Radboud University

Simulating and classifying EEG scalp data for code-modulated visual evoked potential brain computer interfaces

Master thesis in Artificial Intelligence
Radboud University Nijmegen

Author:
Thijs Luttikholt,
Student number: S1007894
t.luttikholt@student.ru.nl
Radboud University Nijmegen
Master AI - Intelligent Technology

Daily supervisor
Dr. Jordy Thielen
Radboud University Nijmegen

Second reader
Dr. Michael Tangermann
Radboud University Nijmegen

July 6, 2022

Abstract

Objective. Within the field of brain-computer interface (BCI) research, a common challenge is the lack of ground truth data. Additionally, a BCI typically requires a calibration session in order to reach strong performances with new users, which increases the amount of time required to perform experiments. In this study, a simulation framework is implemented as a forward model which could generate EEG data artificially, in order to investigate whether simulated data could be used instead of empirical data when training machine learning methods. *Approach*. Four versions of a forward model were implemented. Furthermore, two inverse models were implemented, namely a reconvolution model using canonical correlation analysis and an EEGNet model. These inverse models were trained to classify the classes of c-VEP data, using either empirical data or simulated data, followed by a performance evaluation by testing on empirical data. *Main results*. By training inverse models on data generated with a forward model, the inverse models can learn to classify empirical c-VEP data to a certain extent which surpasses chance level. However, this performance is not yet at a level where it can be used in real life applications. *Significance*. This study shows that it is possible to achieve performances higher than chance level on empirical data by using machine learning techniques that were trained using simulated data. Further improvement of the forward models that were used in this study could potentially reduce the amount of empirical data that is needed in future studies within the field of VEP-based BCI.

Contents

1 Introduction

A brain-computer interface (BCI) is a system that enables communication without relying on muscle control (McFarland, Sarnacki, Vaughan, & Wolpaw, 2005). A user of such a system usually performs a specific task, during which brain activity is typically measured using electroencephalography (EEG). The recorded data is subsequently subjected to a signal processing pipeline. This pipeline typically contains preprocessing, feature extraction, feature selection, machine learning for classification or regression, model evaluation, and post-processing steps. This process serves to extract relevant information from the EEG recording, based on which a device can be controlled.

One of the control signals that is often used for BCI is the visual evoked potential (VEP). This paradigm has become known for being a robust and fast method for communication using BCI, having been significantly improved since its inception (Gao, Wang, Gao, & Hong, 2014). Such a VEP is a type of brain activity which is triggered by an external visual stimulus, to which it is also time-locked. Within this VEP paradigm, the method that makes use of the steady-state VEP (SSVEP) response has originally been popular due to its simplicity and speed. This method presents particular flashing frequencies for commands, with the EEG response subsequently matching the flashing frequency of the command that is being paid attention to by the user (Bin, Gao, Wang, Hong, & Gao, 2009; Vialatte, Maurice, Dauwels, & Cichocki, 2010).

A different method within the VEP paradigm is called code-modulated VEP (c-VEP). In this method, commands are flashing according to pseudo-random noise sequences. Similar to the SSVEP approach, the generated EEG responses for the c-VEP approach are relatively more correlated with the specific attended command than with other commands (Martínez-Cagigal et al., 2021). The responses used in both c-VEP and SSVEP approaches are responses to a sequence of flashes. In relation to such sequences of events, the linear superposition hypothesis (see e.g. Notbohm, Kurths, & Herrmann (2016) and Capilla, Pazo-Alvarez,Darriba,Campo, & Gross (2011)) states that the response to a sequence of events can be modeled as the linear summation of the responses to individual events. In the case of the c-VEP and SSVEP paradigms, such an individual event would be an individual uniform flash, which is referred to as the flash-VEP (Vialatte et al., 2010), of which the P2 is a major component (Wyatt-McElvain, Arruda, & Rainey, 2018). One other specific characteristic of the flash-VEP is the amplitude between the N75 and P100 peaks, which has previously been used in a procedure for the detection of postoperative visual impairment (Hayashi & Kawaguchi, 2017).

Researchers that investigate BCI techniques, including those techniques that make use of the VEP paradigm, face a variety of challenges. These challenges include how to handle the existence of artifacts and noise within the measured EEG signals. The presence of artifacts and noise could reduce the accuracy of any BCI system due to decreasing the signal-to-noise ratio (SNR), including the c-VEP systems that are investigated in this study. An example of such an artifact is the line noise which can be caused by electromagnetic interference. Other artifacts are those caused by eye movement, muscle movement, and cardiac activity (Urigüen &

Garcia-Zapirain, 2015; Fatourechi, Bashashati, Ward, & Birch, 2007). Examples of noise in an EEG are pink noise, which is proportional to a $1/f$ power spectrum, and white noise, which has a constant power across the measured frequency range (Barry & De Blasio, 2021). Various techniques have been developed over the years for handling noise and artifacts. Specifically for artifacts, one can distinguish between artifact avoidance, artifact detection, artifactual segment rejection, and artifact removal (Islam, Rastegarnia, & Yang, 2016).

Compared to the described challenges, two other challenges within the field of BCI have a stronger relation to the aim of this study. The first of these challenges is the challenge that, whenever a new user uses a BCI system, the system usually requires a calibration session before users obtain meaningful control. These calibration sessions are usually necessary due to inter-user and intra-user variability of the recorded EEG signals. For different users, differences in mental strategies, physiologies, or other attributes can cause differences in the properties of the measured EEG. Within the VEP paradigm, such differences can occur in the shape of the recorded flash-VEP response. For example, the latency of the previously described P2 component or the amplitude between the N75 and P100 peaks could differ between participants. In order to improve the classification performance of the BCI device for new users, calibration is typically necessary to account for the differences in EEG responses. Such calibration requires additional subject-specific data to be recorded, which in turn increases the duration and the cost of the experiments.

This leads us to the second challenge that has a stronger relation to the aim of this study, which is the challenge concerning the cost of human experiments. Within the field of c-VEP BCI, one open question is which code would lead to optimal performance. In order to evaluate this, new data, obtained by performing measurements with different codes, would be required. However, such experiments would be costly, making this question more difficult to answer. In addition, experiments with a large amount of participants would be preferable for statistical validation. However, each of these participants would need to be instructed in the BCI functionality, and would need to be prepared for EEG measurement. Furthermore, as was mentioned earlier, most BCI systems need a calibration phase before they can produce more accurate results, and therefore require the recording of additional data. All of these factors together lead to experiments that take a larger amount of time to perform. In addition, with human BCI experiments it is difficult to precisely specify a ground truth for the underlying signal. As a result, there is a general lack of ground truth data for training BCI techniques (Lindgren, Merlini, Lecuyer, & Andriulli, 2018).

There are various approaches one can take in order to reduce the impact of the challenges of calibration and lack of ground truth data. One such approach is that of transfer learning. In transfer learning, knowledge from other sessions is transferred to new sessions, with the aim of reducing the amount of calibration data that is required. A second approach could be to create novel BCI systems that could reach acceptable performance without requiring a calibration phase.

An alternative approach is to artificially increase the amount of data that is available by simulating such data. This does not remove the necessity of collecting empirical data for eval-

uation of a BCI technique. However, given a realistic simulation, such simulated data could be used to train a classification framework or to investigate intuitions about a certain task. A simulation framework can be created such that inter-user differences are included in the simulated data, by generating data under a wide range of parameter values. By having access to large amounts of simulated data generated under a wide range of parameter values, of which the ground truth is available, machine learning methods that were trained on this data could potentially show robustness to such variations between trials, and could therefore potentially be relatively robust to inter-user differences. This in turn could reduce the length of the calibration sessions needed for improving the accuracy of a system. An additional advantage of using a simulation framework is that the aspects of an experimental paradigm, such as the codes that are used in the c-VEP paradigm, can be optimized without the necessity of recording empirical data. This advantage can be seen in a study by Torres & Daly (2021), where the effects of, among others, different stimuli modulation sequences on the information transfer rate and accuracy of c-VEP-based BCI approaches were evaluated partially using simulated data.

In this study, we investigated the discussed simulation approach within the field of c-VEP BCI. More specifically, a simulator, which will henceforth be referred to as the forward model, was implemented which can generate single-channel EEG data for c-VEP BCI. This forward model was used to generate EEG data for the c-VEP paradigm using a variety of codes. In addition, two inverse models were implemented, which were trained to predict the classes of trials that were simulated by the forward model. By training these models on simulated data, they could subsequently be directly applied on empirical data, without any further training or calibration.

The forward and inverse models were used to investigate a topic regarding direct classification. More specifically, this study investigated to what degree an inverse model trained on data simulated by the forward model would be able to classify empirical data, without additional training.

2 Related work/background

This study includes the implementation of a simulation framework, of which the data is used in subsequent analyses. Such approaches that make use of simulators to perform statistical inference are known under the umbrella term of simulation-based inference (SBI) (Cranmer, Brehmer, & Louppe, 2020). Various methods for generating simulated data exist, one example of which is the use of generative models such as generative adversarial networks (GANs) and variational auto-encoders (VAEs) (Aznan et al., 2019). A different simulation approach used segmentation on existing data, after which the segments were randomly re-assembled to create new data (Lotte, 2011). An additional approach would be to explicitly model the signal and noise components of the simulated EEG response. Subsequently, these components would be combined, taking into account a specified SNR value. This requires a researcher to make important decisions regarding the exact manner in which to model the signal and noise

components, as well as how these components are combined. These decisions have previously been made by studies introducing two existing simulation frameworks for EEG data, by the names of EEGSourceSim and SimBCI (Barzegaran, Bosse, Kohler, & Norcia, 2019; Lindgren et al., 2018). The SimBCI framework was later used in a study that evaluated its effectiveness within the c-VEP domain, by adapting the framework to be usable in this new domain (Torres & Daly, 2021).

In order to classify data that was generated by such a simulation framework or by performing empirical recordings, an inverse model is required. Previous studies have investigated such classification approaches within the field of BCI. One example is a reconvolution model using canonical correlation analysis (CCA), which was proposed by Thielen et al. for reducing the calibration time of c-VEP BCI systems. This study led to a zero-training approach using an encoding model that predicts the EEG responses based on the stimulus sequence that is used (Thielen, Marsman, Farquhar, & Desain, 2021). In this approach, one spatial filter and one temporal filter are learned, which are subsequently used in a template matching classifier. Learning only one of each of these filters does not allow the approach to learn the variances in data. This limitation may reduce the effectiveness of this approach when it is applied on simulated data or on cross-subject empirical data, where a degree of variance is to be expected.

Alternative classification approaches have been investigated in previous studies. Among the proposed methods is the EEGNet, which is a convolutional neural network that was found to robustly learn a variety of interpretable features across a variety of BCI tasks (Lawhern et al., 2018). Due to the robustness across tasks that was demonstrated by the EEGNet, it is expected to be less affected by the variance within simulated data or cross-subject empirical data. A second alternative classification approach, specifically within the field of c-VEP based BCI, is a method which combines the EEG2Code method with deep learning techniques. The EEG2Code method itself was designed to predict visual stimulation patterns based on EEG data (Nagel & Spüler, 2018). The combination of this EEG2Code with deep learning techniques was found to reach a high information transfer rate (ITR), indicating that it is a relatively fast BCI system (Nagel & Spüler, 2019). The ITR metric itself is often used in research regarding BCI-based spellers. It takes into account not only the speed of classification, but also the amount of possible choices and the accuracy of the classifications.

3 Preliminaries

3.1 Deep learning in EEG-based BCI

Deep learning is a technique often used in artificial intelligence (AI). This technique uses artificial neural networks (ANNs), consisting of various nodes, layers, and connections. These layers can have a wide variety of functionalities, with two examples being dense layers and convolutional layers. It is possible to train such networks for a specific task by providing proper loss functions, inputs, and ground truth values. This training optimizes the parameters of the network in such a way that the values obtained from the loss function are minimized.

