Novel Statistical Methods for EEG-Based Brain-Computer Interfaces

by

Tianwen Ma

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Biostatistics)
in The University of Michigan
2022

Doctoral Committee:

Professor Jian Kang, Chair
Associate Research Scientist Jane E. Huggins
Professor Timothy D. Johnson
Professor Ji Zhu

Tianwen Ma

mtianwen@umich.edu

ORCID iD: 0000-0003-2741-7706

DEDICATION

To my parents: Huiqin and Lian'gen!

To myself that keeps exploring along my own path:

Tel est mon destin

Je vais mon chemin

Ainsi passent mes heures

Au rhythme entêtant des battements de mon coeur

— Jean-Jacques Goldman

ACKNOWLEDGEMENTS

Sitting in front of my computer on Sep 10, 2022, I just cannot imagine I have successfully completed my oral defense. This is a truly arduous journey, and I cannot finish my doctoral study without the help from my dear advisors and friends.

First, I really would like to thank my doctoral committee chair, Professor Jian Kang. The first time when I know Jian was at the Big Data Summer Institute in 2015. He was giving a talk on spatial-temporal data analysis with applications to neuro-imaging studies. I am intrigued by his research topics and that made me decide to choose him as my Ph.D. advisor. He is always patient and optimistic about my work, even when I have been stuck in my first dissertation project for two years. It is always a relief after our weekly meeting because he can always point out my confusion with my brief description.

In addition, I would like to thank my GSRA advisor, Professor Timothy D. Johnson. I accidentally knew his name when I searched potential study concentration upon my graduation application. He was the only person who conducted research on neuro-imaging studies (one year before Jian came back to Michigan), and he was the primary reason I decided to study Biostatistics. I was fortunate to have Tim supervise my regular research work, and I really have gained a lot of experience in clinical research and interactions with physicians at UMHS.

Next, I would like to thank my dissertation committee member, Professor Jane E. Huggins. I was so confused about the BCI data and spent much time understanding the structure of the data. She always pays attention to my questions and finds the most proper way to answer them, and I am really grateful for her support when I wrote my first conference paper.

I also would like to thank my cognate, Professor Ji Zhu. I still remember my first office visit with Ji during my junior year, and we are discussing the prospective graduate admission in Statistics, and he is super nice and responsive to my concerns. During my doctoral study, Ji also provided helpful and useful comments on my statistical modeling.

Finally, I would like to thank Professor Ben Wu (Renmin University of China) for his support on the problem formulation for my first dissertation project. I would like to thank clinical Professor Suzanne Chong (Indiana University). Although it took more than 5 years to finish our first- and second-stage analyses, I was so relieved when Suzanne provided me with helpful advice on the clinical research, supported me with an ASER grant, and told me (actually my mom) how to stay up late in a "scientific" way to look after my grandmother.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

ABSTRACT

An electroencephalogram (EEG)-based brain-computer interface (BCI) is a device that interprets EEG signal patterns in brain activity in response to external stimuli, e.g., the event-related potential (ERP), to operate technology. It has been used to assist people with severe neuromuscular disabilities for normal communication, such as "typing" words without using a physical keyboard. One of the most popular non-invasive BCIs is the P300 ERP-based BCI design. The P300 ERP is a particular ERP that occurs in response to a rare but relevant event (target) among a series of irrelevant events (non-target). It has a *positive* deflection in voltage around *300 ms* post event time. A P300 ERP speller presents a sequence of events on a virtual keyboard and analyzes EEG signals in a fixed time window after each event to make a *binary* decision whether an ERP response is a target event or not. Despite the well-established framework with many successful classifiers, the current ERP-BCI design faces several challenges such as a lack of statistical interpretation of brain activity, relatively slow spelling speed, and long calibration time. In this dissertation, we develop several novel statistical methods and algorithms to address the above challenges.

In the first project, we propose a Bayesian generative model to fit the probability distribution of multi-trial EEG signals in the BCI system. Existing machine learning methods focus on constructing the ERP classifiers, but they pay less attention to interpreting brain activity due to the overlap between adjacent EEG signal segments during the signal pre-processing procedure; our model explicitly addresses this challenge by developing a new Gaussian Process (GP)-based model to estimate the spatial-temporal varying trajectories of P300 ERP responses. The proposed model can select important time windows in which the average brain activity in response to the target and non-target stimuli is different (split) or the same (merge); thus, The GP is termed the split-and-merge GP (SMGP). We also propose a participant-specific information criterion for brain region ranking and selection. Our inference results provide statistical evidence of P300 ERP responses, help design user-specific profiles for efficient BCIs, and demonstrate the importance of ERPs from the visual cortex for P300 speller performance. We design extensive simulation studies based on the database from the University of Michigan Direct Brain Interface Lab (UM-DBI). The robustness and reproducibility of our analysis is justified by cross-participant comparisons and extensive simulation studies.

In the second project, we develop a sequence-based algorithm for adaptive stimulus selection by Thompson sampling in the multi-armed bandit (MAB) problem with multiple selections. Thompson sampling is a heuristic algorithm for sequential decision making that addresses the exploration-exploitation dilemma in the MAB problem. It chooses the optimal action by maximizing the expected reward function with respect to the posterior distribution of the parameters. During each sequence, the algorithm selects a random subset of stimulus groups with a fixed size by the posterior probability, aiming to identify all target stimuli and to improve spelling speed by reducing unnecessary non-target stimuli. In addition, we adopt an efficient method to compute stimulus-specific rewards based on classifier scores under the Bayesian inference framework. We further improve spelling efficiency by integrating a language model into the prior specification. We perform simulations to compare different configurations of stimulus selection paradigms, and show that the proposed adaptive stimulus selection performs more efficiently than the conventional paradigm.

In the third project, we propose a BAyesian SemI-supervised Classification (BA-

SIC) method for data integration of EEG-BCI data from multiple participants. Calibration in BCI refers to the procedure of training the classifier. The existing calibration method only uses data from participants themselves with lengthy training time and thus introducing the noise due to attention shifts and mental fatigue. BASIC aims to reduce the calibration time of a new participant by borrowing information from calibration data of the source participants, which can improve classification accuracy and communication efficiency in the usage of ERP-BCI. BASIC specifies the joint distribution of stimulus-specific EEG signals among source participants via a Bayesian hierarchical mixture model. The posterior inference on BASIC is based on the new participant and selected source participants that are "similar" to the new participant to construct a potentially more powerful classifier. We demonstrate the advantages of BASIC using extensive simulations designed according to the EEG-BCI data collected from the UM-DBI.

CHAPTER I

Introduction to Brain-Computer Interfaces

1.1 Introduction

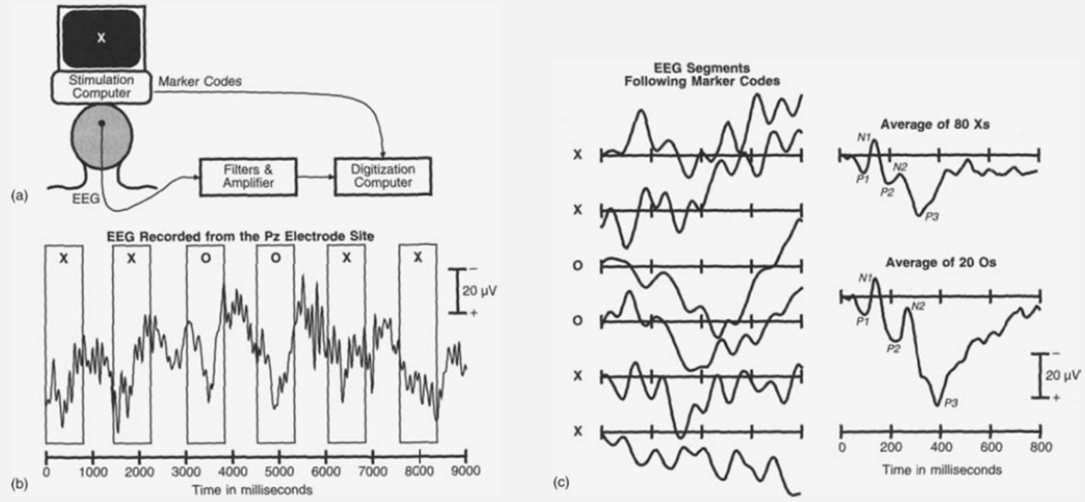
A Brain-Computer Interface (BCI) is a device that interprets and converts brain activity into commands for a computer. A BCI enables the users to respond to their environment only by their brain activity without using muscle control. It is primarily developed to assist people with disabilities to communicate, control artificial limbs, or to control their environment. Research areas include brain activity measurement with invasive and non-invasive technologies, signal processing, signal classification, and new BCI application development (*Hoffmann*, 2007). Among all BCI applications, an electroencephalogram (EEG)-based BCI speller system is a device that enables a person to "type" words by EEG signal patterns in the brain activity in response to external stimuli without using a physical keyboard. It has been used to assist people with severe neuromuscular disabilities, such as amyotrophic lateral sclerosis (ALS), with regular communication (*Wolpaw et al.*, 2018). The brain activity is measured with EEG signals, which has the advantage of non-invasiveness, low cost, and high temporal resolution.

There are several challenges in the EEG-based BCI speller system. First, the signal-to-noise ratio of EEG signals is very low, so to achieve a decent spelling accuracy, users have to repeat the experiment many times to collect enough data. Second, when users spend too much time calibrating this BCI, they may experience variations in attention including fatigue and boredom. Such variations can lead to ignored, misperceived, or delayed brain response activity that may further reduce spelling efficiency. Finally, few methods have made statistical inferences on brain activity under the setting where users interact with BCI continuously.

In this chapter, we start with the basic concepts that lay the foundation of the speller system in Section 1.2, introduce the conventional analysis framework in Section 1.3, present commonly used paradigms in Section 1.4, and review the existing methods for signal classification and online implementations in Section 1.5.

1.2 Basic Concepts

Calibration & Free-typing: We refer to training the binary classifier and testing on the additional data as the procedures of calibration and free-typing, respectively.
Channel (Electrode): An channel is defined as an electrode capturing the brain activity at a certain location. In general, multiple electrodes are placed on the scalp to achieve stable EEG measurement, and the resulting raw EEG signals are in the form of a long matrix.
Event: A common technique to visualize the EEG signals is to present a task (or an event) to a participant while the EEG is recording. The goal is to analyze that part of signal relevant to the event. The event usually starts from the onset of the event and lasts for hundreds of milliseconds (*Rodden and Stemmer*, 2008). We also refer to an event as a stimulus group throughout the entire thesis.
Event-Related Potential (ERP): The technique described previously enables the researchers to identify important brain activity related to the event in the form of the EEG measurement. Since the EEG records electrical activity on the scalp in the unit of voltage, the concept is named the event-related potential (ERP).
Oddball Paradigm: A commonly used experiment that can produce a specific



Luck, Steven J., An Introduction to the Event-Related Potential Technique, figure from page 8 “Example of ERP experiment,” © 2005 Massachusetts Institute of Technology, by permission of The MIT Press.

Figure 1.1: An illustration of the oddball paradigm, which is commonly used to produce a specific ERP component (*Luck*, 2014). Panel A: The subject views the letter X or O presented on a computer screen. Whenever an X or O appears on the screen, a computer program sends this information to the program that records the EEG. The X is presented more often than the O, so a total of 80 times and 20 times are for the X and O, respectively, while the EEG is continuously recorded. Pre-processing techniques such as filtering and amplification are performed to better visualize the EEG signals. Panel B: The EEG is now segmented into 800ms time windows (or epochs) across events, shown by the rectangles. We observe large variability across different events. Panel C: However, if we average the signals across characters, the amplitude change (positive deflection) between 400 ms and 800 ms post event for the infrequent O is generally larger than the frequent X.

ERP signal component. We demonstrate the principle using Figure 1.1 by *Luck* 2014. Two characters X and O are presented to the participant in a random order. Each character is displayed on the screen for a fixed amount of time. When a new character appears, the computer will mark and record the EEG waveforms. The left line of the rectangular block is the onset corresponding to the highlight of each character, while the EEG signals are recorded continuously. The recorded signals are filtered and amplified for better visualization. The EEG signal is then extracted and segmented into 100 response windows of 800 ms, shown by the rectangular blocks. Since the character O appears less frequently than the character X, a total of 80 and 20 windows are for characters X and O, respectively. Although we observe large variability across events, the amplitude change (positive deflection) between 400 ms and 800 ms post event for the infrequent O is larger than the frequent X when we average the signals across characters.

P300 ERP: A particular ERP that occurs in response to a rare, but relevant, event under the oddball paradigm. The relevant (target) P300 ERP has a *positive* deflection in voltage with latency (the delay from the onset of the event to the first response peak) around *300* ms. It is primarily used as biological evidence for classifying EEG signals in the P300 ERP speller (*Rodden and Stemmer*, 2008). Although P300 ERP response is a widely accepted term for this particular signal pattern, we do not rule out other types of ERPs that help interpret and classify brain activity.

1.3 Conventional Analysis Framework

In a visual P300 ERP-BCI speller, participants wear EEG caps that capture multichannel EEG signals, while a virtual keyboard (screen) is presented to the participants. A combination of characters, defined as stimulus groups, are highlighted sequentially on the screen with pre-determined time intervals. Participants are asked to focus on one target character of interest that they wish to type on the screen.

Participants are asked to mentally count when they see a stimulus group containing the target character and to ignore all other stimulus groups. When a stimulus group contains the target character, it is called a target stimulus, and it should elicit a P300 ERP response.

The conventional P300 ERP-BCI design presents a sequence of stimulus groups on a virtual keyboard. Participants focus on a specific character and respond to different stimulus groups with different brain activity (P300 or no P300). These signals are recorded by EEG. Artifact removal, filtering, amplification, and segmentation techniques are applied to raw signals sequentially. The resulting training set consists of a matrix of EEG feature vectors and a vector of their corresponding true labels. Classifiers are constructed to analyze the EEG signals from the onset of stimulus group with a fixed time response window, and make a *binary* decision whether a P300 ERP response is elicited. Then, the binary classification results are converted into character-level probabilities. Finally, the character with the highest probability is selected and passed to participants as feedback. Figure 1.2 provides a flowchart of this procedure for the P300 ERP-BCI speller (*Hoffmann*, 2007).

1.4 Common P300 Speller Paradigms

Most P300 Speller Paradigms are based on the oddball paradigm to elicit P300 ERPs in response to a rare, but relevant event among a series of irrelevant events. The key here is to incorporate the mechanism of the oddball paradigm into constructing stimulus groups. We introduce two P300 speller paradigms: the row-column paradigm and the checkerboard paradigm.

1.4.1 The Row-Column Paradigm

The Row-Column Paradigm (RCP) is one of the most classical stimulus presentation paradigms proposed by (*Farwell and Donchin*, 1988) in 1988. In RCP, the

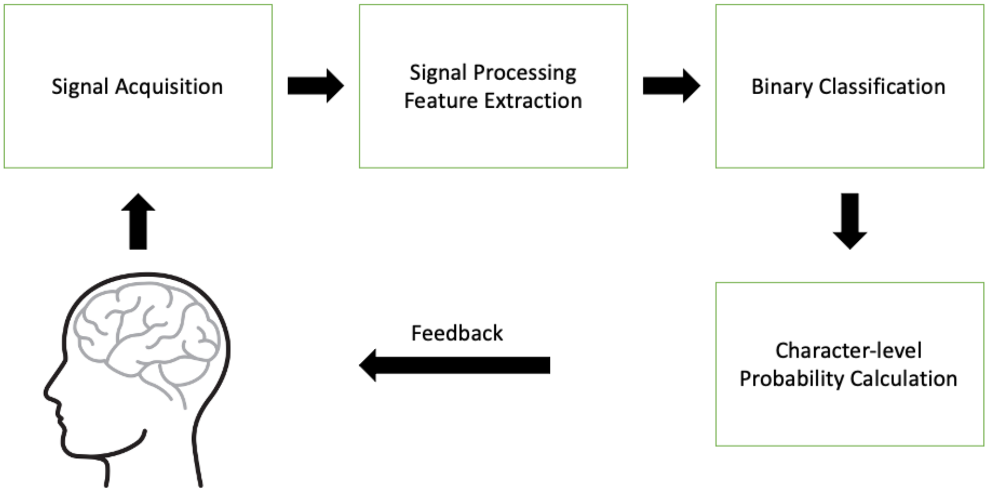
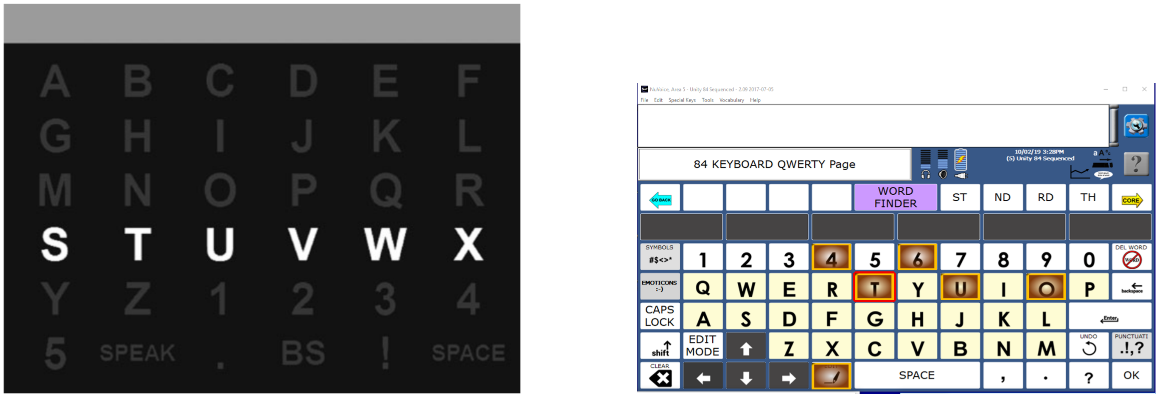


Figure 1.2: An illustration of a closed-loop analysis framework of the P300 ERP-BCI speller. First, participants are asked to wear EEG caps and sit next to a virtual screen. The screen presents a sequence of stimulus groups, to which the human brain elicits different ERP responses. These ERP responses are recorded by the EEG machine. Certain signal pre-processing procedures, including the artifact removal, bandpass filter, signal amplification, and signal segmentation, are performed to the raw signals sequentially such that the resulting dataset consists of a matrix of EEG feature vectors and a vector of their true labels. For the multi-channel EEG signals, it usually concatenate each dimension to form a long vector without considering the spatial dependency. Next, classifiers are constructed to analyze the EEG signals from the onset of stimulus group with a fixed time response window (usually 800 ms), and make a binary decision whether it contains a P300 ERP response or not. Then, the binary classification results are converted into character-level probabilities. Finally, the character with the highest probability is selected and passed to participants for the feedback.

screen for the P300 Speller consists of a 6 × 6 grid of characters. Stimulus groups are defined as rows and columns of characters on the screen. We define a sequence as the stimulus groups that loop over all six rows and columns. The row and column stimulus groups are shown in a random order. Each sequence has exactly *two* target stimulus groups that are supposed to elicit P300 ERPs (one row and one column), and the intersection of two target stimulus groups is the target character of interest. The left panel in Figure 1.3 provides an illustration of the RCP.



A	B	C	D	E	F
G	H	I	J	K	L
M	N	O	P	Q	R
S	T	U	V	W	X
Y	Z	1	2	3	4
5	SPEAK	.	BS	!	SPACE

NuVoice, Area 5 - Unity 84 Sequenced - 2.09 2017-07-05

10/02/19 3:28PM
(5) Unity 84 Sequenced

Figure 1.3: **Left Panel:** An illustration of the virtual keyboard that adopts the row-column paradigm (RCP) with a 6 × 6 grid of characters. The figure is retrieved from the figure 1 in the work (*Thompson et al.*, 2014). The fourth row is currently being highlighted. **Right Panel:** An illustration of the virtual keyboard that adopts the checkerboard paradigm (CBP) with a total of 84 locations. A scatter of characters are currently being highlighted. The figure is taken as a snapshot from the software interface NuVoice software with Unity Language Encoding (*PRC-Saltillo*, 2009).

1.4.2 The Checkerboard Paradigm

The Checkerboard Paradigm (CBP) is a another popular stimulus presentation paradigm proposed by (*Townsend et al.*, 2010) in 2010. Instead of using the regular shapes of rows and columns as the stimulus groups, they define a stimulus group as a scatter of characters on the screen with more randomness. Within each sequence under the CBP design, exactly *two* target stimulus groups are supposed to elicit target P300 ERP responses, and the intersection of two target P300 ERP responses becomes the target character of interest. The CBP can be applied to the virtual keyboard with more general key configurations. The right panel of Figure 1.3 provides an illustration of CBP with an 84-key layout. A detailed description with an example can be found in Section 3.2.1.

1.5 Existing Methods

Existing methods tackle three primary aspects of ERP-based BCI: binary classifier methods, data-driven stimulus selection methods, and calibration-less methods. In this section, we introduce existing methods relevant to three aspects. For the advantages and disadvantages of each method, please refer to the introduction within each chapter. The first aspect is to develop accurate binary classifiers to detect P300 ERPs by treating it as a supervised learning problem in an offline setting. The second aspect is to develop data-driven stimulus selection algorithms to increase spelling efficiency during free-typing sessions, where we assume that the calibration procedure has been completed. The last aspect is to create a calibration-less BCI by borrowing information from existing participants' data while maintaining similar prediction accuracy. Here, calibration-less methods refer to those methods that train the classifier with fewer sequence replications than existing approach. For calibration methods without using any training data from the new participant, we refer to them as calibration-free methods.

