limitations and promising results of the present work. A portion of participants with corrected vision opted to not wear glasses during the experiment. While all participants stated that they could clearly see the stimulus and flicker patterns, this may have affected their EEG responses. However, since this work was focused on a temporal model, any spatial blurring due to the lack of corrected vision is unlikely to alter the present results. Future models that incorporate spatial information may need to account for the participants with corrected vision. Related, due to hardware limitations, the

present study did not use an eye tracker to confirm the participants' fixation on the stimulus. Instead, a fixation task was implemented, which is not as precise as eye-tracking data for confirming eye position and segmenting the corresponding EEG data.

Another issue is how to develop neural networks capable of using single time-series inputs to produce outputs with higher dimensionality, such as multichannel EEG. Since the input used with the ANN presented in this thesis had the spatial information collapsed into a binary representation, the network had to learn a dynamically varying EEG output from a simple binary sequence input, particularly for the fixed frequencies. Most commonplace ANNs typically have access to rich information at the input for representing the output. For example, classifiers need only produce a limited number of classes at the output, forecasting networks produce a single output value corresponding to multiple lagged inputs, and transformers and other sequence-to-sequence models rely on dictionaries to pull from to produce outputs. Additionally, the inputs of the model only included temporal information, and since the common stimuli for VEP experiments are fixed-frequency visual stimuli, this led to a paucity of information for generating dynamic output representations. The m-sequences were included to help address this issue and, in hindsight, it would have been beneficial to include additional, more varied m-sequence or other random stimuli. Overall, this work proposes a preliminary model architecture and loss function, for which aspects of both were parameterized empirically. It is prudent to explore other potential architectures and loss functions, with more objective and tractable parameterizations.

A limitation of the model and other generative ANN models, such as synthetic EEG networks or the model designed in this thesis, is the lack of established approaches to evaluate their performance. As stated in Habashi et al. 2023: "To date,

there is no direct metric that could be considered a real assessment of the generated EEG signals' quality and directly relate it to the performance of the model" [14]. This issue prevents definitive ways of demonstrating that models aimed at generating EEG are producing accurate outputs. Furthermore, this problem is a product of a gap in knowledge in both the fields of computational modeling and neuroscience. Currently, there exists no established set of characteristics that the model output must share with recorded EEG. This problem even extends to other neurophysiological recording techniques as well, including those with much higher signal-to-noise ratios than EEG. The field of quantitative neuroscience must answer the fundamental question of what metrics define a signal from a given neurophysiological recording technique. Additionally, the field of computational modeling must develop new tools to evaluate models along these metrics. Evaluating models typically relies on distance-based metrics, which require normalized or otherwise modified data, but it is more desirable for models to work in natural units to make them more practical.

Perhaps one of the greatest issues plaguing black box models is the inability to link their internal functions to the actual physical process they are modeling, i.e., model interpretability. Specifically for physiological ANN models, this issue proves to be critical for moving the field from a position of potentially useful tools into a position of producing very strong alternatives to *in vivo* models. A potential avenue of inquiry to address this issue is to take a more biophysical approach to the organization and structure of the ANN models. Ideally, this would lead to the models increasingly functioning in the same fashion as the actual physiological system they intend to represent. This includes the incorporation of the spatio-temporal dynamics of the stimulus and neural processing. Along these lines, another step in the evaluation of future physiological ANN models should be to investigate layer dynamics and link them to the function of specific physiological structures.

In conclusion, this work is intended to facilitate and inspire end-to-end models of the visual system. Models capable of accurately capturing the processing of the visual system from retinal input to the electrical activity of the visual cortex. Such research offers great potential for understanding the visual system and stimulus optimization for BCIs. It is also envisioned that similar approaches can be used to study and model other sensory pathways, and even sensory integration, to build a more complete picture of sensory information processing in humans.