Applying these deep learning techniques to the field of EEG-based BCI requires careful consideration of what each layer of an ANN would represent. A classical BCI signal processing pipeline contains preprocessing, feature extraction, feature selection, machine learning for classification or regression, model evaluation, and post-processing steps. Three of these steps can be replaced by a properly designed ANN. Such a replacement is exemplified by the EEGNet architecture developed by Lawhern et al (2018). The first of the steps that can be replaced is the feature extraction step, where deep learning can be used to learn spatial and temporal filters. The EEGNet architecture represents these filters by employing one-dimensional convolutional layers. The type of filter that is learned by these layers depends on the shape of the filter.

The second of the replaceable steps is the feature selection step. In the EEGNet architecture, this is performed by first using regular convolution, followed by pointwise convolution. This combination serves to first learn kernels that summarize the individual feature maps learned by previous layers, followed by optimal merging of these summarized features. This separates the processes of learning how to summarize feature maps and how to merge these features, thereby decoupling the relations within and across the various feature maps. An additional benefit of this combination is that it reduces the amount of parameters that has to be fit during the training phase of the model, compared to directly applying regular convolution with the same input and output shapes as the combination.

The third and final step of the classical BCI signal processing pipeline that was modeled in EEGNet was the classification and regression step. Due to this step typically being performed using machine learning, its implementation within the EEGNet is relatively straightforward. In the EEGNet architecture, this classification step is performed by a dense layer that performs softmax classification (Lawhern et al., 2018). The manner in which the various steps of the BCI signal processing pipeline were modeled in EEGNet was similar to the approach used by Nagel & Spüler (2019).

4 Methods

4.1 Code availability

The Python code that was used for the forward models, the inverse models, and the analyses will be made available at GitHub:

https://github.com/ThijsLuttikholt/cVEP_simulation

4.2 Empirical data

In this study, an empirical dataset of c-VEP data was used both for the evaluation of the simulation framework and to serve as a guideline for the characteristics of the desired outputs of the forward model. The used dataset was the offline analysis subset of the data that was made available in a prior study by Thielen et al. (2021). In the study by Thielen et al., a set of 65

Gold codes was generated and subsequently modulated. A Gold code is a binary sequence, with a set of Gold codes exhibiting minimal cross- and auto-correlation between them. Out of these 65 codes, the best subset of 20 codes was extracted and used for offline analysis. This data was recorded with 30 participants, using the set of 20 modulated Gold codes as stimulation patterns based on which the luminances of various cells on the screen of a tablet with a 60 Hz refresh rate were modulated. These codes are bit-sequences, which are series of ones and zeroes, In these sequences, a bit value of one indicates that the corresponding cell on the screen should be illuminated at that point in the code. Conversely, a bit value of zero indicates that that the cell should not be illuminated. Each of these modulated Gold codes used only two events, namely short events and long events. Here, an event indicates an illumination of the corresponding cell. Short events are represented as a bit value of one, followed by either one or two zeroes (10 or 100), given the 60 Hz screen refresh rate, the corresponding illumination of the cell lasted for 16.67 ms. Long events are represented by two bit values of one, followed by one or two zeroes (110 or 1100), for which the corresponding illumination would last for 33.34 ms. Each full code contained 126 bits, which lasted for 2.1 seconds given the 60 Hz refresh rate of the tablet that was used.

In the measurement of this EEG data, 8 electrodes were used which were placed according to the 10-20 system at the locations Fz, T7, T8, POz, O1, Oz, O2 and Iz. The data were recorded at 512 Hz with these electrodes, and amplified by means of a Biosemi ActiveTwo amplifier. For each of the 30 participants, the used data contained 100 trials, distributed equally over the 20 codes. This data was subsequently preprocessed by using overlap-add finite impulse response (FIR) spectral band-pass filtering between 2 and 30 Hz, followed by downsampling to a frequency of 120 Hz. After preprocessing, each trial contained 31.5 seconds of measured data, amounting to 3780 samples. During the 31.5 seconds of measurement, each of the codes was repeated 15 times. For more details regarding the original data, we refer to Thielen et al. (2021).

4.3 Inverse models

The empirical data can be classified using inverse models. Two such inverse models were implemented, namely a CCA model and an EEGNet model.

4.3.1 Baseline: CCA

The baseline inverse model that was used in this study is the so-called n-train version of the approach introduced by Thielen et al. (2021). This approach used a template matching classifier in order to predict the attended cell out of n possible cells. Let $\mathbf{X} \in \mathbb{R}^{N_c, N_s}$ denote a single trial containing N_s samples for each of N_c EEG channels. Furthermore, let $\mathbf{w} \in \mathbb{R}^{N_c}$ denote a spatial filter over N_c channels. The initial step of the classification, namely to spatially filter the single trial, can now be defined as follows:

$$\mathbf{x} = \mathbf{w}^T \mathbf{X}, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^{N_s}$ is the spatially filtered trial. This single trial is subsequently compared to each of n template responses $\mathbf{t}_i \in \mathbb{R}^{N_s}$, i.e. one template response for each of the n classes. The similarity between $\mathbf{x}$ and each of the template responses is computed as the Pearson's correlation, which is maximized to obtain a class prediction:

$$\hat{y} = \arg\max_i \frac{\mathbf{x}^T \mathbf{t}_i}{\sqrt{\mathbf{x}^T \mathbf{x} \cdot \mathbf{t}_i^T \mathbf{t}_i}} \tag{2}$$

In the n-train approach, the template responses are built using a reconvolution encoding model. The first step in this model is to map the codes, which are bit-sequences, to an event matrix. This means that each bit-sequence is split into event-sequences, where one event-sequence is created for each type of event. As was discussed previously, the paradigm that was used to record the empirical data used two types of events, namely short and long events. In order to match the sampling frequency used to measure said empirical data, the codes, which originally had a 60 Hz frequency due to the refresh rate of the used tablet, were upsampled to a 120 Hz sampling frequency. The full event matrix for a class i therefore has a shape of $\mathbf{E}_i \in \mathbb{R}^{N_e, N_s}$, with N_e indicating the amount of events, which is 2 in this case, and N_s indicating the number of samples.

After creating the event matrix, a so-called structure matrix is created. This is a matrix that not only shows the onset of events, but also their duration, and therefore also shows overlap between events. An example of such a structure matrix can be found in Figure 1. The structure matrix follows a Toeplitz-like structure, representing both the onsets of events and the manner in which they overlap. For each of the N_e events, a separate structure matrix is created, after which the N_e individual structure matrices are concatenated to form the full structure matrix for a class i. This leads to the following shape: $\mathbf{M}_i \in \mathbb{R}^{N_e * N_r, N_s}$. In this shape, the N_r value indicates the amount of samples included in the response to an event, i.e. the assumed duration of the event.

After the creation of the structure matrix, canonical correlation analysis (CCA) is employed to obtain a spatial filter and a response vector. This is done by stacking the structure matrices according to the order in $\mathbf{X}$ for all k training trials:

$$\mathbf{M} = [\mathbf{M}_{y_1}, \mathbf{M}_{y_2}, ..., \mathbf{M}_{y_k}], \tag{3}$$

where $\mathbf{M}_{y_i} \in \mathbb{R}^{N_e * N_r, N_s}$ indicates the structure matrix for the class indicated by the label y_i in training trial i. This leads to a full structure of $\mathbf{M} \in \mathbb{R}^{N_e * N_r, N_s * k}$. Additionally, the trials in $\mathbf{X}$ are concatenated to $\mathbf{S} \in \mathbb{R}^{c, N_s * k}$. Now, an optimized spatial filter $\mathbf{w}$ and a temporal response vector $\mathbf{r}$ are found using CCA:

$$\arg\max_{\mathbf{w},\mathbf{r}} \rho(\mathbf{w}^T \mathbf{S}, \mathbf{r}^T \mathbf{M}') = \frac{\mathbf{w}^T \mathbf{S} \mathbf{M}'^T \mathbf{r}}{\sqrt{\mathbf{w}^T \mathbf{S} \mathbf{S}^T \mathbf{w} \cdot \mathbf{r}^T \mathbf{M}' \mathbf{M}'^T \mathbf{r}}}, \tag{4}$$

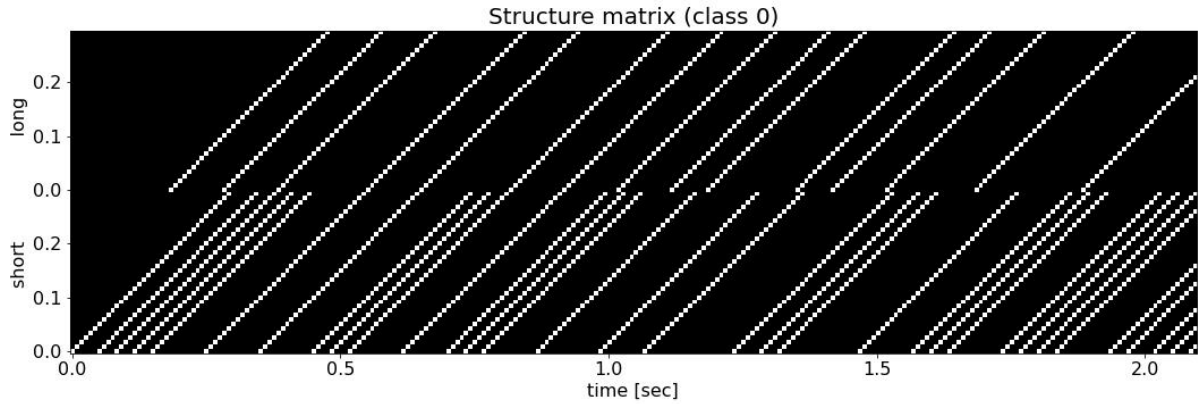

Figure 1: Structure matrix for modulated Gold code 0. This matrix has shape $\mathbb{R}^{N_e * N_r, N_s}$. The x-axis indicates the 252 samples, corresponding to 2.1 seconds of data, upsampled to a 120 Hz sampling frequency in order to match the sampling frequency that was used to measure the empirical data. The y-axis shows 36 samples, corresponding to 0.3 seconds for each of the two event classes, namely long and short events. The bottom row for each event shows the starting points of the various occurrences of the event. Each occurrence lasts for 0.3 seconds, as can be seen by the diagonal entries following the starting points.

where $\mathbf{w}$ is the spatial filter that can be used in Equation 1 to spatially filter individual trials, and $\mathbf{r} \in \mathbb{R}^{N_r * N_e}$ is the response vector containing the concatenation of the responses to the various event types. This response vector can subsequently be used to predict the template responses for any class for which a structure matrix has been calculated:

$$\mathbf{t}_i = \mathbf{r}^T \mathbf{M}_i \tag{5}$$

4.3.2 EEGNet

The second inverse model that was used in this study is a deep learning model by the name of EEGNet (Lawhern et al., 2018). The generalizability of this model has been tested on four different BCI paradigms, and the model showed robust performance across the tasks. Due to the desire for generalization across various participants within the chosen c-VEP domain, a robust model would be preferable. The choice to include EEGNet as a second model was made due to the nature of CCA, which learns only one spatial and one temporal filter for the full dataset. Learning only one of each of these filters for a large amount of data may lead to a reduced performance on datasets with large variance, due to such variance being difficult to capture using only a single filter. We hypothesize that CCA might not perform well on the simulated data, which is data that would likely contain high variance. The reason for this expected high variance is that the simulated data should be representative for a group consisting of multiple participants with inter-subject differences. Therefore, such inter-subject differences would be expected to be captured by the forward model in the form of high variance in the simulated data.