1.5.1 Binary Classifier Methods

The first fundamental problem in P300 ERP-BCI speller is binary classification. Many state-of-the-art machine learning (ML) methods have successfully constructed binary classifiers, such as stepwise linear discriminant analysis (swLDA) (*Donchin et al.*, 2000), (*Krusienski et al.*, 2008), support vector machine (SVM) (*Kaper et al.*, 2004), independent component analysis (ICA) (*Xu et al.*, 2004), linear discriminant analysis (LDA) with xDAWN filter (*Rivet et al.*, 2009), convolutional neural network

(CNN) (*Cecotti and Graser*, 2010), logistic regression (LR) (*Viana et al.*, 2014), random forest (RF) (*Okumuş and Aydemır*, 2017), and XGBoost (*Leoni et al.*, 2021). These discriminant approaches treat target or non-target stimulus groups as the response variable and the extracted-and-segmented EEG signals as feature vectors.

Instead of working in Euclidean space, Riemannian geometry (RG) has recently gained increasing attention due to its fast convergence and it is a natural framework to leverage information from source participants. The Riemannian geometry classifier was originally proposed by *Barachant et al.* in 2011. The input data for Riemannian geometry are the sample covariance matrices. The distance-based algorithm based on Riemannian geometry is called Minimum Distance to Mean (MDM). Instead of computing Euclidean distance between EEG signals within Euclidean space, the MDM method computes the Riemmanian distance between sample covariance matrices on Riemannian manifold, and predicts the class label of which "mean" covariance matrix is the closest to the new covariance matrix with respect to Riemmannian distance. Additional studies, including *Congedo et al.* 2013 and *Barachant and Congedo* 2014, adapted the MDM classifier to the P300 ERP-BCI design by augmenting the covariance matrix with label-specific reference signal patterns. This modified classifier is a combination of first- and second-order statistics and compensates for the loss of temporal structure information with referencing signal components.

1.5.2 Data-driven Stimulus Selection Methods

Another important problem in P300 ERP-BCI speller is to improve spelling efficiency and consider physiological constraints in real-time BCI implementations. One of the most import aspects about data-driven methods is dynamic data collection. *Lenhardt et al.* in 2008 developed a method to dynamically change the number and duration of stimulus groups, according to the subject's current online performance. The naive Bayesian dynamic stopping algorithm (NBDSA) (*Throckmorton et al.*,

2013) specified a stopping criterion on a participant-independent, probability-based (unit-less) metric. In addition, a few studies have incorporated historical EEG data into optimizing decision making on future stimulus selection, known as data-driven stimulus selection methods. *Park and Kim* in 2012 applied a partially observable Markov decision process (POMDP) to compute an optimal decision rule for stimulus selection under the RCP design. *Ma et al.* in 2011 proposed a hierarchy of sets of stimulus groups to solve a stochastic control problem of low computational complexity. They further improved spelling efficiency with a statistical language model. *Kalika et al.* in 2017 developed an adaptive stimulus-based stimulus selection algorithm by maximizing the expected discrimination gain (EDG) function.

1.5.3 Calibration-less Methods

Many recent methods have applied the idea of transfer learning (TL) to reduce the calibration time of the P300 ERP-BCIs. The idea was originally introduced by *Bozinovski and Fulgosi* in 1976. Information was extracted and stored from existing problems, and applied to solve a new, but similar, problem. In statistics, we also denote this concept as data integration (*Lenzerini*, 2002). Different domains for information leverage have been explored.

General Ensemble Learning Methods An intuitive idea to incorporate data from other domains is *ensembles*. Ensembles combine results of different classifiers within the same training set. Each classifier makes predictions on a test set, and the results are combined with a voting process. *Rakotomamonjy and Guigue* in 2008 and *Johnson and Krusienski* in 2009 were the first to apply the ensemble method to P300 ERP-BCIs by averaging the outputs of multiple SVMs and swLDAs (See Section 1.5.1), respectively, where a base binary classifier was trained on a small part of the available data. *Völker et al.* in 2018 and *Onishi* in 2020 applied the ensemble method by averaging the outputs of multiple CNNs to visual and auditory P300 ERP-BCI datasets, respectively. *Onishi and Natsume* also mentioned that ensemble methods with overlapping partitioning criterion yielded better prediction performance than the ensemble methods with a naive partitioning criterion.

Ensemble Learning Generic Information (ELGI) *Xu et al.* in 2015 proposed *Ensemble Learning Generic Information* (ELGI), which combines data from a new participant with data from source participants to form a hybrid ensemble. The was the first time that existing methods used data across participants in the setting of P300 ERP-based BCIs. They split the data of each source participant into target and non-target subsets. They applied the swLDA method to construct the base classifiers by combining different subsets as follows: the target and non-target subsets from the new participant, the target subset from the new participant and the non-target subset from each source participant, and the target subset from each source participant and the non-target subset from the new participant. Thus, the resulting ensemble had $(2N + 1)$ base classifiers, where N is the number of source participants. They further introduced the *Weighted Ensemble Learning Generic Information* (WELGI) (*Xu et al.*, 2016) by adding weights to each base classifier. Similarly, *An et al.* in 2020 proposed a weighted participant-semi-independent classification method (WSSICM) for P300 ERP-based BCIs, where they used SVM as the base classifier. The base classifier was fit by combining the entire data from each source participant and a small portion of data from a new participant. An ad-hoc approach was applied to determine the weighted coefficients of base classifiers for participant selection. Likewise, *Adair et al.* in 2017 proposed an *Evolved Ensemble Learning Generic Information* (eELGI). The authors argued that grouping training sets by participants was not an optimal selection criterion. Instead, they developed an evolutionary algorithm by permuting datasets among source participants to form the base classifiers, which were constructed using swLDA.

Transfer Learning on Riemannian Geometry Recent studies have built the transfer learning by the Riemannian geometry. For example, *Rodrigues et al.* in 2018 presented a transfer learning approach to tackle the heterogeneity of EEG signals across different sessions or participants using Riemannian procrustes analysis (RPA). Before the authors applied the MDM classifier, they applied affine transformations to raw participant-level covariance matrices such that the resulting covariance matrices were less *heterogeneous* across sessions or participants while their Riemannian distances were preserved. *Li et al.* in 2020 also *standardized* the covariance matrices across participants by applying an affine transformation on the participant-specific Riemannian geometric mean covariance matrix. Finally, *Khazem et al.* in 2021 proposed another transfer learning approach, denoted as Minimum Distance to Weighted Mean (MDWM). They combined estimated mean covariance matrices from source participants and the new participant by Riemannian distance. They controlled the trade-off between new and source contributions by the power parameter, but they treated them as a hyper-parameter and did not estimate it during the calibration session.

1.6 Outline of the Dissertation

The rest of the dissertation is organized into three chapters. Chapters II to IV describe research specific to three challenges: a lack of statistical interpretation, low free-typing efficiency, and lengthy calibration time.

In Chapter II, we explore the mechanism of neural activity in response to external stimuli and address the challenge of overlapping ERPs between adjacent stimuli *explicitly*. We develop a new GP-based prior to the spatial-temporal varying trajectories of P300 ERP responses. The proposed prior facilitates selecting important time windows in which the average brain activity in response to the target stimuli and non-target stimuli is different (split) or the same (merge); thus, it is termed the split-and-merge GP (SMGP). We make fully posterior inferences on participant-and-channel-specific P300 ERPs in a fixed EEG response window. Finally, we perform brain region ranking by the participant-specific information criterion.

In Chapter III, we propose a sequence-based adaptive stimulus selection method using the Thompson Sampling approach. We frame the problem as a multi-armed bandit problem with multiple actions. During each sequence, the proposed algorithm selects a fixed subset of stimulus groups by the posterior probability. The algorithm aims to identify all target stimulus groups and enhance spelling speed by reducing the number of unnecessary non-target stimulus groups. We perform extensive simulation studies based on the CB paradigm and demonstrate the robustness of our algorithm by considering both ideal and practical scenarios. Finally, we apply the language model prior to further increase spelling speed.

In Chapter IV, we propose a BAyesian SemI-supervised Classification (BASIC) method to build a participant-dependent, calibration-less framework. Here, the semi-supervised framework is slightly different from the regular setting. On the stimulus-level, it is a supervised learning problem because the stimulus labels are known during calibration process; on the participant-level, it is a regular semi-supervised learning problem because we do not observe the participant labels among the pool of source participants. BASIC reduces the calibration time of a new participant by borrowing data on the level of source participants and specifies the joint distribution of stimulus-specific EEG signals via a Bayesian hierarchical mixture model. We specify the cluster 0 to be the one that matches the new participant, and selection indicators indicate of the resemblance between the new participant and source participants. We use the cluster 0 directly to predict testing data of the new participant. Finally, our proposed hierarchical framework is flexible to other base classifiers with clear parametric forms.

In Chapter V, we summarize our contributions of this dissertation and discuss potential future work.

CHAPTER II

Bayesian Inferences on Neural Activity in EEG-Based Brain-Computer Interfaces

2.1 Introduction

2.1.1 Background

A brain-computer interface (BCI) is a device that interprets brain activity to operate technology. An electroencephalogram (EEG)-based BCI speller system is a particular BCI device that enables a person to "type" words without using a physical keyboard by recording EEG brain activity. It has been used for assisting people with disabilities, such as amyotrophic lateral sclerosis (ALS), with regular communication (*Wolpaw et al.*, 2018). The brain activity is measured with EEG signals, which have the features of non-invasiveness, low cost, and high temporal resolution.

The conventional BCI framework is based on the event-related potential (ERP) BCI design, known as the P300 ERP-BCI design (*Farwell and Donchin*, 1988). However, we also include other types of ERPs that help interpret and classify the brain activity. An ERP is a signal pattern in the brain activity in response to an external event. The P300 ERP is a particular ERP that occurs in response to a rare, but relevant event (e.g., highlighting a group of characters on the screen). The relevant (target) P300 ERP has a *positive* deflection in voltage with the latency (the delay from the onset of the event to the first response peak) around *300* ms (*Rodden and Stemmer*, 2008). The rightmost plot in Figure 2.1 shows the typical target and non-target P300 ERPs from a real participant.

There are three challenges in making valid inferences on brain activity in the P300 ERP-BCI system. First, the signal-to-noise ratio of the EEG signals is quite low. A typical P300 ERP-BCI system requires collecting data from multi-dimensional input and repeated sequences of events. Second, to reduce the time to complete the sequence of events necessary to present all the keys on the virtual keyboard, we minimize the time between adjacent events within each sequence and between adjacent sequences. Thus, the time between events is shorter than the time required to produce a P300 ERP response. Therefore, the observed EEG signal is a mixture of overlapping ERP responses, which may or may not contain a P300 ERP. As far as we know, no formal statistical methods can resolve this mixture and make valid inferences on the overlapping responses. Finally, during the calibration time in the current P300 ERP-BCI system, participants may experience variations in attention from fatigue to boredom, leading to missed or delayed responses that may obscure statistical inferences.

2.1.2 Conventional Framework with Motivating Dataset

The conventional P300 ERP-BCI design presents a sequence of events on a virtual keyboard and analyzes the EEG signals in a fixed time response window after each event to make a *binary* decision whether a P300 ERP response is produced by that event, which forms the fundamental basis of the P300 ERP-BCI operation. For multi-channel EEG signals, channel-specific EEG signal segments are concatenated for binary classification. Here, an EEG channel is defined as an electrode capturing brain activity. Multiple electrodes are placed on the head to achieve stable prediction accuracy. The binary classification results are then converted into character-level

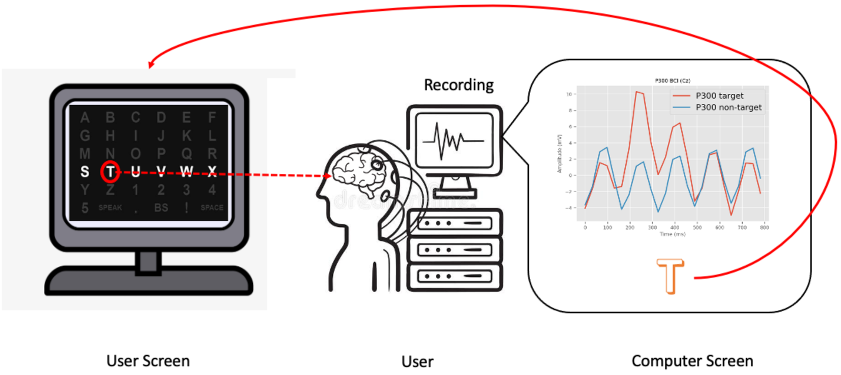


Figure 2.1: An illustration of the conventional procedure of the P300 ERP-BCI operation. The P300 ERP-BCI design presents a sequence of events on a virtual screen to the user. The user focuses on a specific character and responds to different events with different brain signals (P300 or no P300). These brain signals are recorded by the EEG machine. Classifiers are then constructed to analyze EEG signals in a fixed time response window after each event to make a binary decision whether a P300 ERP response is produced. Finally, the binary classification results are converted into character-level probabilities, and the character with the highest probability is shown on the screen.

probabilities. We denote "key" and "target key" as a generic character to be typed and the specific character that the user wants to type, respectively. Usually, events within each sequence cover all the possible keys, but multiple keys can exist in each event. Thus, the P300 ERP-BCI is designed to identify the unique key from the intersection of all events that produce P300 ERP responses within each sequence. Finally, the conventional P300 ERP-BCI design presents a fixed number of events (stimuli) with a fixed number of sequences before the final decision is made. Figure 2.1 describes the procedure of the conventional P300 ERP-BCI operation.

To better illustrate the framework, we briefly introduce the motivating dataset following the experimental protocol by (*Thompson et al.*, 2014). It is part of the database of non-invasive experimental data in the P300 ERP-BCI experiments conducted at the University of Michigan Direct Brain Interface Laboratory (UM-DBI), where the data of 41 participants were recorded under the same protocol mentioned above. Each participant copied one or multiple multi-character phrase(s) during the experimental session. The dataset of each participant consisted of the training (calibration) data and the testing (free-typing) data. We created a participant-specific classifier with the training data and tested on the free-typing data. The study adopted the row-and-column paradigm (RCP) design developed by *Farwell and Donchin* in 1988. The BCI display screen was a 6×6 grid of characters. Each event was either a row stimulus or a column stimulus. The order of the row and column stimuli was random, and it looped through all rows and columns every consecutive 12 stimuli, called a sequence. For each character of interest, participants were asked to mentally count when they saw a row or column stimulus containing the character of interest and to ignore stimuli that did not include the current character of interest. Thus, each sequence always had two events stimuli that were supposed to elicit P300 ERPs (one row and one column) out of every 12 events. In particular, the left side of Figure 2.1 shows 36 characters in a 6×6 grid with the fourth row being highlighted.

Many state-of-the-art machine learning (ML) methods such as stepwise linear discriminant analysis (swLDA) (*Donchin et al.*, 2000), (*Krusienski et al.*, 2008), logistic regression (LR) (*Viana et al.*, 2014), random forest (RF) (*Okumuş and Aydemır*, 2017), support vector machine (SVM) (*Kaper et al.*, 2004), convolutional neural network (CNN) (*Cecotti and Graser*, 2010), independent component analysis (ICA) (*Xu et al.*, 2004), and recent XGBoost (*Leoni et al.*, 2021) have successfully constructed binary classifiers for P300-ERPs. These discriminant approaches treat target or non-target stimuli as the response variable and the truncated-and-concatenated EEG signal segments as feature vectors. Although these approaches are straightforward to implement, it is difficult for them to make statistical inferences about brain activity with overlapping P300 ERP responses. The Gaussian graphical model is a powerful tool to model the conditional dependency structure among multiple Gaussian random variables, and the fglasso algorithm by (*Qiao et al.*, 2019) studies the conditional dependency among p random functions.

As a flexible tool for Bayesian nonparametrics and machine learning, the Gaussian Process (GP), a stochastic process where every finite collection of its realizations follows a multivariate normal distribution, has been widely used for modeling functional and dependent data over time and space (*Rasmussen*, 2003). Different extensions of GPs have been proposed for different neuroscience applications. In particular, for feature selection in scalar-on-image regression, the soft-thresholded GP prior (*Kang et al.*, 2018) models sparse, continuous and piece-wise smooth functions. This prior has also been extended to model the sparsity and dependence in the effects of nodes over a graph in the framework of Bayesian network marker selection (*Cai et al.*, 2020). However, none of these existing GPs can be directly applied to detection of our P300 ERPs in EEG signals.

2.1.3 Our Contributions

To the best of our knowledge, we are among the first to study the probability distribution of multi-trial EEG signals from real participants in BCI experiments using a Bayesian generative model. Our Bayesian analysis explores the mechanism of neural activity in response to external stimuli. Our model *explicitly* addresses the challenge of overlapping ERPs between adjacent stimuli, and the model can be applied to multi-channel EEG signals without signal concatenation nor segmentation. We develop a new GP-based prior to the spatial-temporal varying trajectories of P300 ERP responses. The proposed prior facilitates selecting important time windows in which the average brain activity in response to the target stimuli and non-target stimuli is different (split) or the same (merge); thus, it is termed the split-and-merge GP (SMGP). We make fully posterior inferences on participant-and-channel-specific P300 ERPs in a fixed EEG response window.

Based on our Bayesian analysis, we first aim to identify significant split time windows for frontal, central, parietal, parietal-occipital, and occipital channels. We do not expect to identify significant split time windows for channels close to ears. We study the neural activity patterns among both healthy controls and participants with the Amyotrophic Lateral Sclerosis (ALS) disease under the ERP-BCI design. Finally, we perform the brain region ranking by the participant-specific information criterion. We hypothesize brain regions associated with the cognitive function as well as the visual function will be selected with high reproducibility across participants (*Brunner et al.*, 2010). In addition, we expect that the signal to detect target P300 ERPs for the participant with ALS is weaker than healthy controls, but it should still be significant for classification. Finally, we expect that it may take longer for senior participants than for the young participants to reach the peak of target P300 ERP responses (*Polich et al.*, 1985).

2.2 The Model

2.2.1 Notation and Problem Setup

We begin with the notation. Denote by $\mathbb{R}$ the real line. For any interval $\mathcal{A} \subset \mathbb{R}$, let $\mathbb{I}_{\mathcal{A}}(t) = 1$ if $t \in \mathcal{A}$ and 0 otherwise. Denote by $\mathcal{N}(\mu, \Sigma)$ a normal distribution with mean μ and variance (covariance) Σ. Denote by $\mathcal{GP}(\mu, \kappa)$ the GP with the mean function μ and the covariance kernel κ. All the time variables in this manuscript are multiples of a pre-specified unit time.

Our model focuses on the multi-channel EEG data for one participant. Suppose a total of L target characters are typed for BCI calibration in the training data. For each character $l (l = 1, \dots, L)$, the BCI generates I sequences of $J (J = 12)$ stimuli consisting of six row stimuli, denoted as $1, \cdots, 6$ and six column stimuli, denoted as $7, \cdots, 12$ on the 6×6 keyboard in a random order (Figure 2.2a.). Let $i (i = 1, \dots, I)$ index the sequence. For the ith sequence of the lth target character, let $\mathbf{W}_{l,i} = (W_{l,i,1}, \dots, W_{l,i,12})^\top$ represent the starting time points of the J stimuli (stimulus-occurring indicators) and take values from permutations of $\{1, \cdots, 12\}$. For example, $\boldsymbol{W}_{l,i} = (\underbrace{8, \cdots, 3, 2, \cdots 11}_{J\text{dimension}})^\top$ indicates that the first row, $\cdots$, the last row, the first column $\cdots$ and the last column appear in the 8th stimulus, $\cdots$ 3rd stimulus, 2nd stimulus, $\cdots$, and 11th stimulus, respectively. Let $\mathbf{Y}_l = (Y_{l,1}, \dots, Y_{l,12})^\top$ represent the stimulus-type indicators, where $Y_{l,j} \in \{0, 1\}$ with the constraint $\sum_{j=1}^{6} Y_{l,j} = \sum_{j=7}^{12} Y_{l,j} = 1$. The event $Y_{l,j} = 1$ indicates the lth target letter is located in the jth row stimulus for $j = 1, \dots, 6$ and the $(j-6)$th column stimulus for $j = 7, \dots, 12$. Thus, each possible value of $\mathbf{Y}_l$ uniquely determines one target character on the 6×6 keyboard. For example, $\mathbf{Y}_l = (\underbrace{0, 0, 0, 1, 0, 0}_{\text{row}}, \underbrace{0, 1, 0, 0, 0, 0}_{\text{column}})^\top$ indicates that the target letter is "T" located at the fourth row and the second column. We drop the sequence index i for $\boldsymbol{Y}_l$ because the stimulus-type indicators are always the same given the same character l. For all the sequences, the time domain of the EEG signals are registered to $[0, T]$. Finally, suppose we consider E channels of EEG signals and let $e (e = 1, \cdots, E)$ index the channel, and we denote $X_{l,i,e}(t)$ as the observed EEG signal intensity of the ith sequence and lth target character from channel e at time $t \in [0, T]$.