Appendix A

ABBREVIATIONS

ACC	AutoCorrelation Correlation or Correlation between Auto-correlations
ALS	Amyotrophic Lateral Sclerosis
ANN	Artificial Neural Network
BCI	Brain-Computer Interface
CCA	Canonical Correlation Analysis
CNN	Convolutional Neural Network
cVEP	Code-Modulated Visual Evoked Potential
ECoG	Electrocorticography
ERP	Error Related Potential
EEG	Electroencephalography
fMRI	functional Magnetic Resonance Imaging
fNIR	functional Near InfraRed
FFN	Feed Forward fully connected linear Network
FFT	Fast Fourier Transform
FRNN	Fully Recurrent Neural Network
GAN	Generative Adversarial Network
GRU	Gated Recurrent Unit network
LGN	Lateral Geniculate Nucleus
LSTM RNN	Long Short-Term Memory Recurrent Neural Network
MAE	Mean Absolute Error

MEG	Magnetoencephalography
MSE	Mean Square Error
ReCon FFNN	Residual Connection Feed Forward system identification Neural Network
RNN	Recurrnet Neural Network
SCP	Slow Cortical Potentials
sEEG	Stereotactic Electroencephalography
SSVEP	Steady State Visual Evoked Potential
TVEP	Transient Visual Evoked Potential
VEP	Visual Evoked Potential

Appendix B

ADDITIONAL DATA TABLES

Remainder of page left intentionally blank.

Testing Data Losses							
Participant	Mean Pointwise Time MSE	Mean Pointwise Time MAE	Mean Pointwise Time Compound Loss	Mean Time Standard Deviation MAE	Mean Correlation Loss	Mean Frequency Magnitude MSE	Mean Frequency Phase MAE
1	1.5051	0.9799	1.2425	0.2685	1.0038	0.2846	0.6789
2	1.5408	0.9874	1.2641	0.2592	1.0132	0.3077	0.6774
3	1.5694	0.9960	1.2827	0.2627	1.0344	0.3062	0.6806
4	1.5307	0.9837	1.2572	0.2647	1.0141	0.3094	0.6766
5	1.5472	0.9891	1.2681	0.2508	1.0261	0.2996	0.6787
6	1.5343	0.9818	1.2581	0.2599	1.0315	0.2998	0.6798
7	1.5287	0.9822	1.2555	0.2604	1.0175	0.3125	0.6778
8	1.5757	0.9998	1.2878	0.2755	1.0461	0.2882	0.6843
9	1.5219	0.9732	1.2476	0.2674	1.0018	0.3077	0.6755
10	1.5497	0.9866	1.2682	0.2471	1.0149	0.3167	0.6798
11	1.5203	0.9788	1.2495	0.2364	1.0156	0.3064	0.6770
12	1.5216	0.9766	1.2491	0.2672	1.0059	0.3251	0.6761
13	1.5321	0.9839	1.2580	0.2509	1.0163	0.3076	0.6775
14	1.5157	0.9815	1.2486	0.2578	1.0189	0.3039	0.6802
15	1.5131	0.9770	1.2450	0.2727	1.0349	0.3044	0.6801
Average	1.5338	0.9838	1.2588	0.2601	1.0197	0.3053	0.6787

Table 5.: Testing Data Losses. Several different ways to measure the difference between the actual EEG and model outputs for only the testing data.