The EEGNet architecture itself can be divided into three main blocks. The first block contains two convolutional steps in sequence, with batch normalization in between. The first of these steps consists of F_1 one-dimensional convolutional filters, which learn temporal kernels of size $(1,T_f)$ where T_f is half of the sampling frequency. The second step is another one-dimensional convolution, which in this case is used to learn D spatial filters of size $(C,1)$, where C is the amount of input EEG channels, for each of the F_1 temporal kernels. This structure enables the model to learn frequency-specific spatial filters. This is followed by, in order, an exponential linear unit (ELU) activation layer, a two-dimensional average pooling layer with kernel size (1,4), and a dropout layer with dropout probabilities of 0.25 when training on empirical data and 0.5 when training on simulated data. In the second block, a one-dimensional convolution is followed by F_2 pointwise convolutions. Following this combination are, in order, a batch normalization, an ELU activation layer, a two-dimensional average pooling layer, a dropout layer, and lastly a flattening operation. The third and final block of the EEGNet is the classification block. This block originally consists of a dense layer that performs softmax classification over the n possible classes. In this study, this is modeled using a two-dimensional convolution layer

In the study by Lawhern et al. (2018), two specific versions of the EEGNet were evaluated. These versions, named according to the rule EEGNet-F_1,D, were the EEGNet-4,2 and the EEGNet-8,2. However, the largest amount of participants that was used in any of the four paradigms tested in said study was 26. In comparison, a full simulation framework would be expected to generate data that is representative for a larger participant group, thereby being expected to include a higher degree of variance. Therefore, we hypothesize that a version of the EEGNet with a larger capacity would be beneficial for this study. Such a larger-capacity version of EEGNet was found in study by Waytowich et al. (2018), who used an EEGNet-96,1 architecture in an approach using the SSVEP response. TThis study features that same architecture to make use of its larger capacity. The exact structure of the architecture as it was used in this study, including its parameters, can be found in Table 1.

The EEGNet was trained using an Adam optimizer and a cross entropy loss function. The batch sizes for the training, validation and testing data were set to 32. Lastly, the learning rate for the training was set to 6e-3. For this learning rate, a learning rate scheduler was implemented. This scheduler multiplied the learning rate by a factor of 0.1 whenever 5 successive iterations occurred without improving upon the highest recorded validation accuracy.

During the training phase of the network, a maximum of 500 iterations would be executed before ending the training of the network. However, the training additionally made use of an early stopping strategy, resulting in a second point at which the training could end. In the event that 20 successive iterations occurred without surpassing the highest recorded validation accuracy, the training was stopped, and the network was reset to the state it had when the highest validation accuracy was obtained.

Block	Layer	#filters	kernel size	stride	padding	#params	output	activation	options
1	Input						(B,C,T)		
	Reshape						$(B,1,C,T)$		
	2D convolution	F_1	$(1,T_f)$	1	$(0,T_f//2)$	$T_f * F_1$	(B,F_1,C,T)	Linear	$Bias = False$
	BatchNorm					$2 * F_1$	(B,F_1,C,T)		
	2D convolution	$D * F_1$	$(C,1)$	1	$(0,0)$	$C * D * F_1$	$(B,D * F_1,1,T)$	Linear	$Max\_norm = 1$, $bias = False$, $groups = F_1$
	BatchNorm					$2 * D * F_1$	$(B,D * F_1,1,T)$		
	Activation						$(B,D * F_1,1,T)$	ELU	
	2D average pooling		$(1,4)$	$(1,4)$			$(B,D * F_1,1,T//4)$		
	Dropout						$(B,D * F_1,1,T//4)$		$p = 0.25$ or $p = 0.5$
2	2D convolution	$D * F_1$	$(1,16)$	1	$(0,16//2)$	$16 * D * F_1$	$(B,D * F_1,1,T//4)$	Linear	$bias = False$, $groups = D * F_1$
	2D convolution	F_2	$(1,1)$	1	$(0,0)$	$F_2 * (D * F1)$	$(B,F_2,1,T//4)$	Linear	$bias = False$
	BatchNorm					$2 * F_2$	$(B,F_2,1,T//4)$		
	Activation						$(B,F_2,1,T//4)$	ELU	
	2D average pooling		$(1,8)$	$(1,8)$			$(B,F_2,1,T//32)$		
	Dropout						$(B,F_2,1,T//32)$		$p = 0.25$ or $p = 0.5$
	Flatten						$(B,F_2 * (T//32))$		
3	2D convolution	n	$(T,1)$	1	$(0,0)$	$((F_2 * (T//32)) + 1) * n$	(B,n)		$bias = True$

Table 1: Architecture of the EEGNet, abbreviations used are:
B =batch size, C =number of input channels, T =number of time points, T_f =half of the sampling frequency, F_1 =number of temporal filters,D =number of spatial filters per temporal filter, F_2 =number of pointwise filters, N_c =number of classes
The corresponding values that were used for this study were:
$B = 32$, $C = 1$, $T = 2.1 * fs = 252$ where $fs = 120$ indicates the sampling frequency, $T_f = 0.5 * fs = 60$, $F_1 = 96$, $D = 1$, $F_2 = 96$, $N_c = 20$. For the dropout rate p, a value of 0.5 was used when training on empirical data, and a value of 0.25 was used when training on simulated data

4.4 Forward model

The inverse models can be used to classify data. However, they typically require empirical calibration data before being applicable in practice. Due to the high cost of BCI experiments, recording such empirical data is not desirable. Therefore, a forward model was implemented which can simulate data that could potentially replace the required empirical calibration data.

The forward model that was was used in this study is based on the summation of individual noise and signal components. A simplified version of the inner workings of the model can be seen in the following formula:

$$\mathbf{X} = \mathbf{S} + \mathbf{N}, \tag{6}$$

where $\mathbf{X} \in \mathbb{R}^{c,m}$ is a single simulated trial with c channels and m samples. $\mathbf{S} \in \mathbb{R}^{c,m}$ is the signal component, and $\mathbf{N} \in \mathbb{R}^{c,m}$ is the noise component. As indicated, the formula shows a simplified version of the forward model. More specifically, the combination of the signal and noise components is not simply linear addition. Instead, this combination is dependent on factors such as the SNR. See Section 4.4.3 for more details.

In the various components of the forward model, frequent use is made of a truncated normal distribution, which will therefore be introduced in advance. The choice to use a truncated normal distribution rather than a standard normal distribution was made in order to ensure that the sampled values of the parameters remain within a range that is deemed to be realistic. The formula for this truncated normal distribution is as follows:

$$\psi(a, b, \mu, \sigma^2), \tag{7}$$

where ψ refers to the truncated normal distribution itself, a and b determine the truncated interval, μ is the mean, and σ^2 is the variance. The variance can be calculated as follows:

$$\sigma^2 = \min(|r_1 - \mu|, |r_2 - \mu|)/2, \tag{8}$$

where r_1 and r_2 are respectively the intended lower bound and upper bound of the distribution. Finally, the truncated interval can be determined based on the formulae:

$$a = (r_1 - \mu)/\sigma^2, \tag{9}$$

$$b = (r_2 - \mu)/\sigma^2. \tag{10}$$

4.4.1 Signal modeling

For the signal aspect of the forward model, three separate phases can be distinguished. The first phase consists of generating a response pair, containing a response to a short event, as well as a response to a long event. This phase was implemented in multiple ways, of which three will be discussed here. For details regarding other attempted implementations we refer to the appendix. The first approach will henceforth be referred to as the CCA version of signal modeling. This approach consisted of applying the CCA inverse model to the empirical data of indidivual participants and splitting the recorded response vectors into responses to each individual event type. This led to 30 response pairs, all of which contained 0.3 seconds of data for both the responses to short events and the responses to long events. The generation of the responses was subsequently performed by randomly choosing 1 of these 30 response pairs. This approach to signal modeling is most closely linked to the specific empirical data that were used in this study. Therefore, when testing on this specific data this approach would be expected to perform relatively well, whereas a lower capacity for generalization to other data would be expected.

In the second approach, henceforth referred to as the Gamma approach, the response pairs were modeled using a quadruple-Gamma function, which was based on the double-Gamma HRF function (Friston et al., 1998). This quadruple-Gamma function is defined as follows:

$$S(t) = c_1 * \frac{t^{\alpha_1}\beta_1^{\alpha_1}e^{-\beta_1 t}}{\Gamma(\alpha_1)} - c_2 * \frac{t^{\alpha_2}\beta_2^{\alpha_2}e^{-\beta_2 t}}{\Gamma(\alpha_2)} - c_3 * \frac{t^{\alpha_3}\beta_3^{\alpha_3}e^{-\beta_3 t}}{\Gamma(\alpha_3)} - c_4 * \frac{t^{\alpha_4}\beta_4^{\alpha_4}e^{-\beta_4 t}}{\Gamma(\alpha_4)}, \tag{11}$$

where $S(t)$ is the value of the response at time t, the α_i values are related to the time of each peak, the β_i values have an effect on both the time and amplitude of the peaks, and lastly the c_i values mostly influence the amplitudes of the 4 peaks. The first 3 of these peaks model the most prominent peaks in the VEP responses. The fourth of the peaks is used to ensure that the value of the modeled signal returns to zero. In order to generate the required pairs of responses, the optimal values for the α_i and c_i parameters were estimated, while each of the β_i values was set to a value of 1. For the estimation of the optimal values, each individual response in the response pairs that were extracted using CCA was used as a ground truth. The mean squared error (MSE) between the ground truth and a simulation using the quadruple-Gamma function was used as the objective function. This resulted in 30 new response pairs, where each individual response was simulated using its corresponding set of estimated optimal parameter values. The final responses that were generated by this approach were randomly sampled from this new set of 30 response pairs. Similar to the CCA approach, this Gamma approach would be expected to perform relatively well on the specific empirical data that were used in this study. However, its capacity for generalization to other data would be expected to be relatively low.

The third and final approach for simulating the response pairs, henceforth referred to as the Sig approach, made use of a different mathematical model. In this model, four individual sigmoids were simulated and combined into a full response. This model is as follows:

$$S(t) = \begin{cases} l_0 + \frac{l_1 - l_0}{1+e^{-k_0*((t-t_0)-(t_0^0-t_0))}}, & \text{if } t_0 <= t <= t_1 \\ l_1 + \frac{l_2 - l_1}{1+e^{-k_1*((t-t_1)-(t_1^0-t_1))}}, & \text{if } t_1 < t <= t_2 \\ l_2 + \frac{l_3 - l_2}{1+e^{-k_2*((t-t_2)-(t_2^0-t_2))}}, & \text{if } t_2 < t <= t_3 \\ l_3 + \frac{l_4 - l_3}{1+e^{-k_3*((t-t_3)-(t_3^0-t_3))}}, & \text{if } t_3 < t <= t_4 \\ 0, & \text{otherwise} \end{cases}, \quad (12)$$

where l_i is the amplitude of peak i, k_i is the logistic growth rate for sigmoid i out of the four individual modeled sigmoids. t_i^0 is the time of the midpoint of sigmoid i, and t_i is the time of peak i. The values of t_0 and t_4 were used to force the simulated signals to start and end at a value of 0. In total, this simulation approach requires 18 parameters. These are the five t_i values, the corresponding five l_i values, the four k_i values, and the four t_i^0 values. The first four aspects of the equation are used to model respectively the shape of the signal from the onset until the first peak, from the first to the second peak, from the second to the third peak, and from the third peak to the end of the signal. Here, the three peaks that are indicated correspond to the most prominent peaks in the VEP responses, which were also used previously in the Gamma approach.

In this study, the values of the k_i parameters were all set to a value of 1.0. In addition, the values for the t_0 and t_4 parameters were set to 0.0 and $0.3 - t_3 - 1\text{e-}5$ respectively and their corresponding l_0 and l_4 values were set to 0.0. For all other parameters, truncated Gaussian distributions were created. The exact bounds and means for these distributions were chosen based on visual inspection of the 30 response pairs obtained by CCA. This yielded a total of 20 distributions, of which 10 were used for modeling the responses to short events, and the other 10 for modeling the responses to long events. For every simulated trial, the parameter values were drawn anew from these distributions. These values were subsequently used in Equation 12 to generate the required response pairs. An example of both a simulated response to short events and a simulated response to long events can be found in Figure 2. This Sig approach would, contrary to the CCA and Gamma approaches, be expected to have a higher capacity of generalization to other datasets. However, due to the reduced link to the empirical data used in this study, it would be expected to perform relatively worse on this specific dataset.