2.2.2 A Bayesian Generative Model

Suppose we are interested in making inferences on the P300-ERP in a window of length T_z right after the onset of the stimulus. We refer to T_z as the response window length and assume T_z is a multiple of d for simplicity, where d is the stimulus-to-stimulus interval. The total length of time T per sequence is then defined as $T = T_z + (J - 1)d$. We consider the observed EEG signals $X_{l,i,e}(t)$ as a mixture of the J stimulus-induced potentials given stimulus-type indicators $\mathbf{Y}_l$ and stimulus-occurring times $\mathbf{W}_{l,i}$ as follows: For any $t \in [0, T]$,

$$\begin{aligned} X_{l,i,e}(t) &= M_{l,i,e}(t) + \epsilon_{l,i,e}(t), \qquad \tau_{l,i,j} = t - (W_{l,i,j} - 1)d, \\ M_{l,i,e}(t) &= \sum_{j=1}^{J} \left[\beta_{1,e}(\tau_{l,i,j}) Y_{l,j} + \beta_{0,e}(\tau_{l,i,j})(1 - Y_{l,j})\right] \mathbb{I}_{[0,T_z]}(\tau_{l,i,j}), \end{aligned} \tag{2.1}$$

where $M_{l,i,e}(t)$ is the expected EEG signals at time t from channel e induced by J stimuli that occur at different time points. The two unknown functions $\beta_{1,e}(\tau)$ and $\beta_{0,e}(\tau)$ $(\tau \in [0, T_z])$ represent the average brain activity responses to the target and the non-target stimulus, respectively. To simplify the problem, we assume that the shape and magnitude of ERP functions only depend on the stimulus-type indicators, regardless of the stimulus location or the stimulus order. The random noise $\epsilon_{l,i,e}(t)$ characterizes the intrinsic brain activity of channel e that is unrelated to the stimulus responses. Assuming that $\epsilon_{l,i,e}(t)$ is spatially-correlated across channels and temporally dependent, we consider the following additive model:

$$
\begin{aligned}
\epsilon_{l,i,e}(t) &= \zeta_{l,i,e} + \varepsilon_{l,i}(t),\\
\boldsymbol{\zeta}_{l,i} &= (\zeta_{l,i,1}, \dots, \zeta_{l,i,E})^\top \sim \mathcal{N}(0, C_s),\\
\varepsilon_{l,i}(t) &= \rho_{t,0} + \sum_{m=1}^{q} \rho_{t,m}\varepsilon_{l,i}(t - md) + \varepsilon_{l,i,0}(t), \quad \varepsilon_{l,i,0}(t) \sim \mathcal{N}(0, \sigma_x^2),
\end{aligned}
$$

where $\zeta_{l,i,e}$ is the channel-specific random effect and $\zeta_{l,i,1}, \dots, \zeta_{l,i,E}$ jointly follows a multivariate normal distribution with the mean zero and the covariance matrix C_s. The temporal random effect $\varepsilon_{l,i}(t)$ is assumed to follow an autoregressive model of order q and noise variance σ_x^2. For a given channel e and a letter l, Figure 2.2c illustrates the proposed Bayesian generative model for half the length of a sequence. Among the consecutive six stimuli, there exists one target stimulus at the 4th stimulus.

2.2.3 The Split-and-Merge GP

To identify the time window that contains major differences in brain activity responses between target and non-target stimuli, we develop a new GP-based model to model the joint prior distribution of $\beta_{0,e}(\tau)$ and $\beta_{1,e}(\tau)$, for $\tau \in [0, T_z]$, named as the split-and-merge GP (SMGP). For $k = 0, 1$, we assume that $\{\beta_{k,1}(\tau), \dots, \beta_{k,E}(\tau)\}$ are independent and marginally follow the same prior distribution specified by the SMGP. For simplicity, we drop the channel-specific subscript e to specify the SMGP as follows:

$$\beta_k(\tau) = \alpha_k(\tau)\zeta(\tau) + \alpha_0(\tau)\{1 - \zeta(\tau)\}, \tag{2.2}$$

where $\alpha_k(\tau) \sim \mathcal{GP}(0, \kappa_\alpha)$ and $\zeta(\tau) \in [0, 1]$. Note that $\beta_0(\tau) = \alpha_0(\tau)$ and $\beta_1(\tau)$ is the weighted average between $\alpha_1(\tau)$ and $\alpha_0(\tau)$ by $\zeta(\tau)$. When $\zeta(\tau) = 0$, $\beta_0(\tau) = \beta_1(\tau)$ with probability one, i.e. the two processes are merged; when $\zeta(\tau) = 1$, $\beta_0(\tau) \neq \beta_1(\tau)$

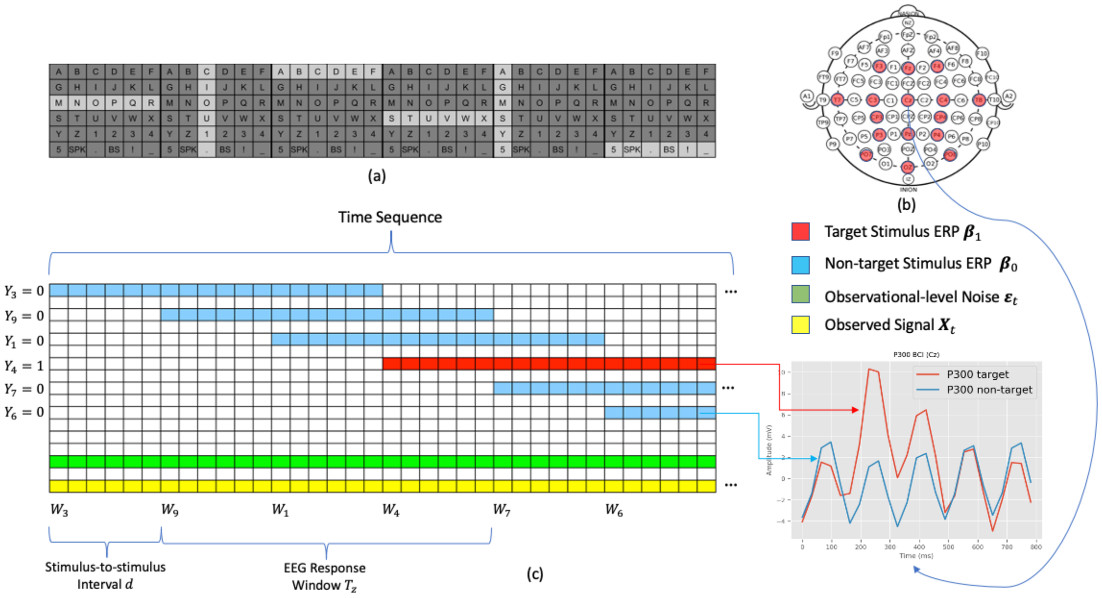

(a)
(b)
Target Stimulus ERP $\boldsymbol{\beta}_1$
Non-target Stimulus ERP $\boldsymbol{\beta}_0$
Observational-level Noise $\boldsymbol{\varepsilon}_t$
Observed Signal $\boldsymbol{X}_t$
Time Sequence
$Y_3 = 0$
$Y_9 = 0$
$Y_1 = 0$
$Y_4 = 1$
$Y_7 = 0$
$Y_6 = 0$
W_3
W_9
W_1
W_4
W_7
W_6
Stimulus-to-stimulus Interval d
EEG Response Window T_z
(c)
P300 BCI (Cz)
P300 target
P300 non-target
Amplitude (mV)
Time (ms)
INION

Figure 2.2: (a). A figure showing a 6 × 6 grid screen of the ERP-BCI speller system, where only one row or one column was being flashed grey for each stimulus. (b). A figure from *Wikimedia Commons* by Brylie Christopher Oxley / CC0, 2017, demonstrating a 64-channel EEG locations using the International 10–20 standard developed by (*Jasper*, 1958). Channels marked with red were used in our ERP-BCI design. (c). An illustration of the data generative mechanism of a single-channel EEG sequence under the ERP-BCI design. Red, blue, green, and yellow blocks represented target responses, non-target responses, background noise irrelevant to stimuli, and observed signals ($\boldsymbol{X}_t$). $\boldsymbol{W}, \boldsymbol{Y}$ were stimulus-occurring indicators and stimulus-type indicators. We assumed each stimulus-related potential could be characterized by $\boldsymbol{\beta}_1$ or $\boldsymbol{\beta}_0$ with a long and fixed response window; the observed signal was generated when we aligned different signal components and summed up at each time point. For example, given the target character was "T", the fourth stimulus was the target one. The graph in the bottom right of the figure illustrates the empirical ERP estimates from channel Cz based on a real participant, where target and non-target ERP estimates were averaged over 570 and 2850 EEG signal segments, respectively. A significant magnitude difference between target and non-target ERPs was observed around 300 ms post-stimulus.

with probability one. Thus, we refer to $\zeta(\tau)$ as the split-and-merge indicator process. Let $\mathcal{W}_s = \{\tau : \zeta(\tau) > \zeta_0\}$ and $\mathcal{W}_m = \{\tau : \zeta(\tau) \leq \zeta_0\}$ represent the split time window and the merge time interval, respectively, where ζ_0 is a hyper-parameter. For efficient posterior inference on $\mathcal{W}_s$ and $\mathcal{W}_m$, we define the truncated GP (TGP) similar to the ordinary GP as follows. A time-continuous stochastic process $\{\zeta(\tau), \tau \in \mathcal{T}\}$ is a truncated GP if and only if for every finite set of indices $\tau_1, \cdots, \tau_p$ in the index set $\mathcal{T}$, $\zeta_{\tau_1}, \cdots, \zeta_{\tau_p}$ follows a multivariate truncated Gaussian distribution, where the truncated domain has the block rectangular shape. In this case, we assign a TGP prior with mean 0.5 and covariance kernel κ_ζ truncated on $[0, 1]$ to $\zeta(\tau)$, i.e. $\zeta(\tau) \sim \mathcal{TGP}_{[0,1]}(0.5, \kappa_\zeta)$.

2.3 Posterior Inference

2.3.1 Model Representation and Prior Specification

Let $\mathcal{MN}(\boldsymbol{M}, \boldsymbol{U}, \boldsymbol{V})$ denote a matrix normal distribution with location matrix $\boldsymbol{M}$ and two scale matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ (*Dawid*, 1981). We rewrite equation (2.1) in the form of matrix normal distribution such that

$$\boldsymbol{X}_{l,i} \sim \mathcal{MN}\left(\boldsymbol{M}_{l,i}, \boldsymbol{C}_t, \boldsymbol{C}_s\right), \tag{2.3}$$

where $\boldsymbol{X}_{l,i} = (\boldsymbol{X}_{l,i,e})_{e=1}^{E}$ and $\boldsymbol{M}_{l,i} = (\boldsymbol{M}_{l,i,e})_{e=1}^{E}$ are matrix-wise observed EEG signals and predicted EEG signals using convolution for the ith sequence, lth target character, respectively. $\boldsymbol{C}_s$ and $\boldsymbol{C}_t$ are the spatial and temporal covariance matrices jointly characterizing the random error $\boldsymbol{\epsilon}_{l,i} = (\boldsymbol{\epsilon}_{l,i,e})_{e=1}^{E}$, respectively. Equation (2.3) can be expressed as

$$vec(\boldsymbol{X}_{l,i}) \sim \mathcal{N}\left(vec(\boldsymbol{M}_{l,i}), \boldsymbol{C}_s \otimes \boldsymbol{C}_t\right), \tag{2.4}$$

where $\otimes$ is the Kronecker product and $vec(\cdot)$ is the vectorization operator that converts the matrix to the column vector. The log-likelihood of the matrix normal model is

$$\sum_{l,i} -\frac{T}{2}\log\det(\boldsymbol{C}_s) - \frac{E}{2}\log\det(\boldsymbol{C}_t) - \frac{1}{2}\mathrm{tr}\left[\boldsymbol{C}_s^{-1}\left(\boldsymbol{X}_{l,i} - \boldsymbol{M}_{l,i}\right)^T \boldsymbol{C}_t^{-1}\left(\boldsymbol{X}_{l,i} - \boldsymbol{M}_{l,i}\right)\right]. \tag{2.5}$$

Therefore, we rewrite the mean structure of $\boldsymbol{M}_{l,i}$ with convolution as follows:

$$\begin{aligned} vec(\boldsymbol{X}_{l,i}) &\sim \mathcal{N}\left(Diag(\boldsymbol{G}_{l,i})vec(\boldsymbol{\beta}), \boldsymbol{C}_s \otimes \boldsymbol{C}_t\right), \quad i = 1, \cdots, I, \quad l = 1, \cdots, L, \\ \boldsymbol{\beta} &= (\boldsymbol{\beta}_e)_{e=1}^E, \quad \boldsymbol{\beta}_e = (\boldsymbol{\beta}_{1,e}^T, \boldsymbol{\beta}_{0,e}^T)^T = S(\boldsymbol{\zeta}_e)\boldsymbol{\alpha}_e = A(\boldsymbol{\alpha}_e)\boldsymbol{\zeta}_e, \\ \boldsymbol{\alpha} &= (\boldsymbol{\alpha}_e)_{e=1}^E, \quad \boldsymbol{\alpha}_e = (\boldsymbol{\alpha}_{1,e}^T, \boldsymbol{\alpha}_{0,e}^T)^T, \end{aligned} \tag{2.6}$$

where $\boldsymbol{\beta}_{1,e}, \boldsymbol{\beta}_{0,e}$ are channel-specific responses to target and non-target stimuli after we have applied the SMGP prior. $\boldsymbol{\alpha}_{1,e}, \boldsymbol{\alpha}_{0,e}$ are channel-specific responses to target and non-target stimuli before selection. They follow the $\mathcal{GP}(0, \kappa_\alpha)$ with the scale parameters $\sigma_{0,1,e}^2, \sigma_{0,0,e}^2$. We use a $\gamma-$exponential function shown in equation (2.7) to specify the kernel covariance.

$$k(x_i, x_j) = \sigma_0^2 \exp\left\{-\left(\frac{||x_i - x_j||_2^2}{s_0}\right)^{\gamma_0}\right\}, \tag{2.7}$$

where $0 \leq \gamma_0 < 2, s_0 > 0$. In practice, we treat them as the hyper-parameters and select the optimal pair by the Bayes factor (*Kass and Raftery*, 1995). $\boldsymbol{\zeta}_e$ follows the truncated normal distribution $\mathcal{TN}_{\mathcal{D}}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with the prior mean $\mathbf{0.5}$ and the prior covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\zeta}}$ on the truncated domain $[0,1]^{T_z}$. We use the method by (*Li and Ghosh*, 2015) in 2015 for efficient sampling. S, A are linear transformations that map $\boldsymbol{\alpha}_e, \boldsymbol{\zeta}_e$ to $\boldsymbol{\beta}_e$. $G_{l,i}$ is the linear transformation that maps $\boldsymbol{\beta}_e$ to the predicted EEG signals via convolution. For $\boldsymbol{C}_s$, we decompose $\boldsymbol{C}_s$ as $\sigma_x^2\tilde{\boldsymbol{C}}_s$, where σ_x^2 follows the inverse gamma distribution $\Gamma^{-1}(a_s, b_s)$ with the shape parameter a_s and the rate parameter b_s, and $\tilde{\boldsymbol{C}}_s$ is a positive definite matrix characterized by the distance measure among selected channels. To simplify, we assume all selected channels share the same distance such that $\tilde{\boldsymbol{C}}_s$ has a compound symmetry structure dependent on the scalar parameter ρ_s. We use an adaptive rejection sampling method (*Gilks and Wild*, 1992) to sample ρ_s, where it is originally generated from the uniform distribution $U(0,1)$. For $\boldsymbol{C}_t$, we assume it depends on the vector $\boldsymbol{\rho}_t$ that follows a discrete uniform distribution $\mathcal{U}_d(\mathcal{V}_\rho)$, where $\boldsymbol{\rho}_t$ is a 2-dimension vector and takes values from a discrete set $\mathcal{V}_\rho$ for which the correlation matrix is invertible, i.e., $||\boldsymbol{\rho}_t||_1 < 1$. Finally, the prior specification is as follows:

$$\begin{aligned} &\alpha_{1,e} \sim \mathcal{GP}(0, \sigma_{1,e}^2 \kappa_\alpha), \quad \alpha_{0,e} \sim \mathcal{GP}(0, \sigma_{0,e}^2 \kappa_\alpha), \quad \boldsymbol{\zeta}_e \sim \mathcal{TN}_{[0,1]}(\mathbf{0.5}, \boldsymbol{\Sigma}_{\boldsymbol{\zeta}}), \\ &\sigma_x^2 \sim \Gamma^{-1}(a_s, b_s), \quad \rho_s \sim U(0,1), \quad \rho_t \sim \mathcal{U}_d(\mathcal{V}_\rho). \end{aligned} \tag{2.8}$$

2.3.2 Markov Chain Monte Carlo

We perform the standard Markov chain Monte Carlo (MCMC) method to sample parameters from their posterior conditional distribution given the training set. We adopt the Gibbs sampler to simulate the posterior distribution of $\boldsymbol{\alpha}, \boldsymbol{\zeta}, \sigma_x^2, \rho_s$, and ρ_t. Since $\boldsymbol{\zeta}$ takes continuous values between 0 and 1, we average the posterior samples of $\boldsymbol{\beta}_1, \boldsymbol{\beta}_0$ whenever $\boldsymbol{\zeta}$ samples are smaller than the threshold ζ_0 for the explicit split-and-merge effect, where ζ_0 is a hyper-parameter, and it takes discrete values in $\{0.1, 0.2, \ldots, 0.8, 0.9\}$ and the optimal one is selected by the Bayes factor. For the convergence check, we run multiple chains with different seed values, and evaluate the conditional log-likelihood and Gelman-Rubin statistic of each parameter (*Gelman and Rubin*, 1992). Details of the Gibbs sampling scheme can be found in the Supplementary Material.

2.3.3 Posterior Predictive Probability for Character Classification

Under the RCP design, the selection of the target character requires the selection of the target row among six candidate rows and the target column among six candidate columns. Let $\boldsymbol{W}^*, \boldsymbol{Y}^*$, and $\boldsymbol{X}^*$ be I^* sequences of stimulus-occurring indicators, stimulus-type indicators, and I^* sequences of matrix-wise EEG signals from new observations given the same target character ω, respectively. Let $\boldsymbol{\Theta}$ be the parameter set defined in equation (2.1). Let $\boldsymbol{y}^\omega \in \{0, 1\}, r^\omega, c^\omega$ be the stimulus-type indicator, row index, and column index associated with the target character ω, respectively. The probability of ω as the target character is

$$\Pr\left(\boldsymbol{Y}^* = \boldsymbol{y}^\omega \mid \boldsymbol{X}^*, \boldsymbol{W}^*, \boldsymbol{X}, \boldsymbol{W}, \boldsymbol{Y}\right) = \int \Pr\left(\boldsymbol{Y}^* = \boldsymbol{y}^w \mid \boldsymbol{\Theta}; \boldsymbol{X}^*, \boldsymbol{W}^*\right) \pi\left(\boldsymbol{\Theta} \mid \boldsymbol{X}, \boldsymbol{W}, \boldsymbol{Y}\right) d\boldsymbol{\Theta}$$

$$= \quad \int \Pr\left(\boldsymbol{Y}^* = \boldsymbol{y}^\omega, y^\omega_{r^\omega} = y^\omega_{c^\omega} = 1, y^*_j = 0, j \notin \{r^\omega, c^\omega\} \mid \boldsymbol{\Theta}; \boldsymbol{X}^*, \boldsymbol{W}^*\right) \pi\left(\boldsymbol{\Theta} \mid \boldsymbol{X}, \boldsymbol{W}, \boldsymbol{Y}\right) d\boldsymbol{\Theta}$$

where $\Pr\left(\boldsymbol{Y}^* = \boldsymbol{y}^\omega, y^\omega_{r^\omega} = y^\omega_{c^\omega} = 1, y^\omega_j = 0, j \notin \{r^\omega, c^\omega\} \mid \boldsymbol{\Theta}; \boldsymbol{X}^*, \boldsymbol{W}^*\right)$ is proportional to

$$\Pr(\boldsymbol{Y}^* = \boldsymbol{y}^\omega) \prod_{i=1}^{I^*} \pi(\boldsymbol{X}^*_i \mid \boldsymbol{\Theta}; y^\omega_{r^\omega} = y^\omega_{c^\omega} = 1, y^\omega_j = 0, j \notin \{r^\omega, c^\omega\}, \boldsymbol{W}^*_i).$$

Here, $\Pr(\boldsymbol{Y}^* = \boldsymbol{y}^\omega) = 1/36$ is the predictive prior on each candidate character if we do not have prior knowledge about the inferred target character. In practice, when we need multiple sequences to select the target character, we compute the cumulative character-based posterior conditional probability vector by multiplying sequence-specific posterior conditional likelihood estimates together.

2.4 Analysis of EEG-BCI Data

We perform the analysis of EEG-BCI data and demonstrate the detailed results from one real BCI participant, referred to as Participant A. Since the primary goal of our analysis is to identify the spatial-temporal pattern of P300 ERP response signals, participants with clear signal patterns are preferred. We select ten participants such that it takes the least squares method fewer than five sequences to achieve 100% accuracy on the training data. First, we fit the model to all 16 channels using the spatial dependency correlation of the compound symmetry structure. We identify the spatial-temporally activated locations. Next, we perform the channel selection based on our method and fit the model to the data for the selected channels using the same spatial dependency assumption. Then, we fit six existing ML methods to the dataset and compare the prediction accuracy of our method to the other ML methods to evaluate the goodness of model fit. Finally, we provide the cross-participant, sensitivity and reproducibility analyses.