Training Data Losses							
Participant	Mean Pointwise Time MSE	Mean Pointwise Time MAE	Mean Pointwise Time Compound Loss	Mean Time Standard Deviation MAE	Mean Correlation Loss	Mean Frequency Magnitude MSE	Mean Frequency Phase MAE
1	0.9945	0.7708	0.8826	0.2481	0.6654	0.1763	0.6020
2	1.0030	0.7716	0.8873	0.2517	0.6641	0.1860	0.5868
3	1.0085	0.7735	0.8910	0.2522	0.6675	0.1813	0.5918
4	0.9729	0.7601	0.8665	0.2477	0.6598	0.1797	0.5860
5	0.9993	0.7677	0.8835	0.2478	0.6606	0.1790	0.5887
6	1.0045	0.7717	0.8881	0.2538	0.6772	0.1871	0.5882
7	1.0165	0.7799	0.8982	0.2569	0.6836	0.1920	0.5906
8	1.0073	0.7749	0.8911	0.2566	0.6767	0.1856	0.5981
9	0.9835	0.7561	0.8713	0.2558	0.6556	0.1833	0.5941
10	0.9627	0.7501	0.8564	0.2400	0.6510	0.1853	0.5784
11	0.9869	0.7621	0.8745	0.2450	0.6667	0.1881	0.5751
12	0.9890	0.7642	0.8766	0.2574	0.6646	0.1985	0.5863
13	0.9962	0.7685	0.8824	0.2492	0.6673	0.1844	0.5868
14	1.0168	0.7814	0.8991	0.2544	0.6829	0.1878	0.5898
15	1.0865	0.8093	0.9479	0.2732	0.7377	0.2034	0.6002
Average	1.0019	0.7708	0.8864	0.2527	0.6720	0.1865	0.5895

Table 6.: Training Data Losses. Several different ways to measure the difference between the actual EEG and model outputs for only the training data.

KNN Pointwise Time Classifier Trained on Actual EEG (K=3)			
Participant	Actual EEG*	Testing Data	Training Data
1	0.9631	0.8092	0.8924
2	0.9255	0.8156	0.8845
3	0.9343	0.8154	0.9156
4	0.9066	0.7393	0.8267
5	0.9251	0.8373	0.9070
6	0.9427	0.7570	0.8522
7	0.9400	0.7883	0.8549
8	0.9931	0.8208	0.9016
9	0.9416	0.7967	0.8619
10	0.9085	0.7313	0.8023
11	0.9007	0.8083	0.8935
12	0.8706	0.6167	0.7297
13	0.9287	0.8060	0.8737
14	0.9040	0.7223	0.8183
15	0.8756	0.6990	0.8008
Average	0.9240	0.7709	0.8543

Table 7.: KNN Pointwise Time Classifier Trained on Actual EEG (K=3). This classifier was built using the recorded EEG from a participant as the reference data. The classifier tried to assign the type of stimulus that was being used to produce the response to each EEG window. The * denotes Self Classification meaning that the results from that column are the classification results from the same data used to train the model.

KNN Pointwise Time Classifier Trained on Model Outputs (K=3)			
Participant	Actual EEG	Testing Data*	Training Data*
1	0.9080	0.9301	0.9375
2	0.6282	0.9552	0.9441
3	0.7151	0.9635	0.9573
4	0.7454	0.9218	0.9320
5	0.6966	0.9562	0.9455
6	0.5986	0.9499	0.9322
7	0.7657	0.9281	0.9255
8	0.8752	0.9354	0.9359
9	0.7180	0.9656	0.9505
10	0.6119	0.9417	0.9318
11	0.5162	0.9490	0.9521
12	0.4235	0.9583	0.9424
13	0.6615	0.9353	0.9297
14	0.6838	0.9322	0.9288
15	0.4558	0.9510	0.9362
Average	0.6669	0.9449	0.9388

Table 8.: KNN Pointwise Time Classifier Trained on Model Outputs (K=3). This classifier was built using the model outputs as the reference data. The classifier tried to assign the type of stimulus that was being used to produce the response to each EEG window. The * denotes Self Classification meaning that the results from that column are the classification results from the same data used to train the model.