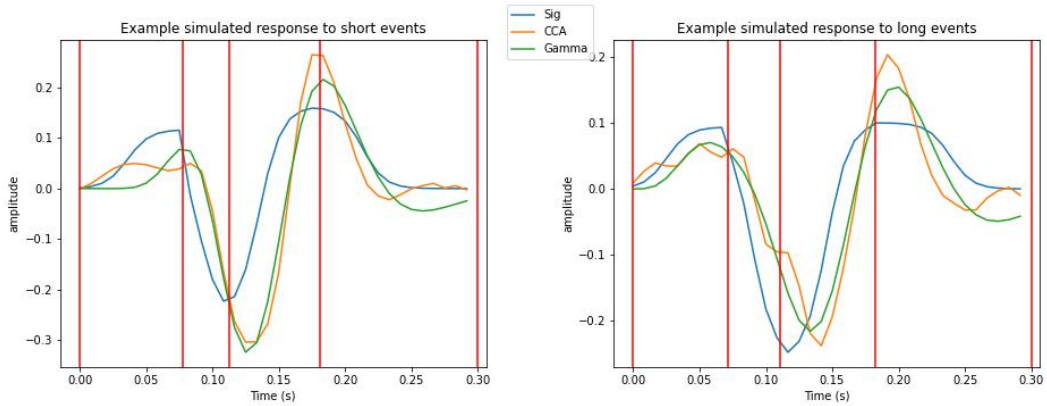

Figure 2: Example simulations for the responses to a short and a long events. The vertical lines indicate, in order, the values of the t_0 up to t_4 parameters for the Sig model. For comparison, example simulations made by the CCA and Gamma approaches are provided as well.

After the simulation of the response pairs, the second phase of the signal modeling consisted of randomly sampling one of the 20 2.1 seconds long codes that was used in the measurement of the empirical data. This code was subsequently repeated a pre-determined amount of times to simulate a trial with repeated codes. For this study, this amount was set to 15 due to this being the amount of repetitions used during the collection of the empirical data, yielding a total code lasting 31.5 seconds.

Finally, the third phase consists of combining the simulated response pairs with the sampled code. This is performed using linear addition, where the response at any point in time is the linear summation of the individual responses that are active at that time. This approach was based on the linear superposition hypothesis (see e.g. Notbohm, Kurths, & Herrmann (2016) and Capilla, Pazo-Alvarez,Darriba,Campo, & Gross (2011)). A mathematical representation of this summation process can be seen in the following equation:

$$\mathbf{S} = \mathbf{r}^T \mathbf{M}, \tag{13}$$

where $\mathbf{S} \in \mathbb{R}^m$ is the response to a sequence of events, $\mathbf{r} \in \mathbb{R}^{m'}$ is the response to a single event, with a length of m' samples, and $\mathbf{M} \in \mathbb{R}^{m',m}$ is the structure matrix for the sequence of events.

4.4.2 Noise modeling

The noise aspect $\mathbf{N}$ of the forward model was modeled as a weighted sum of four individual noise components which were divided by their standard deviations, multiplied by a constant value. This constant value is used to ensure that the resulting simulated noise has a range that is similar to the noise that was recorded in the stimulation-free inter-trial intervals within the empirical data. The full formula to calculate the noise aspect of the forward model is as follows:

$$\mathbf{N} = c_n * \sum_{i=1}^{4} \frac{\mathbf{N}_i}{\sigma(\mathbf{n}_i)} z_i, \tag{14}$$

where c_n is a constant value by which the weighted sum is multiplied, $\mathbf{N}_i \in \mathbb{R}^{c,m}$ indicates the noise simulated by noise component i, z_i indicates the weight value associated with said noise component, and $\sigma(x)$ denotes the standard deviation of x. Examples of the time and frequency domains of all of the four noise components can be seen in Figure 3.

The first of the four implemented noise components was pink noise, representing the noise that follows a $1/f$ frequency spectrum and is typically found in EEG data. This component was implemented using Gaussian power law noise. The implementation that was used to generate this noise was based on an algorithm proposed by Timmer & Koening (1995). This algorithm is described in Algorithm 1.

Algorithm 1 Generating power law noise

1: Choose a power law spectrum $\mathbf{N}_1(f) \sim (1/f)^{\beta}$
2: **for all** Fourier frequency ω_i **do**
3: **if** the number of data points is odd **then**
4: draw 2 Gaussian distributed random numbers
5: Multiply the numbers by $\sqrt{\frac{1}{2}\mathbf{N}_1(\omega_i)} \sim (1/\omega)^{\beta/2}$ ▷ The result is used as the real and imaginary parts of the Fourier transform of the desired data
6: **else if** the number of data points is odd **then**
7: draw 1 Gaussian distributed random number ▷ Due to symmetry, $f(\omega_{Nyquist})$ is always real. Therefore only 1 value is drawn
8: Multiply the number by $\sqrt{\frac{1}{2}\mathbf{N}_1(\omega_i)} \sim (1/\omega)^{\beta/2}$
9: **end if**
10: **end for**
11: Obtain a real valued time series by choosing the Fourier components for the negative frequencies according to $f(-\omega_i) = f(\omega_i)^*$, with * indicating complex conjugation
12: Use backward Fourier transformation of $f(\omega)$, from the frequency domain to the time domain, resulting in the final time series.

The second noise component was used to model white noise, representing the measurement noise in EEG recordings. This component was created by sampling random noise from a Gaussian distribution with zero mean and unit variance. The formula for this component, where $\mathbf{N}_2 \in \mathbb{R}^{c,m}$ is the white noise, and $\mathcal{N}$ indicates a Gaussian distribution, would be as follows:

$$\mathbf{N}_2 \sim \mathcal{N}(0,1) \tag{15}$$

The third noise component represented visual alpha activity. In order to obtain this activity, white Gaussian noise was generated. Subsequently, this noise was band-pass filtered at 8.5 Hz

to 12 Hz using a third-order Butterworth filter with a 120 Hz sampling frequency. The result of this filtering was used as the generated alpha activity. This approach to generating alpha activity was similar to the approach used in the EEGSourceSim framework (Barzegaran et al., 2019). The corresponding formula for this component, where $\mathbf{N}_3 \in \mathbb{R}^{c,m}$ is the resulting alpha activity, and $g(x)$ is the representation of x after being filtered using the Butterworth filter described earlier, can be seen in the following equation:

$$\mathbf{N}_3 \sim g(\mathcal{N}(0,1)) \tag{16}$$

The last noise type that was implemented was that of 50 Hz line noise. Each sample in the simulation of this noise was calculated using a sine function, given the sampling frequency and the intended noise frequency. The calculation can be seen in the following formula:

$$\mathbf{N}_4^i = \sin(2\pi * f_l * (i / f_s)) \tag{17}$$

where $\mathbf{N}_4^i \in \mathbb{R}^1$ is the value of the line noise at sample i, f_l is the intended noise frequency, being 50 Hz in this case, and f_s is the sampling frequency, which in this case amounts to 120 Hz.

In total, the implementation of the noise contains 6 parameters that can be freely varied. These include the β exponent for the pink noise, the 4 weights for the individual noise components, and lastly the c_n value by which the noise is multiplied. Using these parameters, two versions of the noise model were implemented and tested. The first version used pre-defined values for each of the 6 parameters for each simulation that was performed. These values were β=1.0, z_1=$\frac{18}{40}$, z_2=$\frac{2}{40}$, z_3=$\frac{14}{40}$, z_4=$\frac{3}{40}$, c_n=2e-5. An example of the simulated combined noise, using the pre-defined parameter values, as well as the same example after pre-processing, can be seen in Figure 4. In order to allow for comparison, an example of an extracted part of empirical noise, lasting for 1.0 seconds, is provided in Figure 5.

The second version of the noise modeling that was implemented used probability distributions from which to sample the parameter values. In order to implement this version, noise data was extracted from the empirical data, by extracting only the samples that were recorded in between trials, i.e. the stimulation-free inter-trial intervals. Subsequently, for each extracted part of noise data, the optimal values for the 6 parameters were estimated, using the MSE between simulated noise and each extracted part of the noise data as the objective function. Subsequently, for each parameter, Gaussian distributions were fit to the corresponding set of estimated optimal parameter values. During the generation of noise data, each of the 6 parameter values was drawn from its corresponding distribution for every simulated trial. The distributions themselves can be seen in Figure 6.

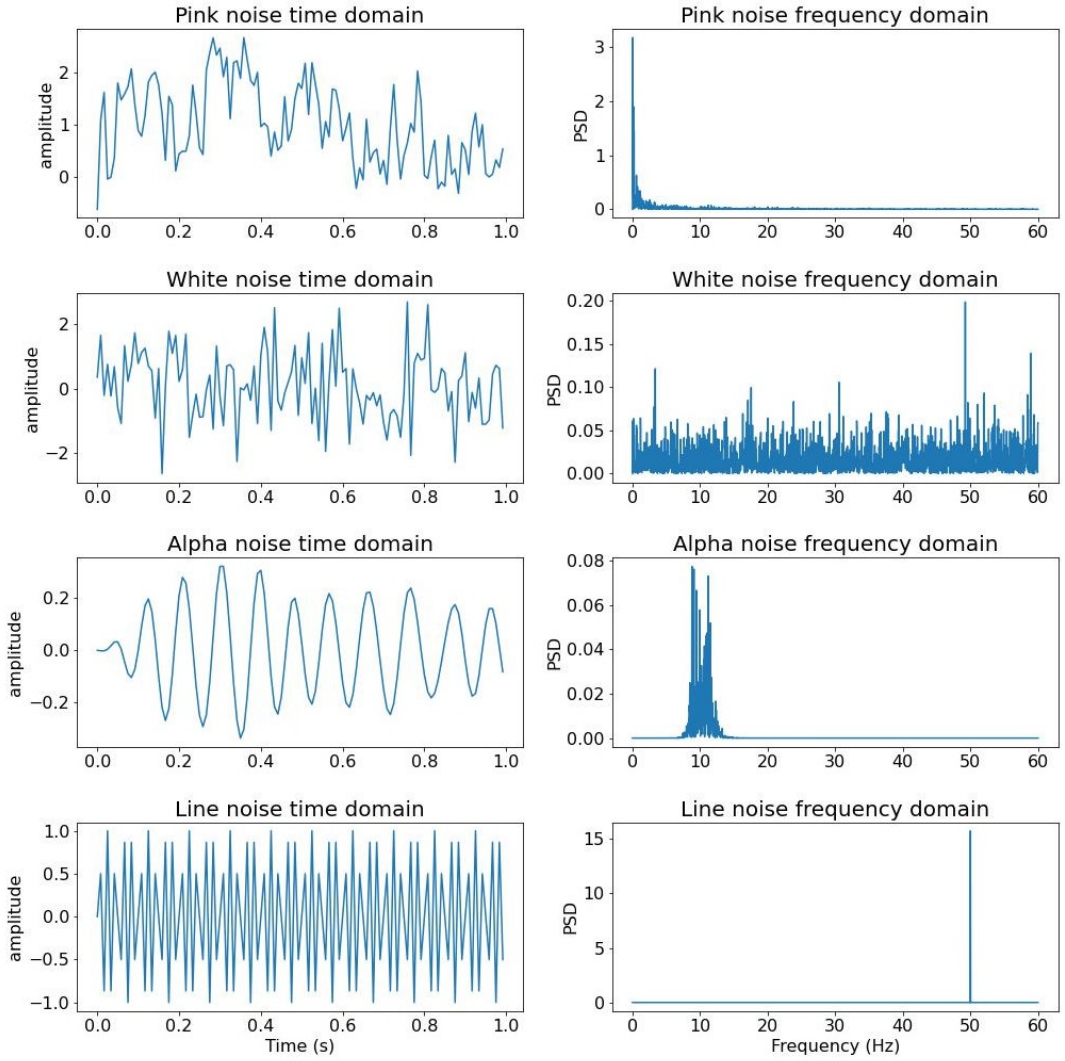

Figure 3: Example simulations of each of the four individual noise components, given a sampling frequency of 120 Hz. **Left column:** time domains for 1 second of simulated data for each noise component. **Right column:** Estimated frequency domains for each of the four noise components, estimated using 31.5 seconds of simulated noise data.