2.4.1 Dataset and Pre-processing

For the training session, each participant was asked to wear an EEG cap with 16 channels corresponding to different regions on the brain surface and sit approximately 0.8 m from a 17-inch monitor with the BCI display. Figure 2.2b shows the spatial distribution of channels. Channels marked with red were used for recording and analysis purposes. The abbreviated names were F3, Fz, F4, T7, C3, Cz, C4, T8, CP3, CP4, P3, Pz, P4, PO7, PO8, and Oz (*Thompson et al.*, 2014). For the calibration dataset, each participant copied a 19-character phrase “THE_QUICK_BROWN_FOX” including three spaces. The stimulus presentation and recording were controlled using the BCI2000 software platform (*Schalk et al.*, 2004). An event was defined as a row stimulus or column stimulus, which highlighted for 31.25 ms and paused for 125 ms afterwards, and the total of 156.25 ms was referred to as the stimulus-to-stimulus interval d. We defined the 12 stimuli flashing all rows and columns as a sequence and defined multiple sequences as a super-sequence. In our P300 ERP-BCI design, a super-sequence corresponded to the EEG signals associated with the given target character. During the training session, each super-sequence included 15 sequences, and a total of 19 super-sequences were collected. Extra time was recorded after the last stimulus in the super-sequence. The length of each super-sequence was about 29,000 ms with the sampling rate of 256 Hz.

The data pre-processing steps can be summarized as follows: First, we applied a notch filter at 60 Hz to remove the power line noise and a band-pass filter between 0.5Hz and 6Hz to all 16 channels and then down-sampled raw signals with a decimation factor of eight. Second, we truncated each character-specific super-sequence into 15 sequence segments, where each sequence segment contained 12 consecutive stimuli and subsequent signals of 20 time points to record the entire ERP response to the last stimulus within the single sequence. Each sequence segment contained 2,500 ms, 80 sampling points.

2.4.2 Model Settings

To evaluate the model performance, we chose the odd sequences in the calibration dataset as the training set and used the even sequences as the testing set. This splitting scheme reduced the overlap between adjacent sequences and attenuated the effect of any shift in attention compared to a random training-testing-split scheme. Since it took time for participants to be familiar with the study design or identify the target characters, we excluded the first sequence of each super-sequence from the training set. Therefore, the training set and testing set both ended up with 133 (7 sequences for 19 characters) 80-dimension sequence segments for each channel. We used the cumulative character-level accuracy at seven sequences for prediction evaluation.

For the SMGP method, κ_α was generated from a γ-exponential kernel with hyper-parameters $s_0 = 0.5, \gamma_0 = 1.8, \sigma^2_{0,1} = 1$, and $\sigma^2_{0,0} = 1$. We ran the MCMC algorithm for 2,000 iterations with 1,000 burn-ins for three chains with different seed values. We concluded that the algorithm converged, as the Gelman-Rubin statistics for the parameters of interest were all smaller than 1.1. In addition, we propose the following statistics to rank the channels for the efficient multi-channel fitting based on the SMGP method. The statistics was defined as

$$R_e^2 = \frac{Var\left(\mathbb{E}(\boldsymbol{X}_e(t) \mid \boldsymbol{M}_e(t)\right)}{Var\left(\boldsymbol{X}_e(t)\right)}, \tag{2.9}$$

where the numerator and the denominator explained the variability of the convolution components in equation (2.1) across sequences and the variability of the observed signals across sequences, respectively. Under our model assumption, R_e^2 took values between 0 and 1. The largest R_e^2 among the 16 channels for Participant A was around 0.02. To examine the proposed information criterion, we included from the optimal two and up to five channels for sub-channel analyses. For each combination of channels, we refitted the model and reported the prediction accuracy.

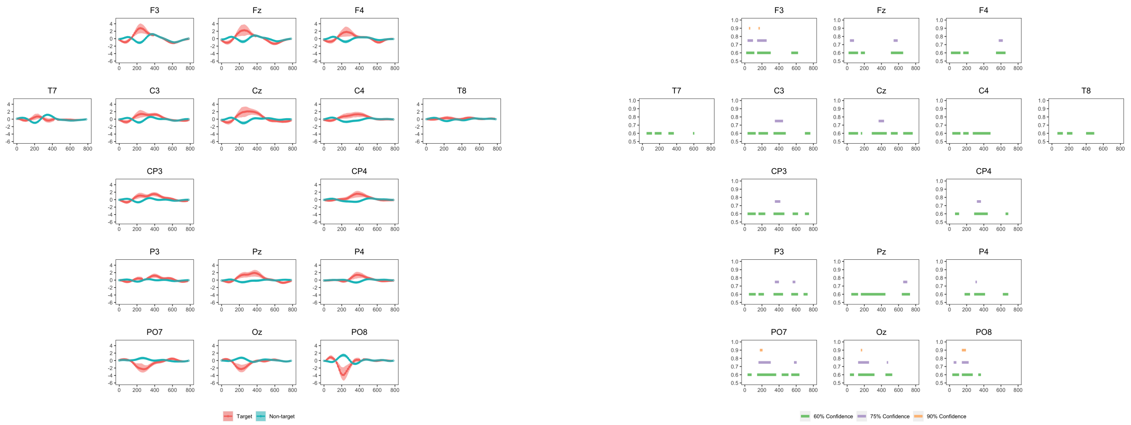


Figure 2.3: **Left Panel**: Channel-specific ERP function estimates of target and non-target stimuli with the 95% credible bands of Participant A. **Right Panel**: Channel-specific significant temporal intervals by varying thresholds of median split probabilities of Participant A. The result was produced by the 16-channel model fitting results. The varying thresholds included $0.6, 0.75$, and 0.9. We arranged the channel-specific plots by their spatial locations. The upper and lower rows represented the front and back of the head. A "z" (zero) referred to a channel placed on the mid-line sagittal plane of the skull. Channels with even numbers $(2, 4, 6, 8)$ referred to the electrode placement on the right side of the head, whereas channels with odd numbers $(1, 3, 5, 7)$ referred to those on the left.

2.4.3 Single-Participant Results

We focus on the results of Participant A in this subsection.

ERP Estimates The left panel of Figure 2.3 showed the mean estimated ERP functions of target and non-target stimuli and their 95% credible bands based on the 16-channel model fitting result. Channel-specific plots are arranged by their relative spatial locations. In general, we saw a clear separation of target against non-target ERP functions for all channels except channel T8. Between 400 ms and 500 ms post stimulus, the target ERP functions gradually declined to zero and collapsed with non-target ERP functions, which shows that our SMGP prior worked well in this case.

Split Windows The right panel of Figure 2.3 showed channel-specific significant split time windows with varying thresholds of median split probabilities of 0.6, 0.75, and 0.9. We rearranged channel-specific brain activity plots by their spatial locations. With 90% posterior probability, the split time windows appeared at 50-65 ms and 160-175 ms for channel F3, at 170-205 ms for channel PO7, at 160-170 ms for channel Oz, and at 150-190 ms for channel PO8 post-stimulus. These significant split time windows corresponded to the first negative peaks of their target ERP curve estimates. For channel Cz, the split time windows appear at 370-430 ms post-stimulus with 75% posterior probability, which approximately corresponded to the first positive peaks of the target P300 ERP response curve estimates. For channel Pz, the split time windows appeared at 650-700 ms post-stimulus with 75% posterior probability. For channels T7, C4, and T8 close to ears, moderate differences in brain activity between target and non-target stimuli were observed, but no split time window was identified with more than 60% posterior probability. A common gap of split time windows around 150 ms was observed, which corresponded to the time points where target and non-target ERP functions first crossed. For time points when target and non-target ERP functions were merged, fewer points were generally selected by the SMGP prior.

Interpretation Two common patterns were observed among the results of the ERP estimates. First, the target ERPs of the frontal and central channels (channel names starting with “F” and “C”) shared the negative drop around 100 ms and reached their first peak with the latency around 250 ms, which corresponded to the N100 and P300 pattern described by Rodden and Stemmer in 2008. Second, the target ERPs of parietal-occipital and occipital channels (channel names starting with “PO” and “O”) reached their negative peaks around 200 ms post-stimulus, and they gradually collapsed with non-target ERP functions *without* reaching a positive peak. Since channels PO7, PO8, and Oz represented the locations of the visual cortex, observing only the negative peaks might be indicative of the pattern of the N2 signal (*Folstein and Van Petten*, 2008). Several discrepancies were also observed. First, the lengths of the split time windows differed among channels. For example, the central channels and frontal channels had the split time window between the onset of the stimulus and 500 ms post-stimulus and between the onset of the stimulus and 400 ms post-stimulus, respectively. Second, the shapes of ERP functions differed among channels. For example, channels C3, CP3, and P3 had secondary peaks around 400 ms post-stimulus, while target ERP functions of other channels collapsed with the non-target ones without clear secondary peaks. Those secondary peaks might be indicative of the pattern of the P3b signals (*van Dinteren et al.*, 2014).

Prediction We compared the prediction accuracy of our SMGP method to other ML methods for Participant A to evaluate the goodness of our model fit. In addition to the same band-pass filter, down-sampling procedure, and splitting scheme, we truncated the original character-specific super-sequence into 180 stimulus signal segments for existing ML methods, where each stimulus signal segment started from the onset of a single stimulus and lasts for 780 ms, i.e. 25 sampling points. Therefore, the training set and testing set both contained 1596 (19 characters, each contained 7 sequences of 12 stimuli) 25-dimension truncated signal segments for each channel. For the swLDA method, the inclusion and exclusion probabilities were 0.1 and 0.15, and at most 30% of the feature vector was selected. Table 2.1 summarizes the cumulative testing prediction accuracy, comparing the SMGP method to other ML methods at seven sequences for the top five selected channels and all 16 channels. The SMGP method achieved 100% accuracy with channels PO8 and PO7, and maintained 100% with more channels included. It performed better than other ML methods. Both the SMGP method and swLDA performed perfectly when all channels were used.

Table 2.1: Cumulative prediction accuracy of Participant A for 19 characters comparing the SMGP method with $\zeta_0 = 0.4$ to other ML methods at seven sequences for the top five selected channels and all 16 channels. The result of channel selection was based on the 16-channel joint fitting result of the SMGP method and the proposed information criterion.

Channels	**SMGP**	**CNN**	**SVM**	**Logistic**	**RF**	**swLDA**	**XGBoost**
PO8, PO7	**1.00**	0.89	0.95	0.95	0.95	0.95	0.95
PO8, PO7, Oz	**1.00**	0.89	1.00	1.00	0.95	1.00	0.95
PO8, PO7, Oz, P4	**1.00**	0.89	1.00	0.95	1.00	1.00	0.95
PO8, PO7, Oz, P4, Cz	**1.00**	0.89	1.00	0.95	1.00	1.00	0.95
All Channels	**1.00**	0.89	0.95	0.95	0.95	1.00	1.00

Sensitivity and Reproducibility We performed sensitivity analysis for the dataset of Participant A by changing the hyper-parameters of the γ-exponential kernel. We assigned 0.4, 0.5, and 0.6 to the scale parameter s_0 and $1.7, 1.8$, and 1.9 to the gamma parameter γ_0. We selected channels PO7, PO8, Oz, P4, and Cz for the sensitivity analysis. Figures S3 and S4 showed the P300 ERP function estimates with 95% credible bands and channel-specific significant temporal intervals by different thresholds of median split probabilities for channels Cz and PO8 under nine variations of kernel hyper-parameters. Overall, the combination of s_0 and γ_0 did not affect either ERP function estimates significantly. For channel Cz, we observed the split window with the threshold of 0.90 when s_0 and γ_0 were in the middle of the hyper-parameter space. Table S4 shows the prediction accuracy with channels PO8, PO7, Oz, P4, and Cz at seven sequences under nine combinations of kernel hyper-parameters. The analysis suggested that a combination of moderate s_0 and γ_0 produced the best prediction performance for Participant A.

2.4.4 Cross-Participant Comparison

First, we applied our information criterion to all 41 participants to identify top-selected channels. Then, we identified spatial-temporal patterns of the neural activity based on selected ten participants. Among ten participants, we selected four typical participants to compare the neural activity patterns between participants with ALS and controls as well as between younger and older participants.

Based on the information criterion in equation (2.9), we identified channels PO7, PO8, Oz, P4, and Cz as the top five selected channels. We performed two sensitivity analyses on channel ranking with respect to bandpass filters and kernel hyper-parameters based on selected ten participants. Overall, they did not change the results much. Details can be found in Section A.3. The target ERPs of frontal and central channels (channel names with "F" and "C") shared the negative drops between 100ms and 150ms and reached their first positive peaks around 300ms post stimulus. Next, the target ERP functions gradually declined to zero and collapse with non-target ERP functions between 600ms and 800ms post stimulus. Finally, the target ERP functions of parietal-occipital, and occipital channels (channel names with "PC" and "O") only reached their negative peaks between 200ms and 250ms post stimulus without reaching further positive peaks.

In comparing the results of Participant E with ALS to the three healthy controls (A, B, and J), Figure 2.4 showed the ERP function estimates of channels Fz of the four participants. We identified a common positive peak for target ERP functions around 300 ms post stimulus although Participant E had the smallest peak magnitude of 0.6 μV compared to the remaining three above 2.0 μV. Finally, we compared the neural activity patterns of two young participants (A and B, around 25 years old) with two senior participants (E and J, around 60 years old). The split-and-merge time windows (SMTW) of frontal channels appeared significantly different between the young and senior participants. On Channel “Fz”, the target ERP functions of all participants showed another negative peak after the first major positive peak. For the young participants (A and B), the target ERP functions merged with the non-target ERP functions after the second negative peak within the 800 ms post-stimulus window; however, for the senior participants (E and J), the target ERP functions were significantly below the non-target ERP functions. One reason is that generally, it takes longer for senior participants to achieve the peak of the target P300 response (*Pavarini et al.*, 2018). Therefore, for a senior participant, if the ERP response window is set to be longer, the target ERP functions may merge with non-target ERP functions after 800 ms.

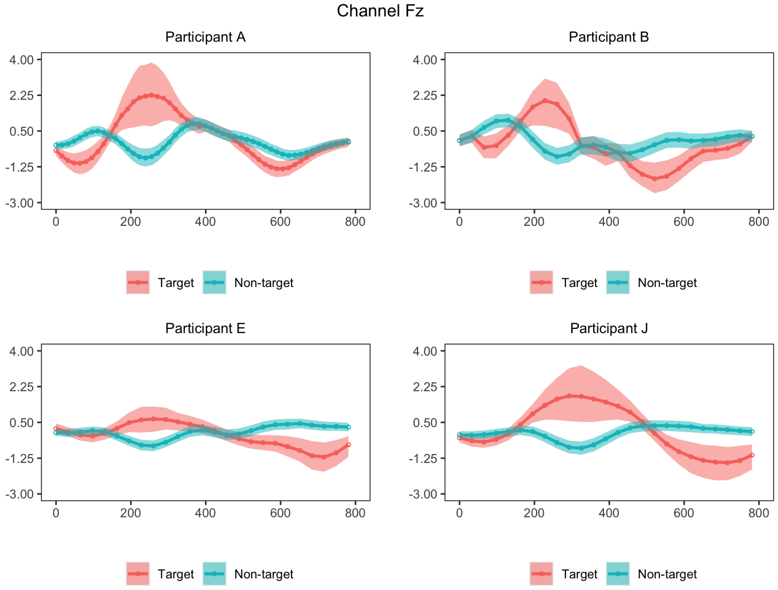


Figure 2.4: ERP function estimates of target and non-target stimuli with 95% credible bands of Participants A, B, E, and J at channel Fz. Participants A and B were young female healthy controls, while Participants E and J were elderly men, of whom only E was diagnosed with ALS.

2.5 Simulations

We performed several simulation studies to make statistical inferences and compare the prediction accuracy of our method to other ML methods. To make the simulated data resemble the real data, we assumed the simulated data with an additive signal-and-noise effect. For the signal component, we applied the convolution rule, and designed the ERP functions based on the work by (*Hoffmann et al.*, 2008) For the noise component, we considered both Gaussian and student-t distributions to mimic different tail distributions with variances close to the real data. We also considered the autoregressive correlation structure to model the temporal association of the background noise. Finally, we considered a scenario where, given true stimulus-type indicators, a subset of target stimuli was randomly selected as non-target ones. This pattern mimicked a situation when participants missed target stimuli due to an attention shift in practical BCI use.

Section 2.5.1 presents a multi-channel simulation study to examine channel ranking and selection by our information criterion, and to evaluate the SMTW with our inference-based criterion. Section 2.5.2 presents the single-channel simulation study with different mis-specification scenarios to test the robustness of our analysis.

2.5.1 Channel Selection and Ranking

Setup We randomly generated stimulus-occurring indicators and stimulus-type indicators with 19 characters of interest, "THE_QUICK_BROWN_FOX," including three spaces. To evaluate the performance and the channel ranking, we designed two groups of pre-specified mean response functions (MRFs 1 and 2). MRF 1 had different temporal separation effects, while MRF 2 had channel-specific SNR values (Figure S1). We considered a true generative scenario with two levels of noise variance, i.e., $\sigma_x^2 \in \{20, 40\}$. We simulated the noise assuming a temporal relationship of $AR(2)$ with the parameter $\rho_t = (0.5, 0)$ and a spatial dependency relationship of compound symmetry structure with the parameter $\rho_s = 0.5$. The EEG signals were generated with a response window of length 935 ms, i.e. 30 time points. We performed 100 dataset replications for this scenario. For each dataset, we generated five sequences per character for training and testing.

Model Settings and Diagnostics All simulated datasets were fitted with equation (2.3). A feature vector was defined as a 3-dimensional super-sequence matrix with five replications and the channel-specific response window was of length 935 ms, i.e. 30 time points. The covariance kernel κ_α was assumed with a γ-exponential kernel. The length-scale, gamma, and scaling of non-target stimuli were $s_0 = 0.5, \gamma_0 = 1.8$, and $\sigma^2_{0,0} = 0.5$, respectively. For simulation studies with MRF 1, the peak ratios of target to non-target stimuli were all 5; for simulation studies with MRF 2, the peak ratios of target to non-target stimuli were 5, 2, and 1, respectively. We ran the MCMC for $2,000$ iterations with $1,000$ burn-ins. The MCMC convergence was assessed by running three chains with different seeds and initial values. The Gelman-Rubin statistics for the parameters of interest were smaller than 1.1, indicating an approximate convergence for each model fit.

Results To evaluate the SMTW, we defined two quantities, the inference-based split window ratio (ISWR) and the inference-based merge window ratio (IMWR) as follows:

$$\text{ISWR}(\boldsymbol{\zeta}) = \frac{\mid \{t : \hat{\zeta}(t) > \zeta_0 \ \&\ \zeta(t) = 1\} \mid}{\mid \{t : \zeta(t) = 1\} \mid}, \ \text{IMWR}(\boldsymbol{\zeta}) = \frac{\mid \{t : \hat{\zeta}(t) \leq \zeta_0 \ \&\ \zeta(t) = 0\} \mid}{\mid \{t : \zeta(t) = 0\} \mid}.$$

Since the swLDA method explicitly performed feature selection, we defined the estimation-based selection window ratio (ESWR) and the estimation-based exclusion window ratio (EEWR) as follows:

$$\text{ESWR}(\boldsymbol{\zeta}) = \frac{\mid \{t : \hat{\zeta}(t) = 1 \ \&\ \zeta(t) = 1\} \mid}{\mid \{t : \zeta(t) = 1\} \mid}, \quad \text{EEWR}(\boldsymbol{\zeta}) = \frac{\mid \{t : \hat{\zeta}(t) = 0 \ \&\ \zeta(t) = 0\} \mid}{\mid \{t : \zeta(t) = 0\} \mid}.$$

Table 2.2: **Upper Panel**: Cumulative prediction accuracy for the multi-channel simulation study under the true generative mechanism with $\sigma_x^2 = 20, \rho_t = (0.5, 0), \rho_s = 0.5$ comparing the SMGP method to other ML methods. The split threshold of SMGP method was $\zeta_0 = 0.5$. Point estimates and standard errors averaged over 100 datasets were reported. Results of the SMGP method were marked in bold. Overall, the SMGP method had the highest and most precise prediction accuracy. **Lower Panel**: The ISWR, IMWR of the SMGP method and the ESWR, EEWR of the swLDA method for the multi-channel simulation study under the true generative mechanism with $\sigma_x^2 = 20, \rho_t = (0.5, 0)$. Channel-specific point estimates and standard errors averaged over 100 datasets were reported.