KNN Frequency Magnitude Classifier Trained on Actual EEG (K=10)			
Participant	Actual EEG*	Testing Data	Training Data
1	1.0000	0.8290	0.9289
2	1.0000	0.8156	0.9287
3	1.0000	0.8759	0.9797
4	1.0000	0.7852	0.9260
5	1.0000	0.8238	0.9497
6	1.0000	0.7852	0.9033
7	1.0000	0.7716	0.8893
8	1.0000	0.8125	0.9190
9	1.0000	0.8582	0.9513
10	1.0000	0.7406	0.8898
11	1.0000	0.8292	0.9420
12	1.0000	0.7042	0.8081
13	1.0000	0.7904	0.9242
14	1.0000	0.7380	0.8908
15	1.0000	0.8073	0.9234
Average	1.0000	0.7978	0.9169

Table 9.: KNN Frequency Magnitude Classifier Trained on Actual EEG (K=10). This classifier was built using the recorded EEG from a participant as the reference data. The classifier tried to assign the type of stimulus that was being used to produce the response to each EEG window. The * denotes Self Classification meaning that the results from that column are the classification results from the same data used to train the model.

KNN Frequency Magnitude Classifier Trained on Model Outputs (K=10)			
Participant	Actual EEG	Testing Data*	Training Data*
1	0.9898	0.8717	0.9429
2	0.9723	0.8875	0.9654
3	0.9671	0.8968	0.9805
4	0.9741	0.8582	0.9528
5	0.9837	0.8749	0.9591
6	0.9237	0.8321	0.9520
7	0.9683	0.8665	0.9393
8	0.9746	0.8865	0.9555
9	0.9477	0.8926	0.9685
10	0.9548	0.8031	0.9516
11	0.9500	0.8573	0.9643
12	0.8973	0.8292	0.9482
13	0.9764	0.8321	0.9502
14	0.9328	0.8328	0.9416
15	0.9246	0.8396	0.9477
Average	0.9558	0.8574	0.9546

Table 10.: KNN Frequency Magnitude Classifier Trained on Model Outputs (K=10). This classifier was built using the model outputs as the reference data. The classifier tried to assign the type of stimulus that was being used to produce the response to each EEG window. The * denotes Self Classification meaning that the results from that column are the classification results from the same data used to train the model.

Correlation Between Autocorrelation of Actual EEG and Model Outputs for Training Data		
Participant	Mean	Standard Deviation
1	0.8390	0.1542
2	0.7624	0.1324
3	0.7898	0.1407
4	0.7943	0.1506
5	0.7814	0.1261
6	0.7527	0.1344
7	0.7693	0.1290
8	0.8164	0.1573
9	0.7869	0.1460
10	0.7383	0.1470
11	0.7193	0.1141
12	0.6978	0.1313
13	0.7582	0.1413
14	0.7512	0.1489
15	0.7036	0.0982
Average	0.7640	0.1368

Table 11.: Correlation between Autocorrelation of Actual EEG and Model Outputs for Training Data. The autocorrelations for all model outputs and all segments of actual EEG were calculated. Then the correlation between pairs that shared the same input flicker pattern was calculated. Statistics on the distributions of the correlations are presented in this table.

Correlation Between Autocorrelation of Actual EEG and Model Outputs for Testing Data		
Participant	Mean	Standard Deviation
1	0.7147	0.2057
2	0.6155	0.1465
3	0.6389	0.1854
4	0.6436	0.1791
5	0.6399	0.1414
6	0.6196	0.1328
7	0.6243	0.1596
8	0.7169	0.1780
9	0.6405	0.1665
10	0.5838	0.1169
11	0.5833	0.0934
12	0.5546	0.0815
13	0.6124	0.1595
14	0.6109	0.1530
15	0.5839	0.0894
Average	0.6255	0.1459

Table 12.: Correlation between Autocorrelation of Actual EEG and Model Outputs for Testing Data. The autocorrelations for all model outputs and all segments of actual EEG were calculated. Then the correlation between pairs that shared the same input flicker pattern was calculated. Statistics on the distributions of the correlations are presented in this table.

REFERENCES

[1] D. Yoshor et al. "Receptive Fields in Human Visual Cortex Mapped with Surface Electrodes". In: *Cerebral Cortex* 17.10 (Dec. 2006), pp. 2293–2302. ISSN: 1047-3211. DOI: 10.1093/cercor/bhl138. eprint: https://academic.oup.com/cercor/article-pdf/17/10/2293/880224/bhl138.pdf. URL: https://doi.org/10.1093/cercor/bhl138.