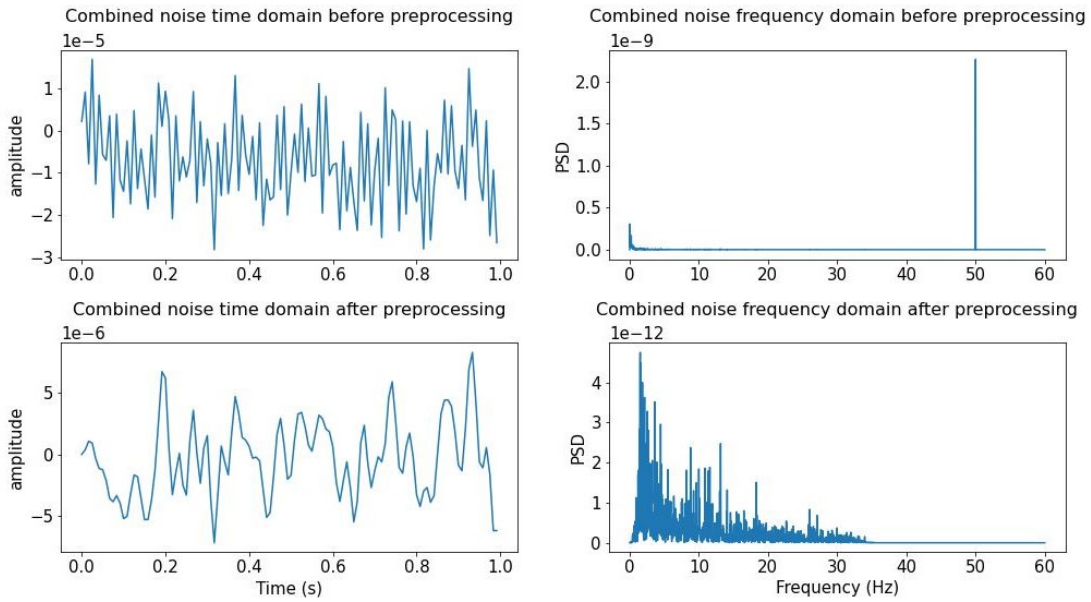

Figure 4: An example simulation of the combined noise using pre-defined parameter values, as well as its pre-processed version, given a sampling frequency of 120 Hz. **Left column:** time domains for the first 1 second of 31.5 seconds of simulated data for both the example simulation and its pre-processed version. **Right column:** Estimated frequency domains for both the example simulation and its pre-processed version, estimated using 31.5 seconds of data.

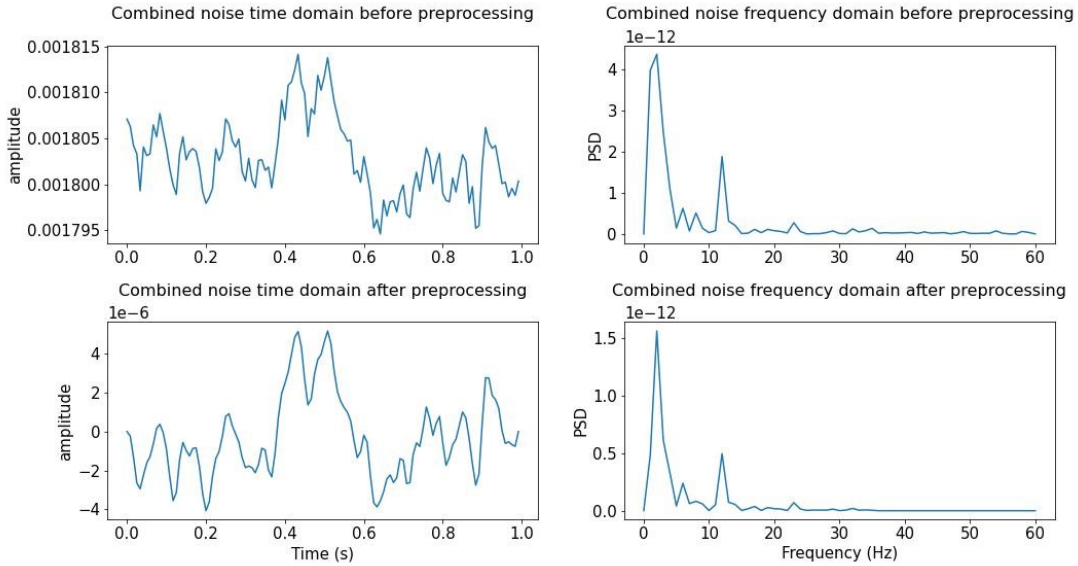

Figure 5: An example of empirical noise, as well as its pre-processed version, given a sampling frequency of 120 Hz. **Left column:** time domains for the full 1 second of extracted empirical data for both the example itself and its pre-processed version. **Right column:** Estimated frequency domains for both the example itself and its pre-processed version, estimated using 1 second of data.

4.4.3 Combining signal and noise

After simulating the signal and noise components, they are combined to form the final output of the forward model. Each of the two components is first divided by its standard deviation, in order to normalize the scales of the two components. After this division, an addition of the two resulting variables is performed. Lastly, the variable resulting from this addition is multiplied by a constant, in order to adapt the range to be similar to the range of the empirical data. This combination can be seen in the following equation:

$$\mathbf{X} = (\frac{\mathbf{S}}{\sigma(\mathbf{S})} * \upsilon_{snr} + \frac{\mathbf{N}}{\sigma(\mathbf{N})}) * c_c, \tag{18}$$

where υ_{snr} indicates the SNR value for the trial that is being simulated, $\sigma(x)$ indicates the standard deviation of x, and c_c is a constant that is used to adapt the range of the data. This constant was set to a value of 2e-5, as this value was found to yield good results. In this study, the exact value of the constant was not explicitly investigated, hence consistently using the value of 2e-5 is not guaranteed to be optimal.

The SNR values that were used in combining the signal and noise components were drawn from a truncated Gaussian distribution. For drawing the SNR values in this study, a distribution was chosen with a mean of μ=0.68, a minimum bound of r_1=0.5 and a maximum bound

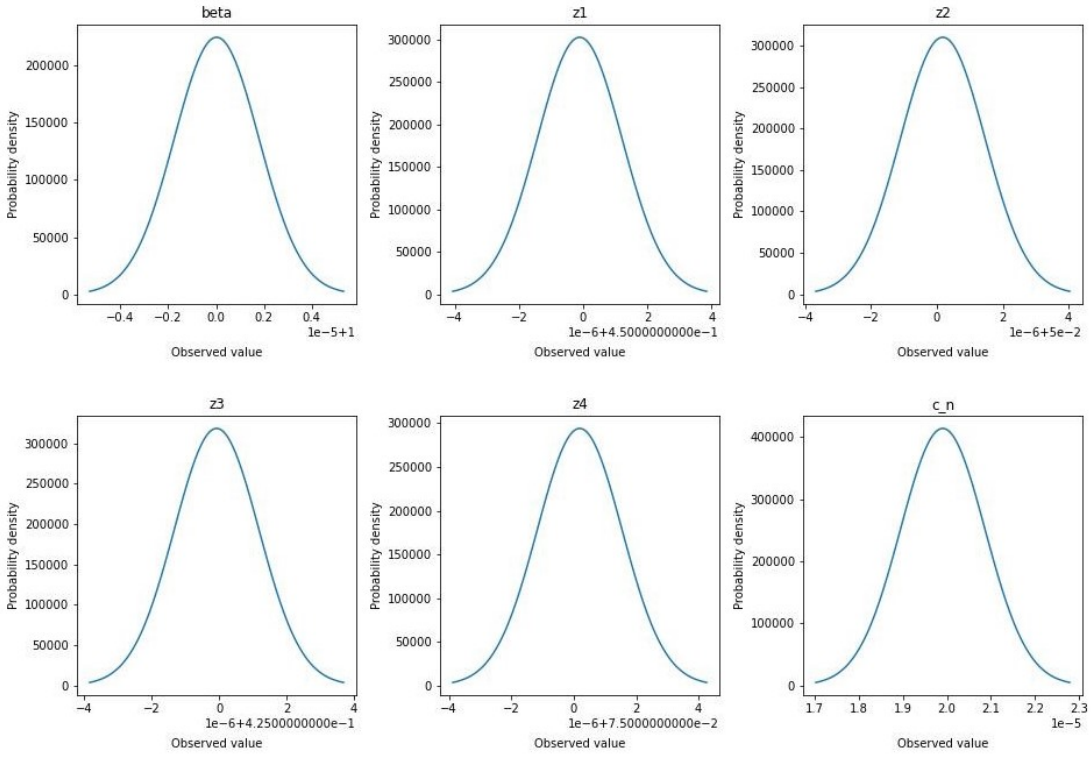

Figure 6: The probability density functions for each of the 6 noise parameters.

of r_2=1.0. This particular mean value was chosen due to it having shown strong results in the initial phases of the study, whereas the bound values were chosen arbitrarily. In the use of the truncated Gaussian distribution, the calculation of the σ^2 was adapted for the distribution for the SNR values, which can be seen in the following formula:

$$\sigma^2 = \max(|r_1 - \mu|, |r_2 - \mu|)/2 \tag{19}$$

As can be seen, the adaptation used the calculation of the maximum, rather than the minimum. By using the maximum, the width of the distribution increase, leading to a higher likelihood for values in the vicinity of the bound farthest from the mean, as compared to their likelihood when using the minimum function. This increased likelihood was deemed to be desirable, in order to generate a more diverse range of values, hence the adaptation was performed.

Finally, the formula for the υ_{snr} was as follows:

$$\upsilon_{snr} \sim \psi(a, b, 0.68, 0.16) \tag{20}$$

4.5 Analysis

For each of the two inverse models, three types of analyses were performed. The first of these was performed by training and testing the inverse models on empirical data, and will be referred to as the emp>emp analysis. This analysis is used as the baseline performance, to which the results of other analyses are compared. This analysis was performed using a within-subject design, where the performance for each subject was individually analyzed using 5-fold chronological cross-validation. For the analysis using EEGNet, the training data for each fold was further split into 80% training data and 20% validation data. This split was treated as a necessary step for the training of the EEGNet, whereas the CCA model does not require such a step. As a result, no such split was performed for the CCA model.

The second type of analysis was performed by training and testing on data that was simulated by the forward model, and will be referred to as the sim>sim analysis. This analysis is performed to provide an indication of whether the simulated data can be learned by the inverse models in the first place. For this analysis, 1500 training samples and 300 testing samples with a length of 31.5 seconds were simulated. For the EEGNet analysis, an additional 300 samples of validation data were generated.

The third and final type of analysis was performed by training on simulated data and testing on empirical data, and will be referred to as the sim>emp analysis. This analysis is most related to the aim of this study, being to create a forward model of which the data can be used to train inverse models for application on empirical data. For this analysis, 1500 training samples of 31.5 seconds were generated, with 300 additional validation samples for training the EEGNet model. The empirical data on which the trained models were tested was split into folds that were identical to those used for the emp>emp analysis type, i.e. 5 sets of data per participant. This was done to ensure that the obtained results of the sim>emp analysis could be directly compared to those of the emp>emp analysis.