	Testing Sequences		
Methods	3	4	5
SMGP	**0.91 (0.07)**	**0.96 (0.04)**	**0.99 (0.03)**
Neural Network	0.76 (0.10)	0.87 (0.08)	0.92 (0.07)
SVM	0.81 (0.09)	0.89 (0.07)	0.94 (0.06)
Logistic Regression	0.76 (0.08)	0.87 (0.07)	0.91 (0.06)
Random Forest	0.76 (0.10)	0.86 (0.08)	0.92 (0.06)
swLDA	0.85 (0.08)	0.93 (0.06)	0.97 (0.04)
XGBoost	0.67 (0.11)	0.77 (0.09)	0.85 (0.08)

	SMGP		**swLDA**	
Channels	**ISWR**	**IMWR**	**ESWR**	**EEWR**
1	0.98 (0.03)	0.56 (0.11)	0.32 (0.07)	0.69 (0.08)
2	0.99 (0.03)	0.56 (0.12)	0.32 (0.07)	0.75 (0.09)
3	0.99 (0.02)	0.59 (0.11)	0.26 (0.07)	0.8 (0.09)

Table 2.2 summarized the channel-specific ISWR, IMWR of the SMGP method and the ESWR, EEWR of the swLDA method, and the cumulative prediction accuracy over the number of testing sequences with $\sigma_x^2 = 20$ comparing the SMGP method to other ML methods. The ISWR of the SMGP method was close to 100%, which indicated that our method identified relevant temporal features better than the swLDA method. Our method also had the highest and most precise prediction accuracy among all methods. Similar results were obtained when we used $\sigma_x^2 = 40$. Plots of ERP function estimates for both $\sigma_x^2 = 20, 40$, prediction accuracy, and the SMGP prior evaluation for $\sigma_x^2 = 40$ were shown in the Supplementary Material. For simulation studies with varying SNR values, the means and standard errors of R_e^2 estimates were $20.52(1.55), 9.94(1.07), 4.81(0.82)$ for $\sigma_x^2 = 20$, and

$10.66(1.05), 4.90(0.68), 2.48(0.53)$ for $\sigma_x^2 = 40$ (values multiplied by 100). The information criterion ranked three channels successfully for all the datasets, indicating that the information criterion worked well.

2.5.2 Mis-specification Scenarios

Setup The stimulus-occurring indicators and stimulus-type indicators were generated randomly following the same rule as in Section 2.5.1. We illustrated the design of the pre-specified mean response functions in Figure 2.5. For the data generative mechanism, we considered the following five scenarios with the $AR(2)$ temporal correlation parameter $\rho_t = (0.5, 0)$ and two levels of the noise variance $\sigma_x^2 = 10, 20$. (i). The true generative mechanism scenario simulated the data completely from equation (2.1). (ii). The mis-specified noise scenario simulated the data from equation (2.1) with the noise following a Student-t distribution with 5 degrees of freedom. (iii). The scenario of the shorter response window length simulated the data with pre-specified mean response functions of length 780 ms, i.e. 25 time points. (iv). The scenario of the longer response window length simulated the data with pre-specified mean response functions of length 1,090 ms, i.e. 35 time points. (v). The mis-specified signal scenario simulated the data with a disproportionate distribution of target and non-target stimuli. Given true stimulus-type indicators, a subset (10%) of target stimuli was randomly treated as non-target ones by mistake so that it produced the incorrect target P300 ERPs. The replication size, training sequences, and testing sequences were the same as in Section 2.5.1.

Model Settings and Diagnostics All simulated datasets were fitted with the proposed model with the estimated response window of length 935 ms, i.e. 30 time points. The covariance kernel κ_α was set to an exponential squared kernel. The length-scale, the scaling of target stimuli, and the scaling of non-target stimuli were $s_0 = 0.5, \sigma_{0,1}^2 = 10$, and $\sigma_{0,0}^2 = 0.5$, respectively. We ran the MCMC for $2,000$ iterations with $1,000$ burn-ins. The MCMC convergence was assessed by running three chains with different seeds and initial values. The Gelman-Rubin statistics for the parameters of interest were smaller than 1.1, indicating an approximate convergence.

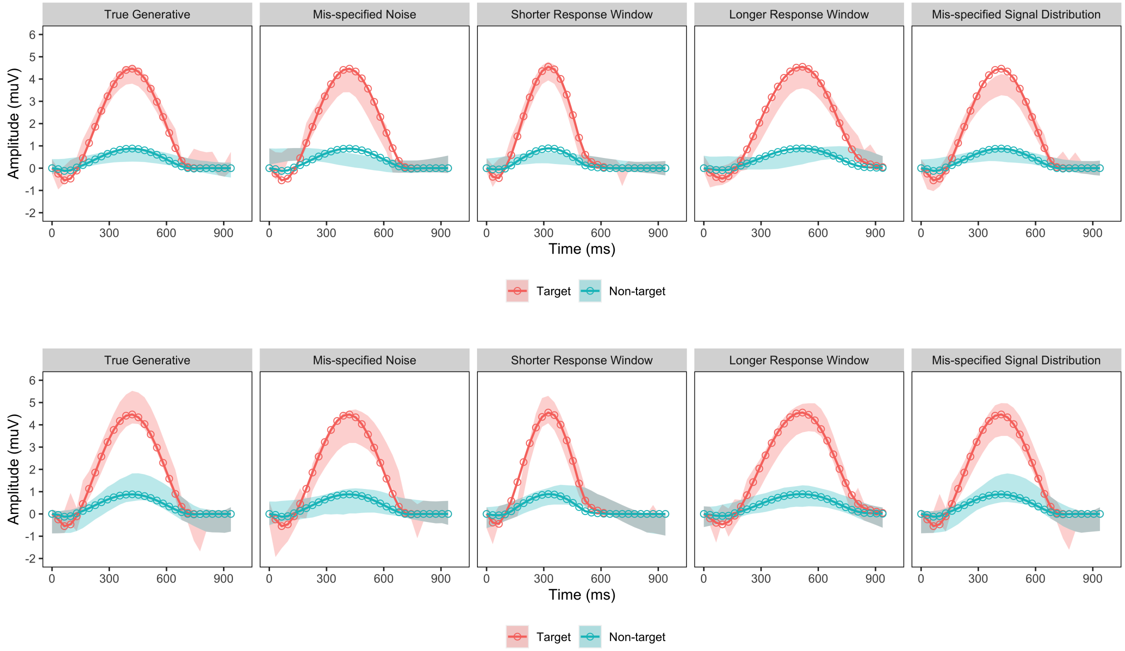


Figure 2.5: The upper and lower panels showed the 95% credible bands of ERP functions to target and non-target stimuli under five simulation scenarios with true parameter $\sigma_x^2 = 10, \rho = (0.5, 0)$ and $\sigma_x^2 = 20, \rho = (0.5, 0)$, respectively. The split threshold was $\zeta_0 = 0.5$. The dots and curves were the true curve values. For the true generative scenario, the credible bands covered the entire true curve. For the mis-specified scenarios, the credible bands almost covered the true curves.

Results Figure 2.5 showed the estimated ERP functions for target and non-target stimuli under five scenarios with true parameters $\sigma_x^2 = 10$ (the upper panel) and $\sigma_x^2 = 20$ (the lower panel). For the true generative scenario, the credible bands covered the entire true curves. For the mis-specified scenarios, credible bands almost covered the true curves. The posterior distributions of σ_x and ρ concentrated around the true values. Table 2.3 summarizes the ISWR, IMWR of the SMGP method and the ESWR, EEWR of the swLDA method under five scenarios with $\sigma_x^2 = 10$ (the upper panel) and $\sigma_x^2 = 20$ (the lower panel). Both point estimates and standard errors over 100 datasets were computed. In the single-channel setting, both the ISWR and IMWR of our method were higher than the ESWR and EEWR of the swLDA method. This result implied that our method identified time windows better than the swLDA method. We also summarized the cumulative prediction accuracy under five scenarios comparing the SMGP method to other ML methods. The prediction accuracy of the SMGP method among the mis-specified scenarios was consistently higher than the other ML methods, suggesting that our analysis was relatively robust to moderate model mis-specifications.

Table 2.3: The detection accuracy of the SMTW of the SMGP and swLDA methods for the single-channel simulation study under five scenarios with $\sigma_x^2 = 10, \rho_t = (0.5, 0)$ in the upper panel and $\sigma_x^2 = 20, \rho_t = (0.5, 0)$ in the lower panel. The split threshold of the SMGP method was $\zeta_0 = 0.5$. Point estimates and standard errors averaged over 100 datasets were reported.

$\sigma_x^2 = 10$	**SMGP**		**swLDA**	
Scenarios	**ISWR**	**IMWR**	**ESWR**	**EEWR**
True Generative	0.89 (0.07)	0.96 (0.05)	0.53 (0.08)	0.75 (0.07)
Mis-specified Noise	0.86 (0.07)	0.94 (0.06)	0.48 (0.07)	0.78 (0.06)
Shorter Window	0.91 (0.07)	0.96 (0.04)	0.64 (0.07)	0.79 (0.07)
Longer Window	0.86 (0.06)	0.96 (0.06)	0.46 (0.07)	0.72 (0.09)
Mis-specified Signal	0.86 (0.07)	0.96 (0.04)	0.49 (0.07)	0.76 (0.07)

$\sigma_x^2 = 20$	**SMGP**		**swLDA**	
Scenarios	**ISWR**	**IMWR**	**ESWR**	**EEWR**
True Generative	0.86 (0.07)	0.94 (0.07)	0.47 (0.07)	0.79 (0.08)
Mis-specified Noise	0.82 (0.08)	0.91 (0.08)	0.41 (0.07)	0.81 (0.07)
Shorter Window	0.88 (0.08)	0.95 (0.05)	0.55 (0.08)	0.84 (0.06)
Longer Window	0.82 (0.07)	0.93 (0.08)	0.41 (0.08)	0.77 (0.08)
Mis-specified Signal	0.83 (0.08)	0.94 (0.07)	0.43 (0.07)	0.81 (0.07)

2.6 Discussion

We have applied a new Bayesian generative framework to model the conditional distribution of multi-sequence EEG signals from real participants under the P300 ERP design. Our Bayesian analysis explored the mechanism of brain activity in response to external stimuli by directly considering the overlapping ERPs between adjacent stimuli without signal concatenation and segmentation. We developed a new GP-based prior to identify the spatial-temporally activated intervals with the split-and-merge GP (SMGP) prior. We proposed an information criterion for channel ranking and confirmed it with existing literature.

We made fully posterior inferences on participant-and-channel specific P300 ERPs with the SMGP prior given a fixed EEG response window. Although past studies by (*D'Avanzo et al.*, 2011) and (*Mowla et al.*, 2018) have developed Bayesian and frequentist filtering methods to estimate amplitude and latency of P300 ERP responses, their results were based on single-trial (sequence) EEG signals, and both methods discarded the spatial dependence among channels. Although the fglasso algorithm by (*Qiao et al.*, 2019) explored the conditional dependence over functional variables, they assumed that different samples of functional variables were independent. However, this independence assumption is violated in the multi-trial P300 ERP-BCI design due to the overlapping signals. Our SMGP method handles multi-channel, multi-sequence, overlapping EEG signals, produces mean P300 ERP estimates with 95% credible bands, and achieves comparable prediction accuracy. When we compare the ERP function estimates of channel Pz for the three methods, they share a small negative drop in amplitude around 100 ms post-stimulus, followed by a major positive peak between 200 ms and 450 ms post-stimulus. Then, the ERP function estimates gradually decline to zero. The identification of channel-specific SMTW provides statistical evidence for the scientific findings of P300 ERP responses.

In terms of channel ranking and selection, the 2015 study by (*McCann et al.*, 2015) pointed out that the difference in P300 ERP-BCI communication efficiency was subtle with five or more channels. Both studies performed channel ranking and selection using the same cohort of data. They identified Cz, Pz, PO7, PO8, and Oz as the top selected channels, which overlaps with our identification of PO7, PO8, Oz, and Cz. These shared selection results provide statistical evidence for spatial distributions of P300 ERP responses. In particular, the finding that channels PO8, PO7, and Oz appear the most frequently supports the finding that the performance of a P300 speller is associated with eye gaze (*Brunner et al.*, 2010). Finally, the participant-specific channel selection helps establish user-specific profiles for efficient brain-computer communications. Thus, we can incorporate user-specific channel selection to design the EEG cap, which increases the implementation speed.

Potential future directions would improve our work. First, we could modify the stimulus presentation paradigm from the current RCP design to the checkerboard design (*Townsend et al.*, 2010). The checkerboard design avoids the refractory effect (*Martens et al.*, 2009) in the RCP design, where participants might miss or fail to produce the second regular P300 ERP response when two target stimuli are too close. In addition, we could measure the participant-specific brain connectivity under the no-control (NoC) condition to specify the prior spatial covariance matrix. For Participant A, we could assume a multi-block compound symmetry structure to estimate within-block, intra-block correlation parameters, and the scalar parameter σ^2. Finally, we could develop the framework of a multi-subject analysis to incorporate the age effect by modifying the priors.

Overall, the proposed generative modeling approach performs innovative statistical inferences on brain activity and provides a promising platform to develop the simulation study framework to test other online P300 ERP-BCI study designs. The Bayesian framework also incorporates prior information such as character-to-character relationships to increase the spelling speed.

CHAPTER III

Adaptive Sequence-based Stimulus Selection in ERP-based Brain-Computer Interfaces

3.1 Introduction

A Brain-Computer Interface (BCI) is a device that interprets patterns of brain activity to assist people with severe neuromuscular diseases with normal communication, such as "typing" words without using a physical keyboard (*Wolpaw et al.*, 2002). One of the most popular non-invasive BCIs is the P300 ERP-based BCI speller (*Farwell and Donchin*, 1988) recorded in the form of the electroencephalogram (EEG) signals. The P300 ERP is a particular event-related potential (ERP) embedded in the EEG signals that occurs in response to a rare, but a relevant event (target stimulus) among a series of irrelevant events (non-target stimuli). The name "P300" comes from the fact that its shape usually has a *positive* deflection in voltage around *300*ms post event time (*Rodden and Stemmer*, 2008).

In a visual P300 ERP-BCI speller, a virtual keyboard is presented to the participant (See Figure 3.3). A combination of characters, defined as the stimulus group, are highlighted sequentially on the screen with pre-specified time intervals. Participants are asked to focus on one target character of interest such that they want to type it on the screen and to mentally count when they see a stimulus group containing the character of interest and to ignore all other stimulus groups. When a stimulus group contains the target character of interest, it is called a target stimulus, and it should elicit a P300 ERP response. The conventional procedure for the P300 ERP speller analyzes EEG signals in a fixed time window after each stimulus to make a *binary* decision whether a target ERP response is elicited. Then, the binary classification results are converted into character-level probabilities. However, despite the straightforward framework, the prediction accuracy is susceptible to noisy EEG signals due to its low signal-to-noise ratio (SNR) property. Therefore, a typical P300 ERP-BCI speller requires collecting data from multi-electrodes with many sequences of replications, where different electrodes are used to capture brain activity on different brain surfaces.

For most existing visual P300 spellers, the set of stimulus groups is usually fixed regardless of target characters of interest. The row-column (RC) paradigm by *Farwell and Donchin* is a typical stimulus selection paradigm following the principle. In the RC paradigm, flash groups are rows and columns of characters in the virtual keyboard. During each sequence, all the row and column stimulus groups are shown with the order permuted. Each sequence has exactly two target stimulus groups, and the intersection of the target stimulus groups is the target character of interest. However, most approaches do not make decisions on subsequent stimulus selection based on previously observed EEG data.

Recently, a few studies have incorporated historical EEG data into the decision making on the stimulus selection, known as *data-driven stimulus selection methods*. Park et al. (*Park and Kim*, 2012) applied the partially observable Markov decision process (POMDP) to compute an optimal stimulus schedule under the RC paradigm. Ma et al. (*Ma et al.*, 2011) proposed a hierarchy of sets of stimulus groups combined with a statistical language model to solve a stochastic control problem of low computational complexity. Kalika et al. (*Kalika et al.*, 2017) developed an adaptive and greedy stimulus-based stimulus selection algorithm based on the expected discrimination gain (EDG) function. These approaches have all made progress in improving the spelling performance compared to the conventional RC paradigm under simulated or real-time BCI settings. However, the POMPD approach becomes difficult to solve for a real-time system with a large search space. The hierarchical approach is likely to accumulate errors, especially for participants with poor performance. The EDG approach also suffers from large computation complexity and approximation is required to estimate the character-level probabilities in presence of the response delay. In addition, none of the methods has applied the adaptive stimulus selection strategies to the Checkerboard (CB) paradigm (*Townsend et al.*, 2010).

We proposed a sequence-based adaptive stimulus selection method by framing the problem as a multi-armed bandit problem with multiple actions (*Komiyama et al.*, 2015). During each sequence, the proposed algorithm selected a fixed subset of stimulus groups by the posterior probability. The algorithm aimed to identify all target stimulus groups and enhance the spelling speed by reducing the number of unnecessary non-target stimulus groups. We applied Thompson Sampling to achieve this goal (*Thompson*, 1933). We performed extensive simulation studies based on the CB paradigm and demonstrate the robustness of our algorithm by considering both ideal and practical scenarios. Finally, we applied the language model prior to initialize the character-level probability to further increase the spelling speed.

3.2 Background

3.2.1 The Checkerboard (CB) Paradigm

In this work, we developed our adaptive stimulus selection algorithm based upon the CB paradigm introduced by (*Townsend et al.*, 2010). The traditional RC paradigm is susceptible to error propagation that leads to attention shifts and frustration for two primary reasons (*Townsend et al.*, 2010). First, due to "adjacency-distraction," the selection errors are most likely to occur next to the target character, especially when non-target stimulus rows or columns that are confusing to participants are close to the target character and they distract the attention of participants. Second, when the target row and column stimuli are too close, participants may ignore or misperceive the second one, which can change the amplitude and shape of P300 ERP responses and lead to poor classification performance, known as the "double-flash" problem. The CB paradigm reduces the impact of the "adjacency-distraction" and completely avoids the "double-flash" problem.

Figure 3.1 provides a simple example of the CB paradigm. Suppose that we have a 3 × 6 keyboard with 18 keys labelled from 1 to 18, and we would like to select the target key with id 8. First, we split the keyboard to two sets (red and blue). We map each set to a 3 × 3 matrix (hidden matrices 1 and 2). Hidden matrices are not necessarily square matrices. The method of mapping is not unique. For the stimulus groups, we extract the rows and columns from each hidden matrix, and end up with H1R1, ..., H1R3, H1C1, ..., H1C3, H2R1, ..., H2R3, and H2C1, ..., H2C3. Each element was a stimulus with three characters being flashed together. A total of 12 stimuli are included within each sequence, and two of them are target ones. Stimuli are presented in the order of rows from hidden matrix 1, rows from hidden matrix 2, columns from hidden matrix 1, and columns from hidden matrix 2, but the order within each row (column) set is random. In this example, H2R3 and H2C1, containing the target key index of 8, are the target stimuli within this sequence.

3.2.2 Thompson Sampling

The multi-armed bandit (MAB) problem is one of the most widely studied sequential decision making problems. In general, during each iteration, a predictor takes one action among a fixed set of actions and receives a reward associated with

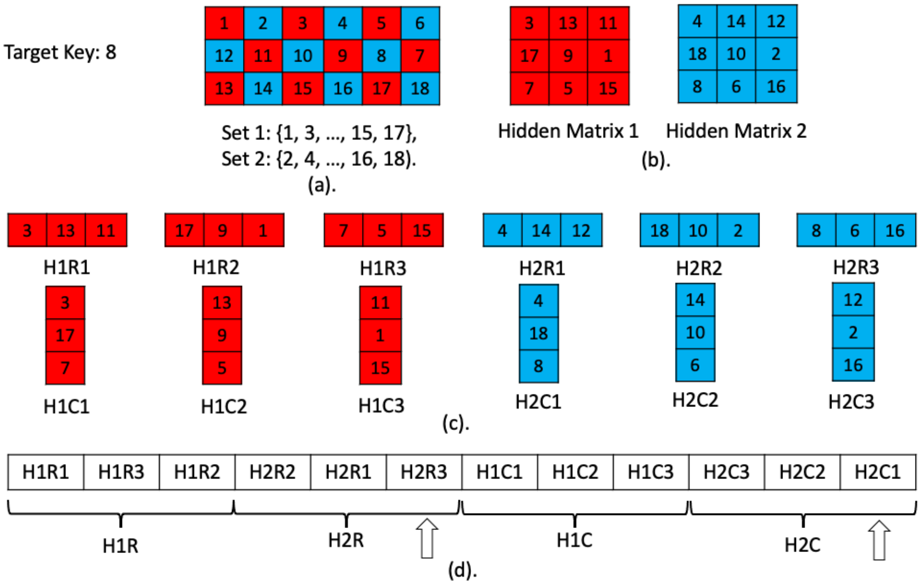


Figure 3.1: An illustration of the checkerboard design. (a). A 3 × 6 keyboard with 18 keys labelled from 1 to 18 in a row switchback order. (b). The keyboard is split into two sets (red and blue) and placed in hidden matrices 1 and 2. (c). We extract rows and columns from two hidden matrices to form 12 stimulus groups within one sequence. (d). We follow the order of rows from hidden matrix 1, rows from hidden matrix 2, columns from hidden matrix 1, and columns from hidden matrix 2 for stimulus presentation. The order within each row (column) set is at random. In this example, H2R3 and H2C1, containing the target key of id 8, are the target stimuli within the drawn sequence.

the selected action. The goal of the predictor is to maximize the cumulative reward over iterations, and the performance is usually evaluated with a regret, which is defined as the difference of the cumulative rewards between the selected and optimal actions. Thompson sampling (TS), originally proposed in (*Thompson*, 1933) in 1933, is a heuristic for tackling the MAB problem where actions are taken in a certain order such that the expected reward functions with respect to the posterior distribution of parameters are maximized. The canonical TS is used to select a single action among a fixed set of actions over multiple iterations. However, in the setting of the P300 ERP-based BCIs, in order to increase the spelling speed, we aim to identify both target stimuli and to reduce the number of unnecessary non-target stimuli. Thus, we need to select multiple actions during each iteration. Fortunately, recent work by (*Komiyama et al.*, 2015) has extended the canonical MAB problem with single action to the MAB problem with multiple actions and provided a theoretical analysis of the optimal regret bound.