[2] B.A. Wandell and J. Winawer. "Imaging retinotopic maps in the human brain". In: *Vision Research* 51.7 (May 2011), pp. 718–737. DOI: 10.1016/j.visres.2010.08.004.

[3] G. Perry et al. "Retinotopic mapping of the primary visual cortex – a challenge for MEG imaging of the human cortex". In: *European Journal of Neuroscience* 34.4 (2011), pp. 652–661. DOI: https://doi.org/10.1111/j.1460-9568.2011.07777.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1460-9568.2011.07777.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1460-9568.2011.07777.x.

[4] I.I.A. Groen et al. "Temporal Dynamics of Neural Responses in Human Visual Cortex". In: *Journal of Neuroscience* 42.40 (2022), pp. 7562–7580. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.1812-21.2022. eprint: https://www.jneurosci.org/content/42/40/7562.full.pdf. URL: https://www.jneurosci.org/content/42/40/7562.

[5] J.J. Shih, D.J. Krusienski, and J.R. Wolpaw. "Brain-Computer Interfaces in Medicine". In: *Mayo Clinic Proceedings* 87.3 (Feb. 2012), pp. 268–279. DOI: 10.1016/j.mayocp.2011.12.008.

[6] F. Sobreira, C. Tremmel, and D.J. Krusienski. "Modeling the Visual Pathway for Stimulus Optimization in Brain-Computer Interfaces". In: *2018 26th European Signal Processing Conference (EUSIPCO)*. 2018, pp. 1672–1675. DOI: 10.23919/EUSIPCO.2018.8553194.

[7] B. Pesaran et al. "Investigating large-scale brain dynamics using field potential recordings: analysis and interpretation". In: *Nature Neuroscience* 21 (2018), pp. 903–919. DOI: 10.1038/s41593-018-0171-8.

[8] D.J. Heeger, E.P. Simoncelli, and J.A. Movshon. "Computational models of cortical visual processing." In: *Proceedings of the National Academy of Sciences* 93.2 (1996), pp. 623–627. DOI: 10.1073/pnas.93.2.623. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.93.2.623. URL: https://www.pnas.org/doi/abs/10.1073/pnas.93.2.623.

[9] G.W. Lindsay. "Convolutional Neural Networks as a Model of the Visual System: Past, Present, and Future". In: *Journal of Cognitive Neuroscience* 33.10 (Sept. 2021), pp. 2017–2031. ISSN: 0898-929X. DOI: 10.1162/jocn_a_01544. eprint: https://direct.mit.edu/jocn/article-pdf/33/10/2017/1962083/jocn_a_01544.pdf. URL: https://doi.org/10.1162/jocn%5C_a%5C_01544.

[10] D. Bálya et al. "A CNN framework for modeling parallel processing in a mammalian retina". In: *International Journal of Circuit Theory and Applications* 30.2-3 (2002), pp. 363–393. DOI: https://doi.org/10.1002/cta.204. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cta.204. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cta.204.

[11] R.T. Schirrmeister et al. "Deep learning with convolutional neural networks for EEG decoding and visualization". In: *Human Brain Mapping* 38.11 (2017),

pp. 5391–5420. DOI: https://doi.org/10.1002/hbm.23730. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/hbm.23730. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/hbm.23730.

[12] A. Wohrer and P. Kornprobst. "Virtual Retina: A biological retina model and simulator, with contrast gain control". In: *Journal of Computational Neuroscience* 26 (2009). DOI: 10.1007/s10827-008-0108-4. URL: https://doi.org/10.1007/s10827-008-0108-4.

[13] D.J. Heeger and K.O. Zemlianova. "A recurrent circuit implements normalization, simulating the dynamics of V1 activity". In: *Proceedings of the National Academy of Sciences* 117.36 (2020), pp. 22494–22505. DOI: 10.1073/pnas.2005417117. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.2005417117. URL: https://www.pnas.org/doi/abs/10.1073/pnas.2005417117.