For all of the simulated data that was used in the analyses, one additional step was performed before the data was used to train either the CCA method or the EEGNet. In this step, the simulated data was preprocessed using FIR spectral band-pass filtering between 2 and 30 Hz. This preprocessing is similar to the preprocessing that was applied on the empirical data.

For each of the three types of analyses, the inverse models were trained to predict the class of 2.1 seconds of data, obtained by splitting the trials of 31.5 seconds into 15 parts. The sim>sim and sim>emp analysis types were both executed multiple times, once for each implemented version of the forward model. In total, four versions of the forward model were implemented and evaluated, which differed only in the versions of signal and noise simulation that were used. The names of the models follow the format: signal modeling version-noise modeling version. For the signal modeling version, CCA refers to the approach using the responses obtained with the CCA inverse model, Gamma refers to the approach using the quadruple-Gamma function, and Sig refers to the approach where parameters are drawn from distributions and modeled using four sigmoid components. For the noise modeling version, Frozen refers to the noise implementation using static pre-defined values, and Drawn refers

Model name	signal implementation version	noise implementation version
CCA-Frozen	responses obtained using CCA	Parameter values pre-defined
CCA-Drawn	responses obtained using CCA	Parameter values drawn from distributions
Gamma-Drawn	responses modeled using quadruple-Gamma	Parameter values drawn from distributions
Sig-Drawn	responses modeled using sigmoids	Parameter values drawn from distributions

Table 2: Details regarding each of the four tested forward model versions, in terms of their implemented signal simulation version and noise simulation version.

to the implementation where noise parameter values are drawn from distribution. Using this naming convention, the four models that were implemented were CCA-Frozen, CCA-Drawn, Gamma-Drawn and Sig-Drawn. The details of the models are provided in Table 2.

In total, 18 sets of accuracy scores were obtained during this study, of which 9 were obtained for each of the inverse models. These 9 were sub-divided into 1 set obtained during the emp>emp analysis, and 4 sets for both the sim>sim and sim>emp analyses, corresponding to the 4 used versions of the forward model. The results obtained during the sim>emp analysis were compared to the results obtained with the corresponding inverse model during the emp>emp analysis. In addition, the results obtained during the sim>emp analysis were compared to other results of the sim>emp analysis which were obtained using a different version of the forward model. Finally, the results of EEGNet were compared to those obtained using CCA. These comparisons were performed using one-sided Bonferroni corrected Wilcoxon signed-rank tests.

5 Results

In this study, accuracies were obtained for each type of analysis (emp>emp, sim>emp, sim>sim), each type of forward model (CCA-Frozen, CCA-Drawn, Gamma-Drawn, Sig-Drawn), and each inverse model (CCA, EEGNet). The mean values of each of these sets, as well as the corresponding standard deviations, can be found in Table 3 and Figure 7. The standard deviations for the emp>emp and sim>emp analyses were calculated by first calculating the per-subject mean accuracies, followed by calculating the standard deviation of these per-subject mean accuracies.

	emp>emp	sim>emp				sim>sim			
Inverse model		CCA-Frozen	CCA-Drawn	Gamma-Drawn	Sig-Drawn	CCA-Frozen	CCA-Drawn	Gamma-Drawn	Sig-Drawn
CCA	62.04±23.47	45.26±22.63	45.23±22.64	42.46±21.51	38.07±20.60	97.28±0.15	97.40±0.25	98.61±0.12	72.09±1.11
EEGNet	47.36±28.09	49.33±24.61	48.56±24.05	44.59±23.45	39.33±22.20	98.55±0.07	98.56±0.15	98.89±0.14	95.04±0.46

Table 3: Mean accuracies over all 30 participants and the corresponding standard deviations for each type of analysis (emp>emp, sim>emp, sim>sim, see Section 4.5), each type of forward model (CCA-Frozen, CCA-Drawn, Gamma-Drawn, Sig-Drawn, see Section 4.4), and each inverse model (CCA, EEGNet, see Section 4.3).

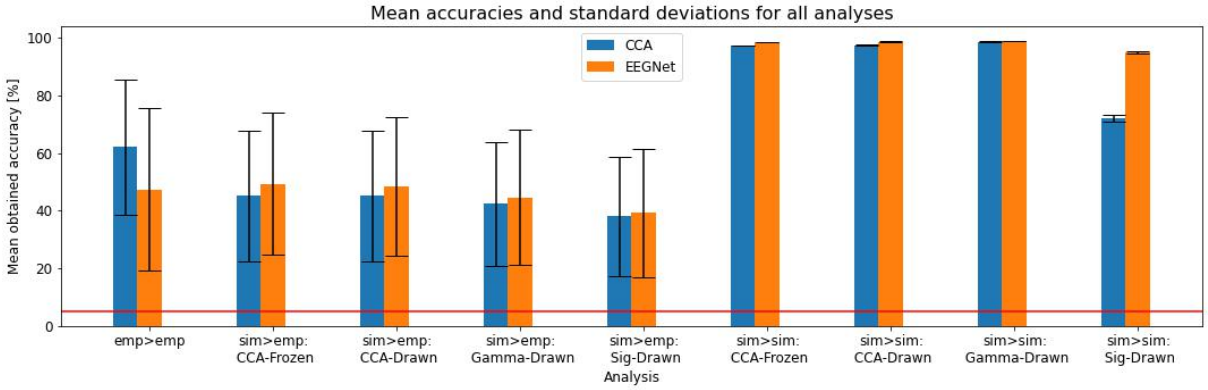

Figure 7: A visualization of the mean accuracies over all 30 participants and the corresponding standard deviations for each type of analysis (emp>emp, sim>emp, sim>sim, see Section 4.5), each type of forward model (CCA-Frozen, CCA-Drawn, Gamma-Drawn, Sig-Drawn, see Section 4.4), and each inverse model (CCA, EEGNet, see Section 4.3). The horizontal red line at 5% indicates chance level.

5.1 Results obtained using the CCA inverse model

The accuracies obtained for all analyses where the CCA inverse model was used and where testing was performed on empirical data were compared using one-sided Wilcoxon signed-rank tests. The p-values of these tests can be found in Table 4. These analyses can provide further insights into how the various forward model versions compare to each other and to empirical data.

It was found that the accuracies obtained by the emp>emp analysis (62.04%) were significantly higher (P<.001, Wilcoxon signed-rank test) than the accuracies obtained by any of the sim>emp analysis (45.26%,45.23%,42.46% and 38.07%). The accuracies obtained in the sim>emp analysis using the CCA-Frozen model (45.26%) were found to be significantly higher (P<.001, Wilcoxon signed-rank test) than those obtained with the Gamma-Drawn and Sig-Drawn models (42.46% and 38.07%) , while not being significantly higher (P>.05, Wilcoxon signed-rank test) than the accuracies obtained with the CCA-Drawn model (45.23%) or those obtained in the emp>emp analysis (62.04%).

The accuracies that were obtained in the sim>emp analysis using the CCA-Drawn model (45.23%) were found to be significantly higher (P<.001, Wilcoxon signed-rank test) than those obtained using the Gamma-Drawn and Sig-Drawn models (42.46% and 38.07%) , while not being significantly higher (P>.05, Wilcoxon signed-rank test) than the results obtained using either of the other two analyses (62.04% and 45.26%). Lastly, the accuracies that were obtained in the sim>emp analysis using the Gamma-Drawn model (42.46%) were found to be significantly higher (P<0.001, Wilcoxon signed-rank test) than those obtained using the Sig-Drawn model (38.07%).

In addition, the sim>sim analysis was performed for each of the four versions of the forward

Analysis	emp>emp	sim>emp: CCA-Frozen	sim>emp: CCA-Drawn	sim>emp: Gamma-Drawn	sim>emp: Sig-Drawn
emp>emp	-	P<.001	P<.001	P<.001	P<.001
sim>emp: CCA-Frozen	P=1.000	-	P=1.000	P<.001	P<.001
sim>emp: CCA-Drawn	P=1.000	P=1.000	-	P<.001	P<.001
sim>emp: Gamma-Drawn	P=1.000	P=1.000	P=1.000	-	P<.001
sim>emp: Sig-Drawn	P=1.000	P=1.000	P=1.000	P=1.000	-

Table 4: The p-values for the one-sided Wilcoxon signed-rank tests, applied on the analyses with the CCA inverse model. To provide an example for how this table should be read, we look at the top-right cell. Where the p-value of P<.001 indicates that the results obtained in the emp>emp analysis were significantly higher than those obtained in the sim>emp analysis with the Sig-Drawn version of the forward model.

model. The mean accuracies obtained in this analysis using the CCA-Frozen, CCA-Drawn and Gamma-Drawn models all exceeded 95%. This indicates that the CCA model is able to learn and predict the classes of the simulated data reasonably well. For the simulation performed using the Sig-Drawn model, the mean accuracy decreased to a value of 72.09%. This indicates that, although the CCA model is still able to learn and predict the classes of the corresponding simulated data to a certain degree, it is less capable of doing so for data simulated using the Sig-Drawn forward model version as compared data simulated using any of the other three forward model versions.

5.2 Results obtained using the EEGNet inverse model

The performances of all analyses that used the EEGNet inverse model and were tested on empirical data were compared using one-sided Wilcoxon signed-rank tests. The p-values resulting from said tests can be seen in Table 5. These analyses can provide further insights into how the various forward model versions compare to each other and to empirical data.

Analysis	emp>emp	sim>emp: CCA-Frozen	sim>emp: CCA-Drawn	sim>emp: Gamma-Drawn	sim>emp: Sig-Drawn
emp>emp	-	P=1.000	P=1.000	P=.232	P<.001
sim>emp: CCA-Frozen	P=.480	-	P=.670	P<.001	P<.001
sim>emp: CCA-Drawn	P=1.000	P=1.000	-	P<.001	P<.001
sim>emp: Gamma-Drawn	P=1.000	P=1.000	P=1.000	-	P<.001
sim>emp: Sig-Drawn	P=1.000	P=1.000	P=1.000	P=1.000	-

Table 5: The p-values for the one-sided Wilcoxon signed-rank tests, applied on the analyses with the CCA inverse model. To provide an example for how this table should be read, we look at the top-right cell. Where the p-value of P<.001 indicates that the results obtained in the emp>emp analysis were significantly higher than those obtained in the sim>emp analysis with the Sig-Drawn version of the forward model.

In the analyses that were performed using the EEGNet inverse model, it was found that the accuracies obtained in the emp>emp analysis (47.36%) were significantly higher (P<.001, Wilcoxon signed-rank test) than those obtained in the sim>emp analysis using the Sig-Drawn version of the forward model (39.33%). However, the accuracies obtained in the emp>emp analysis (47.37%) were not found to be significantly higher (P>.05, Wilcoxon signed-rank test) than the accuracies obtained using any of the other three analyses (49.33%,48.56% and 44.59%) . The accuracies that were obtained in the sim>emp analysis using the CCA-Frozen model (49.33%) were in turn found to not to be significantly higher (P>.05, Wilcoxon signed-rank test) than the results of the emp>emp analysis (47.36%). The accuracies that were obtained in the sim>emp analysis using the CCA-Frozen model (49.33%) were additionally found not to be significantly higher than those obtained using the CCA-Drawn model (48.56%), while being significantly higher (P<.001, Wilcoxon signed-rank test) than those obtained in the sim>emp analysis using any of the other forward model versions (44.59% and 39.33%).

The results that were obtained in the sim>emp analysis using the CCA-Drawn model (48.56%) were found not to be significantly higher (P>.05, Wilcoxon signed-rank test) than those obtained in the in either the emp>emp analysis (47.36%) or those obtained using the CCA-Frozen model (49.33%). In addition these results using the CCA-Drawn model (48.56%) were found to be significantly higher (P<.001, Wilcoxon signed-rank test) than those obtained using the Gamma-Drawn and Sig-Drawn analyses (44.59% and 39.33%). The accuracies that were obtained using the Gamma-Drawn model in the sim>emp analysis (44.59%) were found to be significantly higher (P<.001, Wilcoxon signed-rank test) than those obtained using the

Sig-Drawn model (39.33%). However, these accuracies were not found to be significantly higher (P>.05, Wilcoxon signed-rank test) than those obtained in the emp>emp analysis (47.36%).