In this work, we build our adaptive algorithm upon the problem of the Beta-Bernoulli Bandit (See Example 3.1 in (*Russo et al.*, 2017)). The number of total actions K is the number of stimulus groups that divide the entire virtual keyboard (See Section 3.3.2). An action (or a stimulus group) k produces a reward that follows a Bernoulli distribution with an unknown parameter θ_k. Each θ_k is interpreted as the success probability for each action. We start from a non-informative prior on each θ_k and let these priors follow action-specific Beta distributions with parameters α_k and β_k. The conjugate property between Beta and Bernoulli distributions make it easy to update parameters and fast to converge.

3.2.3 Bayesian Dynamic Stopping Criterion

One of the most important aspects about data-driven stimulus selection methods is the dynamic data collection. Past work in (*Lenhardt et al.*, 2008) developed the method to dynamically change the number and duration of stimulus groups, according to the subject's current online performance. The naive Bayesian dynamic stopping algorithm (NBDSA) in (*Throckmorton et al.*, 2013) specified a stopping criterion on a participant-independent, probability-based (unit-less) metric. Although the classifier scores after each stimulus group (originally transformed by EEG feature vectors via binary classifiers) serve as the natural inputs of the rewards, the actual values are too noisy to use directly. Thus, we modify the NBDSA method to compute "clean" rewards. In general, given the previous character-level probability vector and the resulting classifier score associated with each stimulus group being flashed, we update the character-level probability with the likelihoods of classifier scores accordingly. In this case, other than the rewards that are only available for the selected actions, we update the rewards for the entire action set, which enhances the spelling speed. In the next section, we describe the proposed algorithm in detail.

3.3 Proposed Algorithm

3.3.1 Assumptions

First, we simplify the data generation mechanism of the classifier scores. We assume that the classifier scores of target and non-target stimuli follow normal distributions with means μ_1 and $\mu_0, \mu_1 > \mu_0$, and common variance σ^2. We also consider starting from extracted EEG signals and converted them into μ_1, μ_0, and σ^2 in Sections 3.4.4 and 3.4.5. Second, we assume that the parameters of these two normal distributions are *transferable* between the different flash pattern paradigms under consideration. In other words, we assume that the patterns of P300 ERP responses are stable under the static paradigm and the adaptive stimulus selection paradigm. Finally, although we do not incorporate the impact of the practical constraints directly into the proposed algorithm, we address the modifications for practical implementations in Section 3.3.4.

3.3.2 The Stimulus Group Set

The proposed algorithm is applicable to the paradigm that specifies a valid stimulus group set as follows: Let ω be the index for the target character to spell (denoted as the target index) from a virtual keyboard of size N. We map the characters of the keyboard to the character index set $\mathcal{N}_0 = \{1, \cdots, N\}$. Let $\mathcal{S} = \{S_k : k = 1, \cdots, K\}$ be a stimulus group set such that each element S_k covers character indices of similar sizes, and for each character index n, we always find exactly two stimulus group indices n_1, n_2 such that $\{n\} = S_{n_1} \cap S_{n_2}$. The stimulus group set is a particular way of partitioning the character index set, and the partitioning is not unique. In addition, we can vary the partitioning when we spell the next target character of interest.

3.3.3 The Algorithm

Let T_0 and $p_{\max}$ be the total number of sequences and maximum probability thresholds, respectively. Together they form the stopping criteria. Let $Beta(\alpha, \beta)$ be a beta distribution with shape parameters α, β. For $t = 0$, we initialize the beta distributions and the character-level probability vector $\boldsymbol{P}_0$ with uniform priors and discrete uniform probability of $\frac{1}{N}$, respectively. Let $\theta_k(t)$ be the probability that stimulus group k contains the target index ω for sequence t. We assume that $\theta_k(t) \sim Beta(\alpha_{t,k}, \beta_{t,k})$ with shape parameters $\alpha_{t,k}, \beta_{t,k}$. We sample a vector of $\{\hat{\theta}_k(t)\}$ from the above beta distributions and define $\boldsymbol{I}(t)$ as the indices of the L samples with the largest values. Let $z_{t,l}$ be the classifier score of stimulus group l during sequence t. Let $\mathcal{N}(\mu, \sigma^2)$ be a univariate normal distribution with mean μ and variance σ^2.

$$z_{t,l} \sim \begin{cases} \mathcal{N}(\mu_1, \sigma^2), & \omega \in S_l \\ \mathcal{N}(\mu_0, \sigma^2), & \omega \notin S_l. \end{cases} \tag{3.1}$$

In practice, such a simulation mechanism is equivalent to the complex process described in Section 3.4. Then, we compute the "clean" rewards for each character $n, n = 1, \cdots, N$.

$$
\begin{aligned}
P_{t,n} &= \frac{\prod_{l=1}^{L} l_{t,l,n}(z_{t,l}) P_{t-1,n}}{\sum_{c=1}^{N} \prod_{l=1}^{L} l_{t,l,c}(z_{t,l}) P_{t-1,c}}, \\
l_{t,l,n}(z_{t,l}) &= \begin{cases} l_0(z_{t,l}), & n \notin S_{I_l(t)}, \\ l_1(z_{t,l}), & n \in S_{I_l(t)}, \end{cases}
\end{aligned}
\tag{3.2}
$$

where l_1, l_0, and $\boldsymbol{P}_t$ are likelihood functions of $\mathcal{N}(\mu_1, \sigma^2), \mathcal{N}(\mu_0, \sigma^2)$, and the character-level probability vector after sequence t, respectively. We sum up the probabilities of which character indices belong to the stimulus group $S_k, k = 1, \cdots, K$. Let $\mathbb{I}_A(x)$ be the indicator function that equals 1 if $x \in A$, and 0 otherwise, where A is an ordinary set, then

$$
r_{t,k} = \sum_{n=1}^{N} P_{t,n} \cdot \mathbb{I}_{S_k}(n), \quad k = 1, \cdots, K. \tag{3.3}
$$

Finally, we update the shape parameters for each stimulus group $k, k = 1, \cdots, K$.

$$
\alpha_k \leftarrow \alpha_k + r_{t,k}, \quad \beta_k \leftarrow \beta_k + 1 - r_{t,k}. \tag{3.4}
$$

We repeat the above process until we reach the pre-specified stopping criterion. Algorithm 1 summarizes the adaptive stimulus selection paradigm.

3.3.4 Practical Constraints and Consideration

During the online BCI implementation, we consider two practical constraints: the response delay (RD) and the double flash (DF) issue. First, there is an inevitable response time delay between the presentation of the stimulus group during sequence t and its corresponding classifier score. On one hand, since the inputs of the online

Algorithm 1 A pseudo-code of our sequence-based adaptive stimulus selection. Here, we do not directly take response delay and double flash into consideration.

```
Input: The unit stimulus group set $\mathcal{S} = \{S_1, \cdots, S_K\}$, the subset size $L, (4 \leq L \leq K)$.
Output: Selected Indices $\boldsymbol{I}(t)$ and character-level probabilities $\boldsymbol{P}(t)$.
  for $k = 1, \cdots, K$ do
      Initialize $\alpha_k = 1, \beta_k = 1$.
  end for
  Initialize $\boldsymbol{P}_0$ with uniform probability.
  while $1 \leq t \leq T_0$ and $\max(\boldsymbol{P}_t) \leq p_{\max}$ do
      for $k = 1, \cdots, K$ do
          Sample $\hat{\theta}_k(t) \sim Beta(\alpha_k, \beta_k)$.
      end for
      $I(t)$ = Indices of top-$L$ stimulus groups ranked by $\{\hat{\theta}_k(t), k = 1, \cdots, K\}$.
      for $l = 1, \cdots, L$ do
          Observe $\{z_{t,l}\}$ for stimulus groups from the brain responses indexed by $\boldsymbol{I}(t)$.
          $\boldsymbol{P}_{t+1} \overset{NBDSA}{\longleftarrow} \boldsymbol{P}_t, \{z_{t,l}\}, l_0, l_1$ (See details in Eq.(3.2))
      end for
      for $k = 1, \cdots, K$ do
          $r_{t,k} = \sum_{n=1}^{N} P_{t,n} \cdot \mathbb{I}_{S_k}(n)$.
          $(\alpha_k, \beta_k) \leftarrow (\alpha_k + r_{t,k}, \beta_k + 1 - r_{t,k})$
      end for
  end while
```

implementation are streaming EEG recordings, in order to produce a classifier score, we first apply the bandpass filter then signal segmentation to obtain the extracted EEG signal segments, and finally the binary classifier so that the extracted EEG signal segments can be converted to the scalar values. These procedures require additional time. On the other hand, more importantly, under the CB paradigm, the response window of EEG signals (usually 800 ms) is much longer than the time interval between adjacent stimulus groups (usually shorter than 200 ms). The machine requires additional time to record a complete ERP response after the stimulus has been highlighted on the screen. We address the constraint of this response delay by applying a policy of the "cross iteration update" (See Section 3.4.3). In addition, although the offline CB paradigm avoids the impact of the double-flash issue, it may arise when we apply the algorithm to select a subset of stimulus events. Recall that the CB paradigm divides the entire keyboard into two hidden matrices, where the order of stimulus events are rows of the first hidden matrix, rows of the second hidden matrix, columns of the first hidden matrix, and columns of the second hidden matrix. Since the two target events are within the same hidden matrix, the target row and column are separated by rows and columns of another hidden matrix, and the time length should be sufficient to avoid the overlapping components of two target stimulus events. We address the constraint of the double flash issue only for the data generation mechanism by considering a third-level classifier score or ERP response functions.

Finally, we further improve the spelling efficiency by incorporating a word-level language model (LM). In Algorithm 1, before we start the Thompson sampling, we specify $\boldsymbol{P}_0$ with a uniform probability, which implies that we have no prior knowledge about the upcoming character to spell. The basic idea of the LM model is that given a dictionary of words, when each participant is spelling a new character of a meaningful word, we modify the $\boldsymbol{P}_0$ accordingly. Suppose that the participant wants to spell "THE" and the BCI speller system has correctly spelled the first two

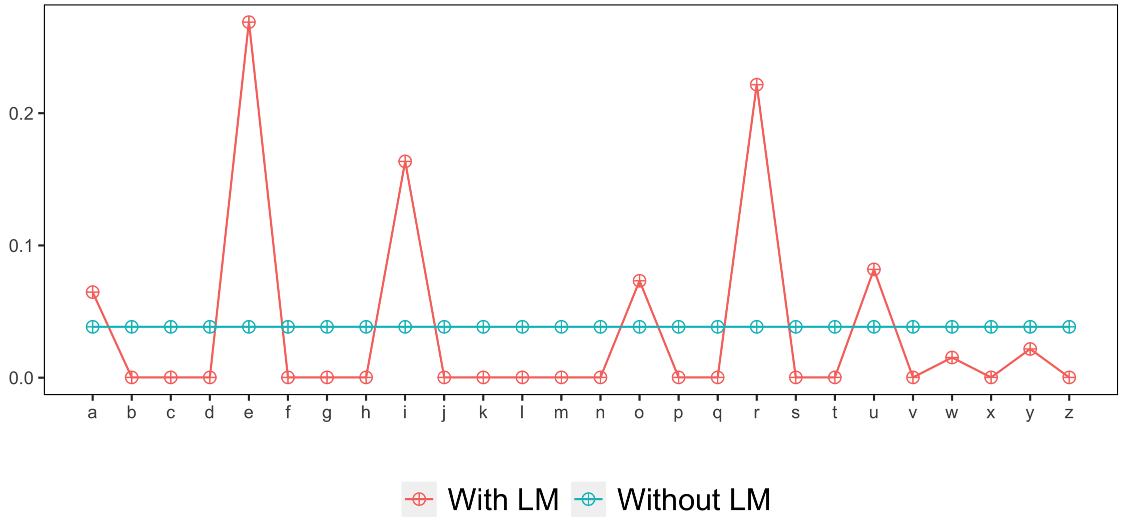


Figure 3.2: An illustration of specifying the initial character-level probability $\boldsymbol{P}_0$ using different prior specifications. Red and blue dots are $\boldsymbol{P}_0$ obtained by with LM method and without LM method, respectively. In particular, $\mathbb{P}\{E \mid \text{Without LM}\} = 0.038$, and $\mathbb{P}\{E \mid \text{With LM}\} = 0.269$. Thus, the specification with LM method potentially helps identify the true character "E" faster than the specification without LM method.

characters "TH." Instead of starting from scratch for the character "E," we compute the conditional probability of each character by searching the words starting with "TH," and the conditional probability is approximately by the frequency percentage within the subset of the words starting with "TH." We demonstrate the difference between two prior specifications in Figure 3.2. The probabilities of character E given no prior information and LM prior specification are 0.038 and 0.269, respectively. We report the results using the LM prior in Sections 3.4.5-3.4.6.

3.4 Experiments and Numerical Results

3.4.1 Experimental Setup

Since there is no available real data to evaluate the online stimulus selection algorithm, we report the numerical results based on extensive simulation studies to compare different configurations of our proposed algorithm to the conventional CB

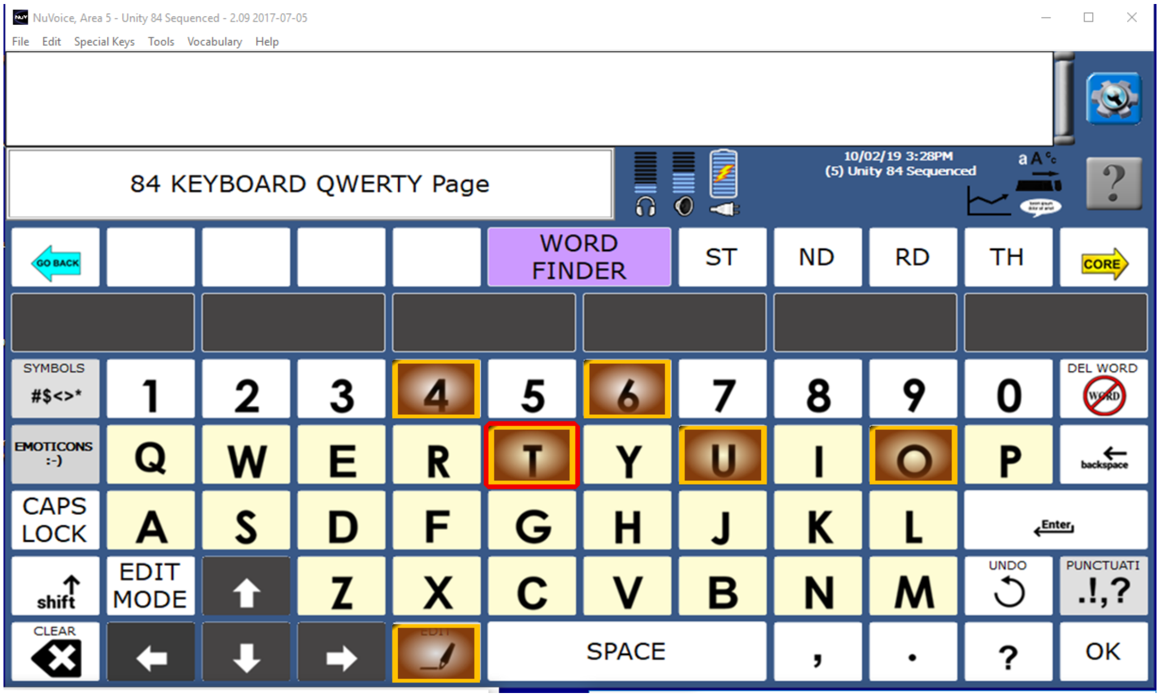


Figure 3.3: An illustration of an 84-key keyboard with a combination of characters, denoted as a stimulus group, being highlighted (*PRC-Saltillo*, 2009). Under the CB paradigm, stimulus groups are not limited to row and column flashes, and the virtual screen looks similar to the physical keyboard.

paradigm. We base the speller grid on Figure 3.3 (*PRC-Saltillo*, 2009), where the adjacent stimulus interval and the EEG response window are 160 ms and 800 ms, respectively. We let L range from 5 to 26. When $L = 5$, it satisfies two requirements: 5 is the minimum number of events that adjusts for the response delay and two target events within the reordered stimulus selection group are separated by at least one event. When $L = 26$, it is equivalent to the conventional CB paradigm because no subset selection process is involved. We design five scenarios: score-based without response delay (testing the algorithm under the most ideal circumstance), score-based with response delay, signal-based with response delay, score-based with response delay and LM prior, signal-based with response delay and LM prior; For Scenarios 2 to 5, we adjust for the response delay by randomly specifying the selected stimulus groups for the first two sequences.

For Scenarios 1, 2, and 4, we follow the framework in (*Kalika et al.*, 2017) by directly generating classifier scores of target and non-target stimulus groups from normal distributions $l_1(z)$ and $l_0(z)$, respectively. We define the following parameters,

$$d' = \frac{\mu_1 - \mu_0}{\sigma}, \tag{3.5}$$

where d' is the detectability index defined in (*Birdsall*, 1973); μ_1 and μ_0 are the mean parameters for $l_1(z)$ and $l_0(z)$, respectively; and σ is the standard deviation shared by both target and non-target stimulus groups. We vary the parameter d' and the subset size L to see how they affect the simulation results.

For Scenarios 3 and 5, we use the EEG signals as the signal input using Equation (2.1). For Scenario 3, we align the single-channel simulated ERP response function with stimulus-type indicators and add Gaussian noises with a variance σ_X^2 and an auto-correlation parameter q. Next, we extract the EEG signal segments from the onset of the stimulus group with a fixed EEG window as the feature vectors. Then, we apply the swLDA method to obtain the corresponding classifier scores. We assume that the swLDA model has been pre-trained from an additional dataset generated by the same simulated ERP responses and background noise structure. We vary the parameter σ_X^2 and subset size L to see how they affect the simulation results. For Scenario 5, we directly use the extracted multi-channel EEG signal segments from the calibration data of Participant A in Section 2.4.3.

We define the probability threshold $p_{\max} = 0.90$ and time threshold $t_0 = 5$ sequences. Since L is an input for the algorithm, we use the number of stimulus groups for consistency. We report the final accuracy and the number of stimuli to satisfy the above joint criteria. In addition, we report the information transfer rate (ITR) as in (*Wolpaw et al.*, 1998) and the BCI utility (BCIU) as in (*Dal Seno et al.*, 2009) to provide more comprehensive metrics that combine accuracy and speed. For Scenarios 1-3, we assume the participant is spelling one target character (e.g., character index of 70) without incorporating the prior information into any candidate characters, and we report the results for $L = 5, 10, 13$, and 26. For each scenario, we repeat different cases 500 times and report the average results with standard deviations across replications.

3.4.2 Score-based without Response Delay

Assuming no response delay, we perform simulation studies to compare the performance of subset selection to the conventional CB paradigm. We fix $\mu_1 = 0.50$ and $\mu_0 = -0.20$, and varied σ among $0.20, 0.30, 0.40, 0.50$, and 0.60, or equivalently, $d' = 3.50, 2.33, 1.75, 1.40$, and 1.17. Notice that the combination of $\mu_1 = 0.5, \mu_0 = -0.2$, and $\sigma = 0.3$ or $d' = 2.33$ is based on the result of a real participant, and we vary σ to demonstrate the robustness of our method. Table 3.1 shows the mean accuracy, number of stimulus groups needed, ITR, and BCIU of Scenario 1 under various σ across 500 replications. The results of Scenario 1 are considered as the upper bound of the performance of our algorithm. As the subset size L increases from 5 to 26, the probability of correctly selecting the target character, ITR and BCIU, while the number of stimulus groups to reach the stopping criterion increases. This suggests that our algorithm is more efficient than the conventional CB paradigm.