[14] A.G. Habashi et al. "Generative adversarial networks in EEG analysis: an overview". In: *Journal of NeuroEngineering and Rehabilitation* 20.40 (Apr. 2023).

[15] D. Purves and G.J. Fitzpatrick. *Neuroscience*. 2nd Edition. Sinauer Associates, 2001.

[16] P. Lennie. "6 - The Physiology of Color Vision". In: *The Science of Color (Second Edition)*. Ed. by S.K. Shevell. 2nd Edition. Amsterdam: Elsevier Science Ltd, 2003, pp. 217–246. ISBN: 978-0-444-51251-2. DOI: https://doi.org/10.1016/B978-044451251-2/50007-6. URL: https://www.sciencedirect.com/science/article/pii/B9780444512512500076.

[17] T. Huff, N. Mahabadi, and P. Tadi. "Neuroanatomy, Visual Cortex". In: *StatPearls Publishing* (2023).

[18] U. Neisser. *Cognitive psychology: Classic edition*. Psychology press, 2014.

[19] R.A. Kinchla and J.M. Wolfe. "The order of visual processing: "Top-down," "bottom-up," or "middle-out"". In: *Perception & Psychophysics* 25 (3 May 1979). DOI: 10.3758/BF03202991. URL: https://doi.org/10.3758/BF03202991.

[20] D. Ingle, M.A. Goodale, and R.J. Mansfield. *Analysis of Visual Behavior*. Ed. by NATO Advanced Study Institute and Brandeis University. MIT Press, 1982.

[21] *The Visual Cortex*. https://azretina.sites.arizona.edu/node/1069. Accessed: 2023-11-20.

[22] C. Baiano and M. Zeppieri. "Visual Evoked Potential". In: *StatPearls* (Jan. 2023). URL: https://www.ncbi.nlm.nih.gov/books/NBK582128/.

[23] H. Hassankarimi et al. "Low-contrast Pattern-reversal Visual Evoked Potential in Different Spatial Frequencies". In: *J Opthalmic Vis Res* 15.3 (Aug. 2020), pp. 362–371. DOI: 10.18502/jovr.v15i3.7455.

[24] N.R. Waytowich, Y. Yamani, and D.J. Krusienski. "Optimization of Checkerboard Spatial Frequencies for Steady-State Visual Evoked Potential Brain–Computer Interfaces". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 25.6 (2017), pp. 557–565. DOI: 10.1109/TNSRE.2016.2601013.

[25] M. Yang et al. "A review of researches on decoding algorithms of steady-state visual evoked potentials". In: *Chinese Journal of Biomedical Engineering* 39.2 (Apr. 2022), pp. 416–425. DOI: 10.7507/1001-5515.202111066.

[26] E.E. Sutter. "The visual evoked response as a communication channel". In: *Proceedings of IEEE Symposium of Biosensors* (1984), pp. 95–100.

[27] B. He et al. “Brain–Computer Interfaces”. In: *Neural Engineering*. Ed. by Bin He. Cham: Springer International Publishing, 2020, pp. 131–183. ISBN: 978-3-030-43395-6. DOI: 10.1007/978-3-030-43395-6_4. URL: https://doi.org/10.1007/978-3-030-43395-6_4.

[28] E.E. Sutter. “The brain response interface: communication through visually induced electrical brain responses”. In: *Journal of Microcomputer Applications* 15.1 (1992), pp. 31–45.

[29] G.D. Johnson and D.J. Krusienski. “Computational EEG Analysis for Brain-Computer Interfaces”. In: *Computational EEG Analysis: Methods and Applications*. Ed. by Chang-Hwan Im. Singapore: Springer Singapore, 2018, pp. 193–214. ISBN: 978-981-13-0908-3. DOI: 10.1007/978-981-13-0908-3_9. URL: https://doi.org/10.1007/978-981-13-0908-3_9.