In addition to the analyses with the Wilcoxon signed-rank test, the sim>sim analysis was performed. In this analysis, it was found that the accuracies obtained using the EEGNet inverse model and any of the forward model versions all exceeded 95%. This can be taken as an indication that it is indeed possibly for the EEGNet inverse model to learn and predict the classes of simulated data to a reasonable degree.

5.3 Comparing CCA to EEGNet

The final part of the results that were obtained in this study regards the comparison between the two used inverse model versions. These comparisons were performed for each of the five analyses that were performed by testing on empirical data, and the comparisons were executed using one-sided Wilcoxon signed-rank tests. The p-values of said tests can be found in Table 6. These analyses can provide further insights into how the inverse models compare to each other under various experimental conditions.

Analysis	emp>emp	sim>emp			
		CCA-Frozen	CCA-Drawn	Gamma-Drawn	Sig-Drawn
CCA>EEGNet	P<0.001	P=1.000	P=1.000	P=1.000	P=1.000
EEGNet>CCA: CCA-Frozen	P=1.000	P<0.001	P<0.001	P<0.001	P<0.001

Table 6: The p-values for the Wilcoxon signed-rank tests, applied on the analyses with the EEGNet inverse model. To provide an example for how this able should be read, we look at the top-right cell. Where the p-value of P=1.000 indicates that the results obtained in the sim>emp analysis with the Sig-Drawn version of the forward model and the CCA inverse model were not significantly higher than those obtained in the sim>emp analysis with the Sig-Drawn version of the forward model and the EEGNet inverse model.

In the table, it can be seen that the accuracies obtained in the emp>emp analysis using the CCA inverse model (62.04%)were found to be signficantly higher (P<.001, Wilcoxon signed-rank test) than those obtained using the EEGNet inverse model (47.36%). For all other analyses, this was not the case (P>.05, Wilcoxon signed-rank test). Conversely, it was found that the results obtained using the EEGNet inverse model in any of the sim>emp analyses (respectively 49.33%,48.56%,44.59% and 39.33%) were significantly higher (P<.001, Wilcoxon signed-rank test) than those obtained using the CCA inverse model (respectively 45.26%,45.23%,42.46% and 38.07%).

6 Discussion

6.1 Relations among the forward model versions and the inverse models

In this study, various analyses were performed, which can be divided into emp>emp, sim>emp and sim>sim analyses. The accuracies that were obtained in each of these analyses are dependent on the inverse model that was used. The results of the sim>emp and sim>sim analyses were additionally dependent on the version of the forward model that was used.

6.1.1 Relations between the inverse models

In the analyses, it was found that the CCA inverse model performed significantly better than the EEGNet inverse model in all the emp>emp analysis, while performing significantly worse for each of the sim>emp analyses. Furthermore, the accuracies obtained in the sim>sim analyses indicate that the inverse models were able to learn the characteristics of the simulated data. Given that any inability to learn simulated data is thus unlikely to be the cause, we hypothesize the difference in performance to have been caused by the differences in nature of the two inverse models in combination with the differences in nature between the training data for the emp>emp analysis and the training data for the sim>emp analysis. For the emp>emp analysis, each inverse model was trained for multiple folds for each participant individually, thus only being trained and tested on a small set of within-subject data each time. As a result, the training data would likely have contained less variance as compared to cross-subject data. This relatively low degree of variance is a favourable condition for the CCA inverse model. However, as a machine learning technique with a relatively high model capacity, the EEGNet that was used in this study would be at risk of overfitting on the training data to a certain degree due to the small variance and small amount of data, leading to a reduced performance on the testing data. This risk of overfitting was visible in the training phase of the EEGNet architecture in the emp>emp analysis, where the architecture performed better on training data than on validation data. This is hypothesized to have been the cause of the accuracies obtained in the emp>emp analysis using the CCA inverse model being significantly higher than those obtained using the EEGNet inverse model.

Conversely, for the sim>emp analyses, the datasets that were used to train the inverse models were larger in size compared to those used in the emp>emp analysis (22500 samples as compared to 1200 samples) and would have contained higher variance, due to being based on the responses of all 30 participants rather than only one specific participant. This led to the opposite situation as compared to the emp>emp analysis, where this time the conditions of the training data would be unfavourable for the CCA inverse model due to the higher variance in the simulated data. For the EEGNet inverse model, the increased amount of data would have reduced the risk of overfitting, whereas the high capacity of the model would still have likely been able to capture a degree of the variance in the data. This is hypothesized to be the likely cause of the EEGNet inverse model performing significantly better than the CCA inverse model in the sim>emp analysis.

6.1.2 Relations between the forward model versions

Aside from the comparison between the inverse models, a comparison was made between the various forward model versions. It was found that the Sig-Drawn model performed significantly worse than the Gamma-Drawn model, which in turn performed significantly worse than the CCA-Drawn and CCA-Frozen models, of which the performances did not differ significantly. Therefore, when testing on the specific data that were used in this study, the CCA-Drawn or CCA-Frozen model would be the best choice for the forward model. However, when testing on entirely different data, the Sig-Drawn model would also be a viable option, even though it performs worst in the current study. This is because the Sig-Drawn model is least directly linked to the specific data that were used in this study. Therefore, we hypothesize the Sig-Drawn model to have a potentially higher capacity for generalization to unseen data.

The hypothesized cause for the relation between the emp>emp results and the sim>emp results obtained using the CCA-Frozen model follows the same intuition as was used for the comparison between the inverse models. The reason for the CCA inverse model to perform significantly better in the emp>emp analysis as compared to the sim>emp analysis is hypothesized to have been a result of the conditions of the training on empirical data being more favourable for CCA as compared to the conditions of training on simulated data. This would be due to a larger degree of variance in the simulated data as compared to the empirical data, which CCA would have difficulty capturing, due to only learning a single spatial filter and a single response vector. Conversely, EEGNet performing better in the sim>emp analysis than in the emp>emp analysis might be explaind by the risk of overfitting, which is higher in the latter than in the former.

Next we observe the relation between the CCA-Frozen and CCA-Drawn versions of the forward model. When using either the CCA or EEGNet inverse model, the CCA-Frozen model did not perform significantly better than the CCA-Drawn model, nor did the CCA-Drawn model perform significantly better than the CCA-frozen model. We hypothesize this to be caused by the small value range of the distributions from which noise parameters are sampled in the CCA-Drawn model. Due to this relatively small range, the effects of sampling from distributions as compared to using the pre-defined parameter values would be minimal. As a result, the difference in accuracies would be small.

The next relation that will be discussed is the relation between the CCA-Drawn and Gamma-Drawn models. For each of the two inverse models, the CCA-Drawn version of the forward model performed significantly better in the sim>emp analysis as compared to the Gamma-Drawn model. The difference between these two models is that the CCA-Drawn model uses signal components that were learned by applying the CCA model on empirical data, whereas the Gamma-Drawn model uses signal components that were modeled based on the response vectors learned by CCA, by using a quadruple-Gamma function. The difference between the performances is not fully unexpected, given that the signal components used by the CCA-Drawn model are directly based on the empirical data, including the data of the tested participants, by learning response vectors using the CCA-mode. Conversely, the signal components

used by the Gamma-Drawn model are merely modeled versions that are similar to said response vectors. This difference means that, for this specific test set, the signals modeled by the CCA-Drawn approach adhere more closely to those in the test set, resulting in higher accuracies. This, together with the obtained mean accuracies using the Gamma-Drawn model (42.46% with the CCA inverse model and 44.59% with the EEGNet inverse model), indicates that the inverse models trained on data simulated with quadruple-Gamma functions do learn to capture a reasonable part of the characteristics of the underlying signals, yet not enough to be on par with the inverse models trained on data simulated using the actual underlying signals themselves.

The last relation that will be discussed is the relation between the Gamma-Drawn and Sig-Drawn models. In the sim>emp analyses using of each of the two inverse models, the Gamma-Drawn model performed significantly better than the Sig-Drawn model. This result was expected for the tests performed in this study. The signal components that were used in the Gamma-Drawn model were, ultimately, modeled directly based on responses that were learned from the empirical data on which the inverse models were tested. In comparison, the signal modeling that was used in the Sig-Drawn model was less directly linked to the used set of empirical data, having only the parameters for the distributions determined based on visual inspection of the responses learned by applying CCA on the empirical data. As a result, the signal components that were used by the Sig-Drawn model do not correspond as directly to the empirical data that was used in this study as the signal components used by the Gamma-Drawn model, leading to lower accuracies. However, the accuracies obtained with the Sig-Drawn model (38.07% with the CCA inverse model and 39.33% with the EEGNet inverse model) do indicate that inverse models trained using data simulated by the Sig-Drawn model can classify empirical data to a certain extent, since the accuracies were higher than the chance level, which was 5%.

6.2 Limitations and future work

The approach that was used in this study shows that training inverse models on simulated data can learn to classify empirical data to a certain extent. However, the approach can be improved upon in future work. There are three main areas in which such future work can be done, namely the forward model, the inverse model, and the testing aspects of this study.

The forward model that was used in this study can be further improved upon. The main area in which further improvement is possible is in the amount of channels that can be modeled by the forward model. The forward model that was implemented in this study can only generate data for one EEG channel. Future work could extend the number of channels that can be modelled using the forward model. This could be attempted by modeling source activities and mapping those activities to various EEG channels by considering head models or employing leadfield matrices, approaches that have previously been used in frameworks called EEGSourceSim (Barzegaran et al., 2019) and SimBCI (Lindgren et al., 2018).

A second aspect of the forward model that can be investigated in future work is the types of

noise that were implemented. In this study, the forward model used pink noise, white noise, visual alpha noise and line noise. However, various types of EEG artifacts are not included in the forward model. Future work could investigate the influence of such artifacts on the accuracies that are eventually obtained by integrating such artifacts in the forward model. Examples of such artifacts could be eye movements and eye blinks, which were previously used in an adaptation of the SimBCI framework by Torres & Daly (2021).

A third manner in which the forward model could be extended in future work is through the further analysis of the signal modeling component and the combination of the signal and noise aspects. In this study, the main inspiration for the signal components came from a set of 30 response pairs that were obtained by applying the CCA model on empirical data. These response pairs were subsequently used to inform the design of the Gamma and Sig versions of the signal modeling. However, the resulting modeled signal components may not be representative for larger populations, thus requiring further investigation by testing on larger amounts of data. In addition, the SNR value distribution that was used in this study was chosen arbitrarily, based on tests performed during this study. As a result, this distribution requires further investigation and tuning, before it can be deemed to be representative for larger populations. These further investigations could be performed by incorporating data from a larger population, based on which the parameter values for the various aspects of the forward model could be further optimized.

Finally, a fourth manner in which the forward model could be extended is through further inspection of the various distributions that were used in the model. The distributions for the noise modeling defined based on the limited amount of noise data that was present in the empirical data used in this study. In addition, the distributions used for the Sig version of the signal modeling component were chosen by visual inspection of the response vectors learned by applying CCA on empirical data. None of these distributions are necessarily optimal. Therefore, future work could incorporate a larger amount of empirical data, based on which the distributions can be further optimized.

The second area in which future work may be performed is that of the inverse models. This study employed two inverse models in order to classify the empirical and simulated data. However, employing other models or adapting the models that were used in this study may lead to increased accuracies with the various forward model versions that were tested in this study. Examples of such a model would be the EEG2Code model introduced by Nagel & Spüler (2018) or its adapted version in which the EEG2Code model was combined with deep learning (Nagel & Spüler, 2019). At present, these models can not be claimed to lead to different accuracies. However, combining knowledge of various types of inverse models, including the models provided as examples, can lead to further improvements of the inverse model.