Table 3.1: The means and standard deviations of the evaluation metrics for the classifier score-based scenario without response delay (Scenario 1) across 500 replications. Metrics include the spelling accuracy, the number of stimulus groups, ITR, and BCIU. We vary the detectability index d' and the subset size L to examine the their effects on the simulation results.

d'	Subset Size L	ITR	BCIU	Accuracy	Stimulus Group
3.50	5	0.96 (0.45)	0.95 (0.44)	0.98 (0.05)	35 (15)
3.50	10	0.74 (0.26)	0.73 (0.26)	0.98 (0.05)	43 (15)
3.50	13	0.67 (0.21)	0.66 (0.21)	0.98 (0.07)	47 (16)
3.50	26	0.50 (0.08)	0.50 (0.08)	0.99 (0.02)	59 (12)
2.33	5	0.67 (0.31)	0.67 (0.31)	0.96 (0.11)	48 (22)
2.33	10	0.53 (0.19)	0.52 (0.19)	0.96 (0.12)	58 (21)
2.33	13	0.48 (0.16)	0.48 (0.16)	0.96 (0.09)	63 (21)
2.33	26	0.35 (0.09)	0.35 (0.09)	0.96 (0.11)	83 (21)
1.75	5	0.50 (0.23)	0.50 (0.23)	0.93 (0.15)	63 (27)
1.75	10	0.38 (0.15)	0.37 (0.15)	0.93 (0.17)	78 (25)
1.75	13	0.32 (0.13)	0.32 (0.13)	0.91 (0.20)	88 (25)
1.75	26	0.23 (0.10)	0.23 (0.10)	0.86 (0.24)	110 (21)
1.20	5	0.37 (0.18)	0.36 (0.18)	0.88 (0.23)	79 (28)
1.20	10	0.26 (0.14)	0.25 (0.14)	0.82 (0.28)	100 (27)
1.20	13	0.22 (0.12)	0.22 (0.12)	0.81 (0.29)	110 (23)
1.20	26	0.14 (0.10)	0.13 (0.11)	0.66 (0.35)	120 (13)
1.17	5	0.25 (0.16)	0.24 (0.17)	0.76 (0.33)	98 (28)
1.17	10	0.18 (0.12)	0.17 (0.13)	0.70 (0.35)	110 (21)
1.17	13	0.15 (0.12)	0.14 (0.13)	0.64 (0.36)	120 (19)
1.17	26	0.09 (0.08)	0.07 (0.09)	0.50 (0.34)	130 (6)

3.4.3 Score-based with Response Delay

We perform simulation studies to compare the performance of subset selection to the conventional CB paradigm with the same parameter set in Section 3.4.2. We assume that the data collection, classifier score generation, and posterior sampling associated with sequence j will be completed by the end of sequence $(j+1)$, denoted as

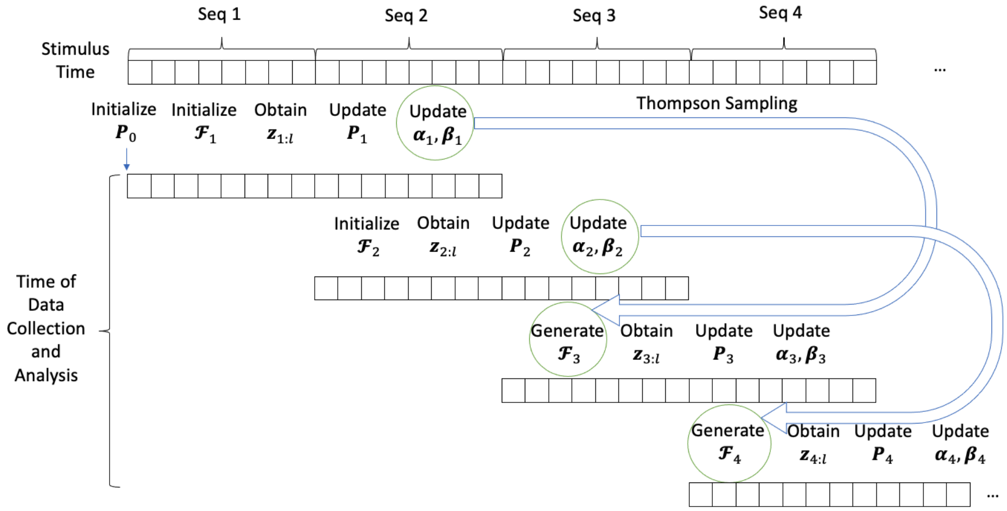

Seq 1
Seq 2
Seq 3
Seq 4
Stimulus Time
...
Initialize $\boldsymbol{P}_0$
Initialize $\boldsymbol{\mathcal{F}}_1$
Obtain $\boldsymbol{z}_{1:l}$
Update $\boldsymbol{P}_1$
Update $\boldsymbol{\alpha}_1, \boldsymbol{\beta}_1$
Thompson Sampling
Initialize $\boldsymbol{\mathcal{F}}_2$
Obtain $\boldsymbol{z}_{2:l}$
Update $\boldsymbol{P}_2$
Update $\boldsymbol{\alpha}_2, \boldsymbol{\beta}_2$
Time of Data Collection and Analysis
Generate $\boldsymbol{\mathcal{F}}_3$
Obtain $\boldsymbol{z}_{3:l}$
Update $\boldsymbol{P}_3$
Update $\boldsymbol{\alpha}_3, \boldsymbol{\beta}_3$
Generate $\boldsymbol{\mathcal{F}}_4$
Obtain $\boldsymbol{z}_{4:l}$
Update $\boldsymbol{P}_4$
Update $\boldsymbol{\alpha}_4, \boldsymbol{\beta}_4$
...

Figure 3.4: The simulated data generation mechanism with the response delay. The upper panel is the time frame for stimulus presentation, while the lower panel is the time frame of data collection and analysis for each sequence. Let $\boldsymbol{\mathcal{F}}_j$ be the set of stimulus groups for sequence j. We randomly initialize the stimulus groups for the first two sequences. We assume that the data collection, classifier score generation, and posterior sampling associated with sequence j will be completed by the end of sequence $(j+1)$, denoted as "cross iteration update". For sequence $j, j > 2$, we generate the stimulus groups based on posterior samples of TS, $\boldsymbol{\alpha} - 2, \boldsymbol{\beta} - 2$. The process is terminated when the stopping criterion is reached (not shown on the figure).

the "cross iteration update" (See Figure 3.4). Thus, we randomly initialize stimulus groups for the first two sequences. For sequence $j, j > 2$, we generate the stimulus groups based on the Thompson sampling results of the posterior samples $\boldsymbol{\alpha}_{j-2}$ and $\boldsymbol{\beta}_{j-2}$. Table 3.2 shows the mean accuracy, number of stimulus groups needed, ITR, and BCIU of Scenario 2 under various σ across 500 replications. Similar patterns to Table 3.1 are observed. The ITR, BCIU, accuracy, and the number of stimulus groups in Scenario 2 are, on average, lower than those in Scenario 1 because by assumption, we record the probability of the sequence prior to the one that reaches the stopping criterion. However, the differences between the results for Scenario 1 and Scenario 2 are moderate.

Table 3.2: The means and standard deviations of the evaluation metrics for the classifier score-based scenario with response delay (Scenario 2) across 500 replications. Metrics include accuracy, the number of stimulus groups, ITR, and BCIU. We apply the cross-iteration update policy to adjust for the response delay. We vary the detectability index d' and the subset size L to examine their effects on the simulation results.

d'	Subset Size L	ITR	BCIU	Accuracy	Stimulus Group
3.50	5	0.79 (0.27)	0.79 (0.26)	0.98 (0.02)	39 (11)
3.50	10	0.60 (0.14)	0.6 (0.14)	0.98 (0.03)	49 (12)
3.50	13	0.55 (0.09)	0.54 (0.09)	0.98 (0.02)	53 (9)
3.50	26	0.35 (0.04)	0.35 (0.04)	0.99 (0.02)	83 (10)
2.33	5	0.59 (0.19)	0.59 (0.19)	0.97 (0.03)	51 (15)
2.33	10	0.43 (0.12)	0.43 (0.12)	0.97 (0.03)	68 (18)
2.33	13	0.39 (0.10)	0.39 (0.10)	0.97 (0.05)	73 (18)
2.33	26	0.26 (0.06)	0.26 (0.07)	0.95 (0.13)	110 (18)
1.75	5	0.43 (0.15)	0.42 (0.15)	0.95 (0.10)	69 (22)
1.75	10	0.32 (0.10)	0.31 (0.10)	0.93 (0.14)	88 (22)
1.75	13	0.28 (0.09)	0.27 (0.10)	0.92 (0.15)	98 (22)
1.75	26	0.17 (0.09)	0.16 (0.10)	0.78 (0.29)	120 (12)
1.20	5	0.32 (0.13)	0.32 (0.13)	0.91 (0.16)	86 (24)
1.20	10	0.23 (0.11)	0.22 (0.12)	0.82 (0.27)	110 (22)
1.20	13	0.18 (0.10)	0.18 (0.11)	0.77 (0.30)	120 (18)
1.20	26	0.10 (0.08)	0.08 (0.09)	0.56 (0.34)	130 (5)
1.17	5	0.23 (0.13)	0.23 (0.14)	0.80 (0.29)	100 (24)
1.17	10	0.15 (0.11)	0.14 (0.12)	0.67 (0.34)	120 (17)
1.17	13	0.11 (0.10)	0.10 (0.11)	0.57 (0.35)	120 (11)
1.17	26	0.06 (0.06)	0.04 (0.06)	0.37 (0.31)	130 (0)

3.4.4 Signal-based with Response Delay

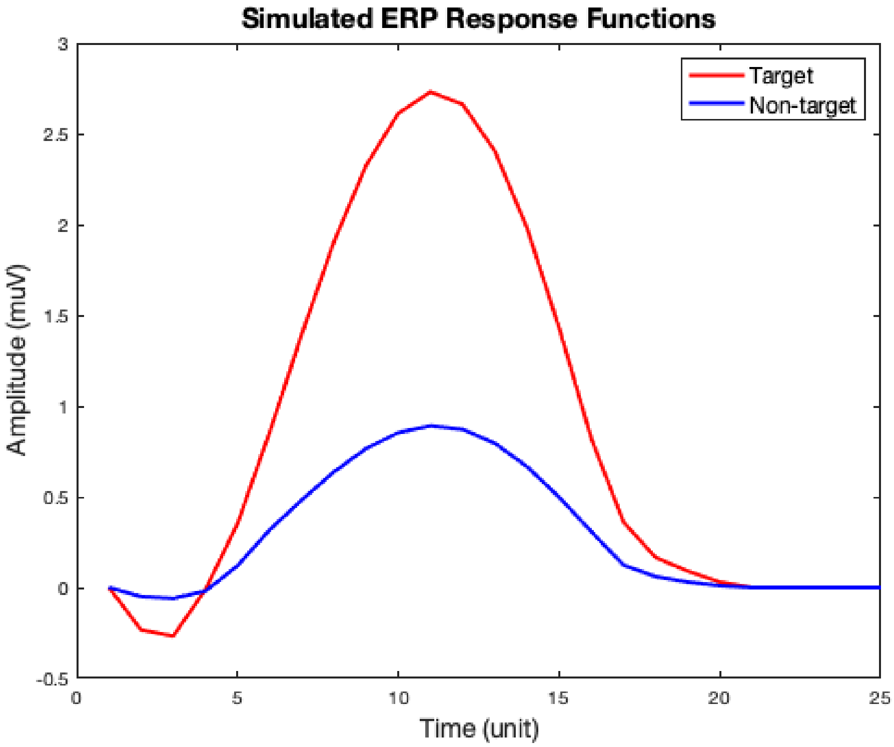


Figure 3.5: The simulated ERP response functions to target (red) and non-target (blue) in Scenario 3. Both ERP response functions contain 25 time points, with the unit time interval representing 32 ms. The peak ratio between target and non-target stimuli is around 3. We assume that the time interval between adjacent stimuli are 160ms, and the extracted EEG response window has the same length as the simulated ERP response functions.

We extend Scenario 2 where we start from the simulated EEG time series. We apply the same updating policy as in Section 3.4.3. We assume that the simulated EEG signals have an additive signal-and-noise effect. We align the simulated ERP responses based on the type of stimuli by the rule of convolution to form the signal component shown in Figure 3.5. We assume the noise component follows a Gaussian distribution with an auto-correlation structure $AR(q)$ and a variance σ_X^2. Here, we only consider a pair of single-channel ERP response functions without incorporating the spatial dependency structure. Then, we extract the EEG signal segments from the onset of each stimulus with the fixed response window as the feature vector, and convert them to the classifier scores using the swLDA weights. Here, we obtain the μ_1, μ_0, σ, and d' by computing the sample means and the sample variance of classifier scores after applying the swLDA weights. Thus, we fix the simulated ERP responses and the auto-correlation structure of $(0.5, 0)$, and vary the noise variance σ_X^2 among $\{0.01, 2, 5.5, 12.5, 20\}$ to match d' approximately for better comparison. Table 3.3 shows the mean accuracy, number of stimulus groups needed, ITR, and BCIU of Scenario 3. The actual d' here may deviate from the d' in Scenarios 1 and 2 due to the randomness in the training set.

Table 3.3: The means and standard deviations of the evaluation metrics for the single-channel signal-based scenario with response delay (Scenario 3) across 500 replications. Metrics include accuracy, the number of stimulus groups, ITR, and BCIU. We vary the noise variance σ_X^2 to match the detectability index d' and the subset size L. The actual d' here may be different from the values in Scenarios 1 and 2 due to the randomness in the training set.

d'	Subset Size L	ITR	BCIU	Accuracy	Stimulus Group
3.50	5	0.88 (0.32)	0.87 (0.32)	1 (<0.01)	37 (11)
3.50	10	0.69 (0.15)	0.68 (0.15)	1 (<0.01)	45 (11)
3.50	13	0.61 (0.08)	0.60 (0.08)	1 (<0.01)	49 (6)
3.50	26	0.38 (<0.01)	0.37 (<0.01)	1 (<0.01)	78 (0)
2.33	5	0.67 (0.21)	0.67 (0.21)	0.97 (0.03)	45 (13)
2.33	10	0.48 (0.10)	0.48 (0.10)	0.96 (0.03)	59 (13)
2.33	13	0.43 (0.08)	0.43 (0.08)	0.96 (0.03)	65 (13)
2.33	26	0.28 (0.04)	0.28 (0.05)	0.97 (0.05)	99 (16)
1.75	5	0.45 (0.12)	0.44 (0.12)	0.96 (0.03)	64 (15)
1.75	10	0.35 (0.09)	0.34 (0.09)	0.95 (0.08)	81 (19)
1.75	13	0.28 (0.07)	0.28 (0.07)	0.94 (0.08)	96 (19)
1.75	26	0.19 (0.08)	0.18 (0.09)	0.83 (0.24)	120 (13)
1.20	5	0.19 (0.13)	0.19 (0.14)	0.71 (0.35)	110 (22)
1.20	10	0.06 (0.08)	0.04 (0.08)	0.35 (0.33)	130 (7)
1.20	13	0.06 (0.07)	0.04 (0.07)	0.37 (0.31)	130 (4)
1.20	26	0.04 (0.06)	0.02 (0.05)	0.28 (0.28)	130 (1)
1.17	5	0.11 (0.12)	0.10 (0.12)	0.48 (0.39)	120 (16)
1.17	10	0.03 (0.05)	0.02 (0.05)	0.22 (0.27)	130 (2)
1.17	13	0.02 (0.03)	0.01 (0.03)	0.16 (0.20)	130 (1)
1.17	26	0.01 (0.02)	0.01 (0.01)	0.13 (0.15)	130 (0)

3.4.5 Score-based with Response Delay and LM Prior

We incorporate the LM prior into the score-based simulation scenario considering the response delay. We fix the combination of parameters $\mu_1 = 0.5, \mu_0 = -0.2$, and $\sigma = 0.5$ that is based on the training data from a participant. We choose the word "BIOSTATISTICS" such that the spelling efficiency could heavily benefit from properly initializing the character-level probability vector. We look at the subset selection size $L = 10$ and compare the results to those with $L = 26$, where no subset selection is involved. The upper and lower panels of Figure 3.6 show the character-level spelling accuracy and minimum number of sequence replications required to reach the stopping criterion, respectively, further stratified by algorithms with and without the LM prior. We observe that the average character-level accuracy with the LM prior is consistently higher than the average without the LM prior, and the average character-level sequence replication size is reduced by at least 50% since the second character. The upper panel of Table 3.4 shows the means and standard deviations of ITR and BCIU with $L = 10$ and 26, stratified by with and without LM methods. Given the same L, the average ITR and BCIU obtained with LM method are higher than those obtained without LM method; given the same prior specification method, the average ITR and BCIU are higher with smaller L.

3.4.6 Signal-based with Response Delay and LM Prior

Finally, we incorporate the language model prior into the signal-based simulation scenario considering the response delay. We use a pair of single-channel ERP response functions in Figure 3.5 to generate the simulated EEG signals with $\sigma_X^2 = 10$. We assume that the participant is spelling the same word as in Section 3.4.5. The upper and lower panels of Figure 3.7 show the character-level spelling accuracy and minimum number of sequence replications required to reach the stopping criterion, respectively, stratified by algorithms with and without the LM prior. We observe that the average

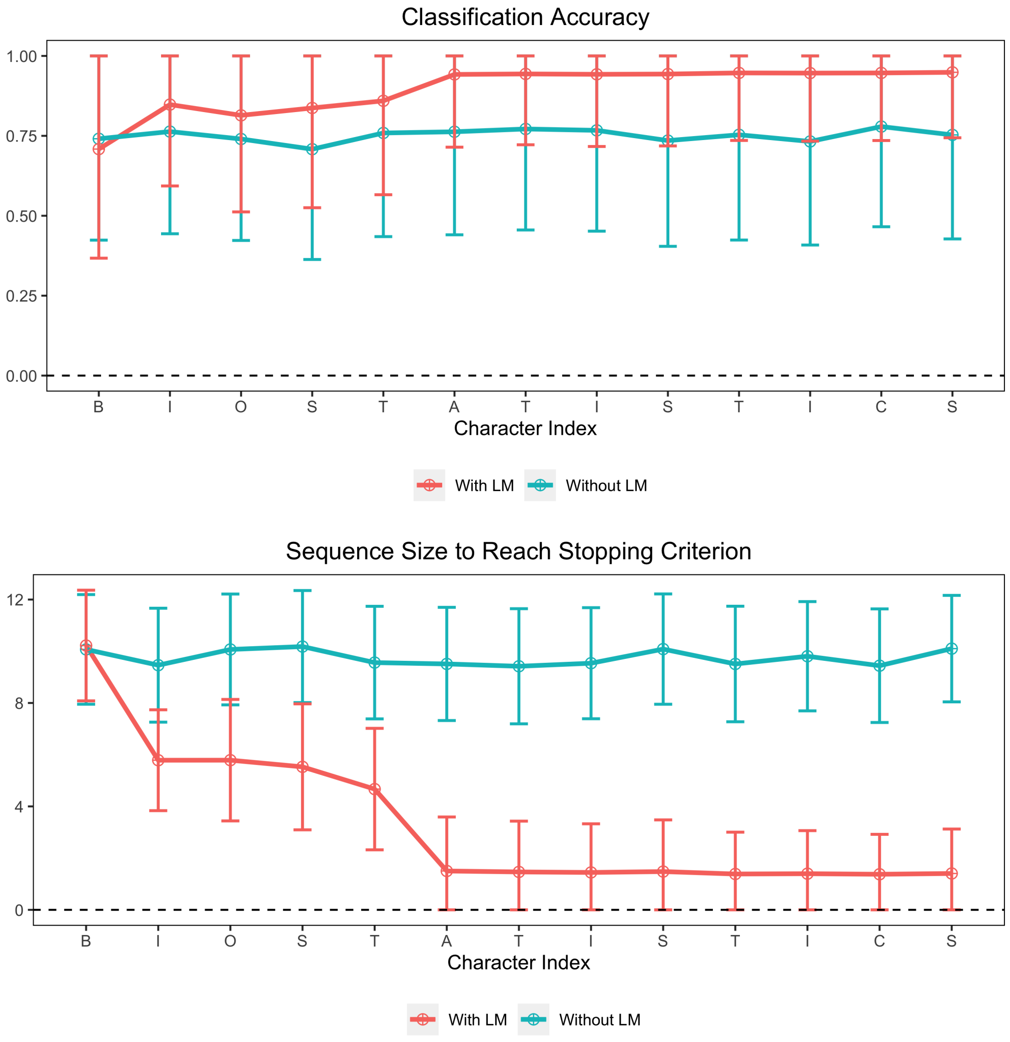


Figure 3.6: Improvements with the language model prior of the score-based simulation study with the response delay (Scenario 4). Parameters are specified based on the training data from a participant as follows: $\mu_1 = 0.5$, $\mu_0 = -0.2$, $\sigma = 0.5$ and subset size $L = 10$. The upper panel shows the average character-level spelling accuracy with the standard deviation across 500 replications, stratified by algorithms with and without the LM prior. The lower panel shows the average character-level minimum sequence replications to reach the stopping criterion with the standard deviation across 500 replications, stratified by algorithms with and without the LM prior. The average character-level accuracy with the LM prior is higher than the average accuracy without the LM prior, and the average character-level sequence replication size is reduced by at least 50%.

character-level accuracy with the LM prior is consistently higher than the average without the LM prior, and the average character-level sequence replication size is reduced by at least 40% starting from the second character. The lower panel of Table 3.4 shows the means and standard deviations of ITR and BCIU with $L = 10, 26$, stratified by with and without LM methods. We observe similar pattern to the results in Section 3.4.5.

Table 3.4: The means and standard deviations of ITR and BCIU for score-based (Scenario 4, upper panel) and signal-based (Scenario 5, lower panel) scenarios adjusting for response delay across 500 replications. We look at the subset size $L = 10$ and 26, where $L = 26$ is the baseline case that no subset selection is involved. We observe similar patterns in both scenarios. Given the same L, the average ITR and BCIU obtained with LM prior are higher than those obtained without LM prior; given the same prior specification, the average ITR and BCIU are higher with smaller L.