[30] G. Bin et al. “A high-speed BCI based on code modulation VEP”. In: *Journal of Neural Engineering* 8.2 (Mar. 2011), p. 025015. DOI: 10.1088/1741-2560/8/2/025015. URL: https://dx.doi.org/10.1088/1741-2560/8/2/025015.

[31] F. Rosenblatt et al. *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*. Vol. 55. Spartan books Washington, DC, 1962.

[32] K. Fukushima. “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”. In: *Biology Cybernetics* 36 (1980), pp. 193–202. URL: https://doi.org/10.1007/BF00344251.

[33] S. Linnainmaa. “Taylor expansion of the accumulated rounding error.” In: *BIT* 16 (1976), pp. 146–160. URL: https://doi.org/10.1007/BF01931367.

[34] I. Sutskever et al. “On the importance of initialization and momentum in deep learning”. In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by Sanjoy Dasgupta and David McAllester. Vol. 28. Proceedings of Machine Learning Research 3. Atlanta, Georgia, USA: PMLR, June 2013, pp. 1139–1147. URL: https://proceedings.mlr.press/v28/sutskever13.html.

[35] D.P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].

[36] A. Graves. *Generating Sequences With Recurrent Neural Networks*. 2014. arXiv: 1308.0850 [cs.NE].

[37] C. Esteban, S.L. Hyland, and G. Rätsch. *Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs*. 2017. arXiv: 1706.02633 [stat.ML].

[38] K. Sako, B.N. Mpinda, and P.C. Rodrigues. “Neural Networks for Financial Time Series Forecasting”. In: *Entropy* 24.5 (May 2022).

[39] M. Miljanovic. “Comparative analysis of Recurrent and Finite Impulse Response Neural Networks in Time Series Prediction”. In: *Indian Journal of Computer Science and Engineering* 3.1 (2012), pp. 180–191.

[40] G. Zerveas et al. “A Transformer-Based Framework for Multivariate Time Series Representation Learning”. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. KDD ’21. Virtual Event, Singapore: Association for Computing Machinery, 2021, pp. 2114–2124. ISBN: 9781450383325. DOI: 10.1145/3447548.3467401. URL: https://doi.org/10.1145/3447548.3467401.

[41] H. Ismail Fawaz et al. “Deep learning for time series classification: a review”. In: *Data Mining and Knowledge Discovery* 33 (2019), pp. 917–963.

[42] A. Dong, A. Starr, and Y. Zhao. “Neural network-based parametric system identification: a review”. In: *International Journal of Systems Science* 54.13 (2023), pp. 2676–2688. DOI: 10.1080/00207721.2023.2241957. eprint: https://doi.org/10.1080/00207721.2023.2241957. URL: https://doi.org/10.1080/00207721.2023.2241957.

[43] B.A. Wandell. *Foundations of Vision*. Sinauer Associates, 1995.

[44] N.R. Waytowich and D.J. Krusienski. “Development of an Extensible SSVEP-BCI Software Platform and Application to Wheelchair Control”. In: *International IEEE EMBS Conference on Neural Engineering* 8 (May 2017), pp. 259–532. DOI: 10.1109/NER.2017.8008406.

[45] V. Martínez-Cagigal et al. “Brain–computer interfaces based on code-modulated visual evoked potentials (c-VEP): a literature review”. In: *Journal of Neural Engineering* 18.6 (Nov. 2021), p. 061002. DOI: 10.1088/1741-2552/ac38cf. URL: https://dx.doi.org/10.1088/1741-2552/ac38cf.

[46] S.W. Golomb. *Shift Register Sequences*. Aegan Park, 1982.

[47] D. Aminaka, Shoji Makino, and T.M. Rutkowski. “Chromatic and high-frequency cVEP-based BCI paradigm”. In: *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2015, pp. 1906–1909. DOI: 10.1109/EMBC.2015.7318755.