The third avenue of exploration that is possible for future work is the adaptation of the analyses. The forward models that were used in this study were based on the empirical data on which testing was performed. As a result, there are no results indicating to what degree the implemented forward model versions generalize to other c-VEP data with potentially different codes, amplifiers, or experimental protocols. This generalization ability can be investigated

in further analyses, by applying the various versions of the forward and inverse models on novel empirical data. We hypothesize that the Sig-Drawn version of the forward model would show the largest capacity for generalization, due to being less directly linked to the specific data that was used in this study as compared to the other versions of the forward model.

6.3 Relation to the literature

One challenge within the field of BCI is that BCI techniques often require a calibration session before reaching high performance levels. However, recording the empirical data for such calibration sessions is expensive. In order to combat this challenge, simulated data could be used to calibrate the BCI techniques, replacing the need for empirical data. The results found and discussed in this study indicate that using such simulated data to train BCI techniques can yield performances that are higher than chance level. This is a large step towards the realization of plug-and-play BCI techniques, which can be directly used by novel users without requiring a lengthy calibration session. Furthermore, it would allow for further optimization of BCI techniques without requiring empirical data, by optimizing the techniques on simulated data.

The accuracies obtained using the forward model implemented in this study are currently lower than the results obtained by Torres & Daly (2021), who used an adapted version of the SimBCI framework for data simulation. One possible cause for this difference is one main aspect in which the proposed forward model is inferior to other simulation frameworks, namely by only being able to model data for a single EEG channel, whereas frameworks such as simBCI (Lindgren et al., 2018) and EEGSourceSim (Barzegaran et al., 2019) can model data for multi-channel EEG. However, further improvement of the forward model used in this study could potentially lead to higher accuracies.

The proposed simulation framework, once extended, can be usable within various BCI techniques. One example relates to the 0-train approach introduced by Thielen et al. (2021). This zero-training approach uses previously classified trials to facilitate classification in subsequent trials. By initializing this approach with data simulated using the proposed framework, the performance on initial trials could potentially be improved.

Lastly, the underlying design of this simulation framework can be used in other EEG-based BCI paradigms, rather than only in the c-VEP paradigm. By choosing codes that use only one type of event, such as the short events used in this study, and enforcing the events in the code to occur at particular frequencies, the present forward model could potentially also be used to model SSVEP data. This allows the data generated by the forward model to be used for approaches such as the SSVEP-based Bremen BCI system (Volosyak, 2011).

7 Conclusion

In this study, two inverse models and multiple forward models were implemented. The inverse models were trained on single-channel EEG data that was either empirical or simulated

by the one of the various versions of the forward models, after which they were tested on empirical data. It was found that, by training inverse models on data simulated by various forward model versions, it is possible to use these trained models to classify empirical c-VEP data at a level higher than chance level, yet lower than the level obtained by training and testing on empirical data. The forward and inverse models that were used in this study could be used to inspire further research into the creation of forward models for c-VEP data.

Finally, we want to highlight one important implication of this study. This implication is that, by training inverse models on data that are simulated with the forward models used in this study, it is possible for the models to learn to classify empirical c-VEP data to a degree that is higher than chance level. This indicates that there is a possibility that forward models can be created that, when combined with an inverse model, can learn to predict the classes of empirical c-VEP data. The present study can therefore be considered to be an initial step towards achieving such a feat.

8 Acknowledgements

I would like to thank dr. Jordy Thielen for acting as the daily supervisor during this study, for investing time to have various meetings and discussions, and for providing valuable feedback, both during the implementation and writing phases of this study. I would also like to thank dr. Michael Tangermann for being the second reader for this thesis project.

References

Aznan, N. K. N., Atapour-Abarghouei, A., Bonner, S., Connolly, J. D., Al Moubayed, N., & Breckon, T. P. (2019). Simulating brain signals: Creating synthetic EEG data via neural-based generative models for improved SSVEP classification. In *2019 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8).

Barry, R. J., & De Blasio, F. M. (2021). Characterizing pink and white noise in the human electroencephalogram. *Journal of Neural Engineering*, *18*(3), 034001.

Barzegaran, E., Bosse, S., Kohler, P. J., & Norcia, A. M. (2019). EEGSourceSim: A framework for realistic simulation of EEG scalp data using MRI-based forward models and biologically plausible signals and noise. *Journal of Neuroscience Methods*, *328*, 108377.

Bin, G., Gao, X., Wang, Y., Hong, B., & Gao, S. (2009). VEP-based brain-computer interfaces: time, frequency, and code modulations [research frontier]. *IEEE Computational Intelligence Magazine*, *4*(4), 22–26.

Capilla, A., Pazo-Alvarez, P., Darriba, A., Campo, P., & Gross, J. (2011). Steady-state visual evoked potentials can be explained by temporal superposition of transient event-related responses. *PloS one*, *6*(1), e14543.

Cranmer, K., Brehmer, J., & Louppe, G. (2020). The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, *117*(48), 30055–30062.

Fatourechi, M., Bashashati, A., Ward, R. K., & Birch, G. E. (2007). EMG and EOG artifacts in brain computer interface systems: A survey. *Clinical Neurophysiology*, *118*(3), 480–494.

Friston, K. J., Fletcher, P., Josephs, O., Holmes, A., Rugg, M., & Turner, R. (1998). Event-related fmri: characterizing differential responses. *Neuroimage*, *7*(1), 30–40.

Gao, S., Wang, Y., Gao, X., & Hong, B. (2014). Visual and auditory brain–computer interfaces. *IEEE Transactions on Biomedical Engineering*, *61*(5), 1436–1447.

Hayashi, H., & Kawaguchi, M. (2017, 04). Intraoperative monitoring of flash visual evoked potential under general anesthesia. *Korean Journal of Anesthesiology*, *70*, 127. doi: 10 .4097/kjae.2017.70.2.127

Islam, M. K., Rastegarnia, A., & Yang, Z. (2016). Methods for artifact detection and removal from scalp EEG: A review. *Neurophysiologie Clinique/Clinical Neurophysiology*, *46*(4-5), 287–305.

Lawhern, V. J., Solon, A. J., Waytowich, N. R., Gordon, S. M., Hung, C. P., & Lance, B. J. (2018). EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces. *Journal of Neural Engineering*, *15*(5), 056013.

Lindgren, J. T., Merlini, A., Lecuyer, A., & Andriulli, F. P. (2018). simBCI—a framework for studying BCI methods by simulated EEG. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *26*(11), 2096–2105.

Lotte, F. (2011). Generating artificial EEG signals to reduce BCI calibration time. In *5th International Brain-Computer Interface Workshop* (pp. 176–179).

Martínez-Cagigal, V., Thielen, J., Santamaría-Vázquez, E., Pérez-Velasco, S., Desain, P., & Hornero, R. (2021). Brain–computer interfaces based on code-modulated visual evoked potentials (c-VEP): a literature review. *Journal of Neural Engineering*.

McFarland, D. J., Sarnacki, W. A., Vaughan, T. M., & Wolpaw, J. R. (2005). Brain-computer interface (BCI) operation: signal and noise during early training sessions. *Clinical Neurophysiology*, *116*(1), 56–62.

Nagel, S., & Spüler, M. (2018). Modelling the brain response to arbitrary visual stimulation patterns for a flexible high-speed brain-computer interface. *PLOS One*, *13*(10), e0206107.

Nagel, S., & Spüler, M. (2019). World's fastest brain-computer interface: combining EEG2Code with deep learning. *PLOS One*, *14*(9), e0221909.

Notbohm, A., Kurths, J., & Herrmann, C. S. (2016). Modification of brain oscillations via rhythmic light stimulation provides evidence for entrainment but not for superposition of event-related responses. *Frontiers in Human Neuroscience*, *10*, 10.

Thielen, J., Marsman, P., Farquhar, J., & Desain, P. (2021). From full calibration to zero training for a code-modulated visual evoked potentials for brain–computer interface. *Journal of Neural Engineering*, *18*(5), 056007.

Timmer, J., & Koenig, M. (1995). On generating power law noise. *Astronomy and Astrophysics*, *300*, 707.

Torres, J. A. R., & Daly, I. (2021). How to build a fast and accurate code-modulated brain-computer interface. *Journal of Neural Engineering*, *18*(4), 046052.

Urigüen, J. A., & Garcia-Zapirain, B. (2015). EEG artifact removal—state-of-the-art and guidelines. *Journal of Neural Engineering*, *12*(3), 031001.

Vialatte, F.-B., Maurice, M., Dauwels, J., & Cichocki, A. (2010). Steady-state visually evoked potentials: Focus on essential paradigms and future perspectives. *Progress in Neurobiology*, *90*(4), 418-438. Retrieved from https://www.sciencedirect.com/science/article/pii/S0301008209001853 doi: https://doi.org/10.1016/j.pneurobio.2009.11.005

Volosyak, I. (2011). Ssvep-based bremen–bci interface—boosting information transfer rates. *Journal of Neural Engineering*, *8*(3), 036020.

Waytowich, N., Lawhern, V. J., Garcia, J. O., Cummings, J., Faller, J., Sajda, P., & Vettel, J. M. (2018). Compact convolutional neural networks for classification of asynchronous steady-state visual evoked potentials. *Journal of Neural Engineering*, *15*(6), 066031.

Wyatt-McElvain, K. E., Arruda, J. E., & Rainey, V. R. (2018). Reliability of the flash visual evoked potential P2: double-stimulation study. *Applied Psychophysiology and Biofeedback*, *43*(2), 153–159.

A Additional attempts at signal modeling

Three additional attempts were made regarding the signal modeling aspect of the forward model that was used in this study. These are largely adapted versions of the forward model versions that were discussed in Section 4.4.1.

The first attempt that was made regarded an adaptation to the Gamma approach. In this attempt, the Gamma approach was extended using distributions and parameter sampling. Based on the response vectors obtained by applying the CCA inverse models on empirical data, optimal parameter values for the quadruple-Gamma function were estimated for each of the responses. Subsequently, distributions were fit based on the set of optimal parameter values for each of the parameters in the quadruple-Gamma function. This was done separately for both the responses to short events and the responses to long events. Finally, the signal components were generated by sampling the parameter values from the aforementioned distributions and using the sampled values as input to the quadruple-Gamma function, yielding modeled responses. In this attempt however, it became apparent that the relation between the various parameters had a large influence on the modeled signals. Consequently, by sampling each parameter independently, the shapes of the modeled signals were no longer similar to the responses used in the Gamma version in Section 4.4.1. This led to this attempt being discarded, given that independent sampling would not be expected to yield strong performances.

The second attempt that was made regarded an adaptation to the Sig version of the signal modeling component. It was attempted to estimate the optimal values for each of the parameters of this component, by comparing the results to the response vectors that were learned by applying the CCA inverse models on empirical data. However, the function that was used to model the signal components in the Sig version is non-discrete. As a result, due to implementational details, the strategy that was used for parameter optimization in the Gamma version of the signal modeling component could not be applied. This led to this attempt not being further investigated, in favour of investigating its version as it is discussed in Section 4.4.1.

The final attempt that was made also regarded an adaptation to the Sig version of the signal modeling component. The Sig version as it is discussed in Section 4.4.1 models the responses to long and short events independently. However, in this final attempt, an implementation was made where the sampled parameter values for the responses to long events were dependent on the parameter values sampled for responses to short events. In particular, the times and amplitudes of the second, third, and fourth peak for the response to long events were made to be dependent on the corresponding values for the response to short events. The performances that were recorded in tests with this final attempt were higher than chance level, yet clearly lower than the performances using the Sig version as it is discussed in Section 4.4.1. This lower performance led to this final attempt being discarded, in favour of the Sig version that was eventually used.