[48] P. Stawicki and I. Volosyak. “cVEP Training Data Validation-Towards Optimal Training Set Composition from Multi-Day Data”. In: *Brain Sciences*

12.2 (2022). ISSN: 2076-3425. DOI: 10.3390/brainsci12020234. URL: https://www.mdpi.com/2076-3425/12/2/234.

[49] F. Gembler et al. "Asynchronous c-VEP communication tools-efficiency comparison of low-target, multi-target and dictionary-assisted BCI spellers". In: *Scientific Reports* 10 (2020).

[50] N.R. Waytowich and D.J. Krusienski. "Spatial decoupling of targets and flashing stimuli for visual brain–computer interfaces". In: *Journal of Neural Engineering* 12.3 (Apr. 2015), p. 036006. DOI: 10.1088/1741-2560/12/3/036006. URL: https://dx.doi.org/10.1088/1741-2560/12/3/036006.

[51] H. Riechmann, A. Finke, and H. Ritter. "Using a cVEP-Based Brain-Computer Interface to Control a Virtual Agent". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 24.6 (2016), pp. 692–699. DOI: 10.1109/TNSRE.2015.2490621.

[52] G.T. Buračas and G.M. Boynton. "Efficient Design of Event-Related fMRI Experiments Using M-Sequences". In: *NeuroImage* 16.3, Part A (2002), pp. 801–813. ISSN: 1053-8119. DOI: https://doi.org/10.1006/nimg.2002.1116. URL: https://www.sciencedirect.com/science/article/pii/S105381190291116X.

[53] g.tec. *g.USBamp*. Austria, 2017.

[54] G. Schalk et al. "BCI2000: A General-Purpose Brain-Computer Interface (BCI) System". In: *IEEE Transactions on Biomedical Engineering* 51.6 (June 2004).

[55] The MathWorks Inc. *MATLAB version: 9.13.0 (R2022b)*. Natick, Massachusetts, United States, 2022. URL: https://www.mathworks.com.

[56] Harold Hotelling. "RELATIONS BETWEEN TWO SETS OF VARIATES*". In: *Biometrika* 28.3-4 (Dec. 1936), pp. 321–377. ISSN: 0006-3444. DOI: 10.1093/biomet/28.3-4.321. eprint: https://academic.oup.com/biomet/article-pdf/28/3-4/321/586830/28-3-4-321.pdf. URL: https://doi.org/10.1093/biomet/28.3-4.321.

[57] R. Reed and R.J. Marks. *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. The MIT Press, Feb. 1999. ISBN: 9780262282215. DOI: 10.7551/mitpress/4937.001.0001. URL: https://doi.org/10.7551/mitpress/4937.001.0001.

[58] Phillip Isola et al. *Image-to-Image Translation with Conditional Adversarial Networks*. 2018. arXiv: 1611.07004 [cs.CV].

[59] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. eprint: https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[60] Ashish Vaswani et al. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].

[61] E.T.B. Lundby et al. *Sparse neural networks with skip-connections for identification of aluminum electrolysis cell*. 2023. arXiv: 2301.00582 [eess.SY].

[62] J.D. Kropotov. "Introduction: Basic Concepts of QEEG and Neurotherapy". In: *Quantitative EEG, Event-Related Potentials and Neurotherapy*. Ed. by J.D. Kropotov. San Diego: Academic Press, 2009, pp. xxxi–lviii. ISBN: 978-0-12-374512-5. DOI: https://doi.org/10.1016/B978-0-12-374512-5.

00027-X. URL: https://www.sciencedirect.com/science/article/pii/B978012374512500027X.

[63] P. Welch. "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms". In: *IEEE Transactions on Audio and Electroacoustics* 15.2 (1967), pp. 70–73. DOI: 10.1109/TAU.1967.1161901.

[64] E. Fix and J.L. Hodges. *Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties*. Archived (PDF) from the original on September 26, 2020. Randolph Field, Texas, 1951.