	ITR		**BCIU**	
L	with LM	without LM	with LM	without LM
10	0.69 (0.16)	0.22 (0.04)	0.68 (0.17)	0.21 (0.05)
26 (Baseline)	0.35 (0.08)	0.12 (0.04)	0.35 (0.09)	0.09 (0.05)

	ITR		**BCIU**	
L	with LM	without LM	with LM	without LM
10	0.90 (0.06)	0.38 (0.03)	0.89 (0.06)	0.38 (0.03)
26 (Baseline)	0.43 (0.03)	0.23 (0.03)	0.43 (0.03)	0.22 (0.03)

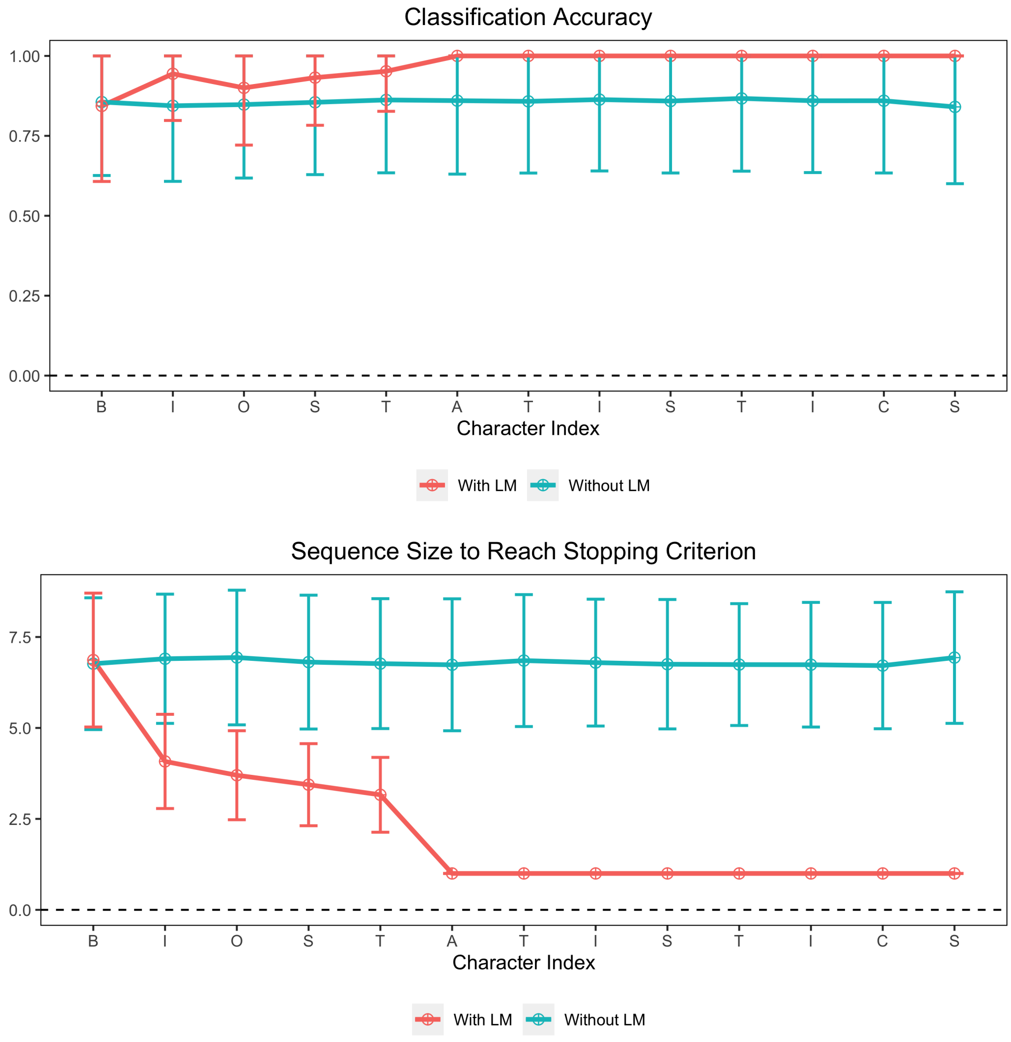


Figure 3.7: Improvements with the language model prior of the signal-based simulation study with the response delay (Scenario 5). A pair of single-channel ERP response functions in Figure 3.5 are used to generate the simulated EEG signals with $\sigma_X^2 = 10$. The subset size L is 10. The upper panel shows the average character-level spelling accuracy with the standard deviation across 500 replications, stratified by algorithms with and without the LM prior. The lower panel shows the average character-level minimum sequence replications to reach the stopping criterion with the standard deviation across 500 replications, stratified by algorithms with and without the LM prior. The average character-level accuracy with the LM prior is higher than the average accuracy without the LM prior, and the average character-level sequence replication size is reduced by at least 40%.

CHAPTER IV

Bayesian Semi-supervised Classification for Data Integration in ERP-based Brain-Computer Interfaces

4.1 Introduction

4.1.1 Challenges and Existing Work

Before participants performed the free-typing sessions, they copied a multi-character phrase to build a participant-specific training profile. We referred to the previous two procedures as testing and calibration, respectively. Since the signal-to-noise ratio (SNR) of EEG signals was generally low, the current calibration strategy simply collected data from participants themselves with a fixed but large amount of sequence replications. However, a lengthy calibration procedure might cause attention shift and mental fatigue. Participants tended to feel bored and might elicit biased target P300 ERPs, which made the calibration process inaccurate and inefficient. Therefore, the challenge became how to reduce participants' calibration time with decent prediction accuracy for the free-typing sessions. Existing work has tackled this problem by applying the idea of transfer learning, which was originally introduced by *Bozinovski and Fulgosi* in 1976, that information was extracted and stored from existing problems to solve a new but similar problem. In statistics, we denoted this concept as data integration (*Lenzerini*, 2002).

This subsection reviews existing works that have explored transfer learning under different domains for information leveraging with applications to P300 ERP-BCIs. First, the general ensemble learning method was an intuitive idea to incorporate data from other domains that combined the results of different classifiers within the same training set. Each classifier made predictions on a test set, and the results were combined with a voting process. *Rakotomamonjy and Guigue* in 2008 and *Johnson and Krusienski* in 2009 applied the ensemble method to P300 ERP-BCIs by averaging the outputs of multiple SVMs and swLDAs, respectively. Their base binary classifiers were trained on a small part of the available data. *Völker et al.* in 2018 and *Onishi* in 2020 applied the ensemble method by averaging the outputs of multiple CNNs to visual and auditory P300 ERP-BCI datasets, respectively. *Onishi and Natsume* also mentioned that ensemble methods with overlapping partitioning criterion yielded better prediction performance than the ensemble methods with a naive partitioning criterion.

As an alternative solution, *Xu et al.* in 2015 proposed the *Ensemble Learning Generic Information* (ELGI), which combined the data of the new participant with the data of source participants to form a hybrid ensemble. They split the data of each source participant into target and non-target subsets. They applied the swLDA method to construct the base classifiers by combining different subsets as follows: the target and non-target subsets from the new participant, the target subset from the new participant and the non-target subset from each source participant, and the target subset from each source participant and the non-target subset from the new participant. Thus, the resulting ensemble had $(2N + 1)$ base classifiers, where N is the number of source participants. They further introduced the *Weighted Ensemble Learning Generic Information* (WELGI) by adding weights to each base classifier

(*Xu et al.*, 2016). Similarly, *An et al.* in 2020 proposed a weighted participant-semi-independent classification method (WSSICM) for P300 ERP-based BCIs, where they used the SVM method as the base classifier. The base classifier was fit by combining the entire data of each source participant and a small portion of data of the new participant. An ad-hoc approach was applied to determine the weighted coefficients of base classifiers for participant selection. Likewise, *Adair et al.* in 2017 proposed the *Evolved Ensemble Learning Generic Information* (eELGI). The authors argued that grouping training sets by participants was not an optimal selection criterion. Instead, they developed an evolutionary algorithm by permuting datasets among source participants to form the base classifiers, which were constructed using the swLDA method.

Finally, Riemannian geometry has gained increasing attention recently due to its fast speed to converge and a natural framework to leverage information from source participants. *Rodrigues et al.* in 2018 presented a transfer learning approach to tackle the heterogeneity of EEG signals across different sessions or participants using the Riemannian procrustes analysis (RPA). Before the authors applied the MDM classifier, they applied certain affine transformations to raw participant-level covariance matrices such that the resulting covariance matrices were less *heterogeneous* across sessions or participants while their Riemannian distances were preserved. *Li et al.* in 2020 also *standardized* the covariance matrices across participants by applying the affine transformation with the participant-specific Riemannian geometric mean covariance matrix. Finally, *Khazem et al.* in 2021 proposed another transfer learning approach, denoted as Minimum Distance to Weighted Mean (MDWM). They combined estimated mean covariance matrices from source participants and the new participant by the Riemannian distance. They controlled the trade-off between new and source contribution by the power parameter, but they treated them as a hyper-parameter and did not estimate it during the calibration session. Although most existing methods applied transformations to make covariance matrices robust among source participants, *Khazem et al.* emphasized that further improvement in prediction accuracy could still benefit from a proper selection of source participants.

4.1.2 Our Contributions

To reduce the calibration time while maintaining similar prediction accuracy, we propose a BAyesian SemI-supervised Classification (BASIC) method to build a participant-dependent, calibration-less framework. To clarify, calibration-less methods refer to those methods that train the classifier with fewer sequence replications than existing strategy that uses a fixed amount of sequence replications from the new participants themselves. For calibration methods without using any training data from the new participant, we refer to them as calibration-free methods. In addition, the semi-supervised framework here is slightly different from the regular setting. On the stimulus-level, it is a supervised learning problem because the stimulus labels are known during calibration process; on the participant-level, it is a regular semi-supervised learning problem because we do not observe the participant labels among the pool of source participants. The BASIC framework reduces the calibration time of a new participant by borrowing data from pre-existing source participants' pool. Instead of performing transformation to make all source calibration data similar, we borrow data on the level of participants, which is an intuitive clustering criterion. BASIC specifies the joint distribution of stimulus-specific EEG signals from all participants via a Bayesian hierarchical mixture model.

Our method has advantages from both inference and prediction perspectives. First, unlike the conventional clustering approach, we specify the baseline cluster as the one that matches the new participant. Therefore, our method has a flavor of semi-supervised learning approach, and the selection indicators has the interpretation of how close source calibration data resembles the new data on the level of participants. Second, we test on the baseline cluster directly without refitting the model with the augmented data. Finally, our proposed hierarchical framework is flexible and can be extended to other classifiers with consistent parametric forms.

4.2 Methods

4.2.1 Problem Setup

Suppose the dataset consists of N source participants and one new participant. Let $n = 0, \ldots, N$ be the participant index, where $n = 0$ and $n = 1, \ldots, N$ refer to the new participant and the source participants, respectively; Let $l = 1, \ldots, L_n$ and $i = 1, \ldots, I_n$ be the character index and sequence index for participant n, respectively. We follow the conventional RCP design such that each sequence contains $J (J = 12)$ stimuli, including six row stimuli from top to bottom $(1, \ldots, 6)$ and six column stimuli from left to right $(7, \ldots, 12)$ on the 6×6 virtual keyboard (See Figure 1.3). For the ith sequence, lth target character, and nth participant, let $\boldsymbol{W}_{n,l,i} = (W_{n,l,i,1}, \ldots, W_{n,l,i,12})^\top$ be a stimulus code indicator that takes values from the permutation of $\{1, \ldots, 12\}$. Under the RCP design, given the target character and the stimulus-code indicators, there are exactly two target stimuli and ten non-target stimuli within each sequence, and we can identify the indices of target stimuli. We define $\boldsymbol{Y}_{n,l,i} = (Y_{n,l,i,1}, \ldots, Y_{n,l,i,12})^\top$ as the stimulus-type indicator for the ith sequence, lth target character, and nth participant, where $Y_{n,l,i,j} \in \{0, 1\}$. For example, given a target character "T" and a stimulus-code indicator $\boldsymbol{W}_{n,l,i} = (7, 9, 10, 5, 1, 2, 8, 11, 6, 4, 3, 12)^\top$, we obtain its corresponding stimulus type indicator $\boldsymbol{Y}_{n,l,i} = (0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0)^\top$, where the indices of 1s correspond to the indices of the 4th row and 2nd column (denoted by 8) in $\boldsymbol{W}_{n,l,i}$. Let $k = 0, \ldots, K-1$ be the cluster index, where $k = 0$ is the cluster that matches the new participant and $k = 1, \ldots, K-1$ are the clusters within source participants. Finally, we incorporate E channels of EEG signals and let $e(e = 1, \dots, E)$ be the channel index. We extract channel-specific EEG segments from the onset of each stimulus with a long EEG response window length T_0. We denote $\boldsymbol{X}_{n,l,i,j,e}(t)$ as the extracted EEG signal segment of the jth stimulus, ith sequence, lth target character, and nth participant from channel e at time $t \in [0, T_0]$.

For the rest of this chapter, let $\mathcal{MN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ be a multivariate normal distribution with mean and covariance matrix parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. Let $\mathcal{MN}(\boldsymbol{M}, \boldsymbol{U}, \boldsymbol{V})$ be a matrix normal distribution with location matrix M and two scale matrix parameters U and V (*Dawid*, 1981). Let $\text{Diag}(\cdot)$ be a diagonal matrix notation. Let $\mathcal{GP}(\mu, \kappa)$ be a Gaussian process with a mean function and kernel μ and κ. Let $\mathcal{LN}(\mu, \sigma)$ be a Log-Normal distribution with mean and scale parameters μ and σ. Let $\mathcal{HC}(x_0, \sigma)$ be a Half-Cauchy distribution with location and scale parameters x_0 and σ. Let $\mathcal{U}(a, b)$ be a Uniform distribution with lower and upper bounds a and b. Let $\text{Dirichlet}(\boldsymbol{\alpha})$ be a Dirichlet distribution with the concentration parameter $\boldsymbol{\alpha}$.

4.2.2 A Bayesian Semi-supervised Model

Given sufficient training data from the source participants and a small amount of training data from the new participant, our primary goal is to reduce calibration time by integrating a proper subset of training data on the level of source participants into the augmented training data for the new participant. We make three assumptions to simplify the model representation as follows: First, let all source participants spell the same character ω with the same sequence replication size I, so we drop the target character index l and the source participant index n from I_n. Second, we assume that target and non-target ERP functions share the same covariance matrix within the same cluster. Finally, we only borrow the target ERP data from from the source participants' pool and completely ignore the corresponding non-target ERP data.

By dropping the character index l and assuming that all the EEG signals are extracted from the onset of the stimulus with a fixed response window of T_0 points. We let $\boldsymbol{X}_{n,i,j}$ and $Y_{n,i,j} \in \{0,1\}$ denote the extracted EEG signal matrix of dimension $T_0 \times E$ and the binary stimulus-type indicator associated with the jth stimulus, ith sequence, and nth participant, respectively.

For the new participant, i.e., when $n = 0$, we assume

$$\boldsymbol{X}_{0,i,j} = \boldsymbol{B}_{0,1} Y_{0,i,j} + \boldsymbol{B}_0 (1 - Y_{0,i,j}) + \boldsymbol{\epsilon}_{0,i,j}. \tag{4.1}$$

where $\boldsymbol{B}_{0,1}$, $\boldsymbol{B}_0$ and $\boldsymbol{\epsilon}_{0,i,j}$ are the target ERP matrix, non-target ERP matrix and random noise matrix of the new participant.

For the nth source participant, i.e., $n \geq 1$, we introduce the latent cluster indicator $Z_n \in \{0, 1, \ldots, K-1\}$ and assume

$$\Pr(Z_n = k) = \pi_k, \tag{4.2}$$

where π_k represents the prior probability of the nth source participant belonging to the kth cluster. With the cluster indicator Z_n, we consider a clustering model for the EEG signal matrix for the nth source participant as follows:

$$\boldsymbol{X}_{n,i,j} = \sum_{k=0}^{K-1} I(Z_n = k)\{\boldsymbol{B}_{k,1} Y_{n,i,j} + \boldsymbol{B}_{k,0}(1 - Y_{n,i,j})\} + \boldsymbol{\epsilon}_{n,i,j}, \tag{4.3}$$

where $\boldsymbol{B}_{k,1}$ and $\boldsymbol{B}_{k,0}$ are the target ERP matrix and the non-target ERP matrix of the source participants in the kth cluster. The term $\boldsymbol{\epsilon}_{n,i,j}$ represents the random noise matrix. Of note, (4.1) and (4.3) assume the source participants in cluster 0 have the same target ERP matrices as the new participant, in which case we can borrow their data to make inferences on $\boldsymbol{B}_{0,1}$. However, our models do not imply that $\boldsymbol{B}_{0,0}$ is equal to $\boldsymbol{B}_0$. That is, we do not assume that non-target ERP signals of the source participants in cluster 0 are the same as those of the new participant. This further implies that the non-target EEG signals, i.e., $\{\boldsymbol{X}_{n,i,j} : Y_{n,i,j} = 0\}$, are not needed for making inferences on $\boldsymbol{B}_0$ and $\boldsymbol{B}_{0,1}$.

For the new participant and each of the source participants, i.e., when $n \geq 0$, we consider an additive model to characterize the spatial-temporal correlation of random noises for the nth participant ith sequence jth stimulus, i.e.,

$$\begin{aligned} \boldsymbol{\epsilon}_{n,i,j} &= \boldsymbol{\xi}_{n,i,j} + \boldsymbol{\varepsilon}_{n,i,j}, \\ \boldsymbol{\xi}_{n,i,j} &= (\boldsymbol{\xi}_{n,i,j,1}, \ldots, \boldsymbol{\xi}_{n,i,j,T_0})^\top, \\ \boldsymbol{\varepsilon}_{n,i,j} &= (\boldsymbol{\varepsilon}_{n,i,j,1}, \ldots, \boldsymbol{\varepsilon}_{n,i,j,E}) \end{aligned} \tag{4.4}$$

where $\boldsymbol{\xi}_{n,i,j,t}, (t = 1, \ldots, T_0)$ is a vector of spatial random effects for E channels at time point t and $\boldsymbol{\varepsilon}_{n,i,j,e}(e = 1, \ldots, E)$ is a vector of temporal random effects for T_0 time points of the channel e. Given $Z_n = k$, we assume $\boldsymbol{\xi}_{n,i,j,t}$ follows a multivariate normal distribution with a zero mean vector and a covariance matrix $\boldsymbol{V}_k \boldsymbol{\Sigma}_k^s \boldsymbol{V}_k^T$, where $\boldsymbol{\Sigma}_k^s$ is a correlation matrix with the compound symmetry structure characterized by a scale parameter $\eta_k \in (0, 1)$ and $\boldsymbol{V}_k = \text{Diag}(\sigma_{k,1}, \ldots, \sigma_{k,E})$. We assume $\boldsymbol{\varepsilon}_{n,i,j,e}$ follows a multivariate normal distribution with a zero mean vector and a temporal correlation matrix $\boldsymbol{\Sigma}_k^t$, where $\boldsymbol{\Sigma}_k^t$ has the structure of the exponential decay characterized by a scale parameter $\rho_k \in (0, 1)$. Furthermore, we assume that within each cluster, the temporal dependence ρ_k is shared across E channels with channel-specific variability. In summary,

$$\begin{aligned} (\boldsymbol{\xi}_{n,i,j,t} \mid Z_n = k) &\sim \mathcal{MVN}(\boldsymbol{0}, \boldsymbol{V}_k \boldsymbol{\Sigma}_k^s \boldsymbol{V}_k^\top), \\ (\boldsymbol{\varepsilon}_{n,i,j,e} \mid Z_n = k) &\sim \mathcal{MVN}(\boldsymbol{0}, \boldsymbol{\Sigma}_k^t), \end{aligned} \tag{4.5}$$

Let $\boldsymbol{\Theta}$ be a collection of all the unknown parameters, where the primary parameters of interest are those associated with the new participant, i.e., $\boldsymbol{B}_{0,1}, \boldsymbol{B}_0, \sigma_{0,e}^2, \boldsymbol{\Sigma}_0^t$, and $\boldsymbol{\Sigma}_0^s$. And we are also interested in making inference on the posterior probability of matching the new participant, i.e. $\Pr(Z_n = 0 \mid \boldsymbol{X}, \boldsymbol{Y})$, for each source participant. The remaining parameters are considered as nuisance parameters.

4.3 Posterior Inferences

4.3.1 Model Reparametrization and Prior Specifications

We rewrite the model by applying the notation of a matrix normal distribution

$$\begin{aligned} \boldsymbol{X}_{n,i,j} \mid Z_n = k, Y_{n,i,j} = 1; \boldsymbol{\Theta} &\sim \mathcal{MN}(\boldsymbol{B}_{k,1}, \boldsymbol{V}_k \boldsymbol{\Sigma}_k^s \boldsymbol{V}_k^\top, \boldsymbol{\Sigma}_k^t), \quad n > 1, \\ \boldsymbol{X}_{n,i,j} \mid Y_{n,i,j} = 1; \boldsymbol{\Theta} &\sim \mathcal{MN}(\boldsymbol{B}_{0,1}, \boldsymbol{V}_0 \boldsymbol{\Sigma}_0^s \boldsymbol{V}_0^\top, \boldsymbol{\Sigma}_0^t), \quad n = 0, \\ \boldsymbol{X}_{n,i,j} \mid Y_{n,i,j} = 0; \boldsymbol{\Theta} &\sim \mathcal{MN}(\boldsymbol{B}_0, \boldsymbol{V}_0 \boldsymbol{\Sigma}_0^s \boldsymbol{V}_0^\top, \boldsymbol{\Sigma}_0^t), \quad n = 0. \end{aligned} \tag{4.6}$$

Let $\boldsymbol{\Sigma}_k^{ts} = \boldsymbol{\Sigma}_k^t \otimes \boldsymbol{V}_k \boldsymbol{\Sigma}_k^s \boldsymbol{V}_k^\top$, then equation (4.6) can be further expressed as

$$\begin{aligned} \text{vec}(\boldsymbol{X}_{n,i,j}) \mid Z_n = k, Y_{n,i,j} = 1; \boldsymbol{\Theta} &\sim \mathcal{MVN}\left(\text{vec}(\boldsymbol{B}_{k,1}), \boldsymbol{\Sigma}_k^{ts}\right), \quad n > 1, \\ \text{vec}(\boldsymbol{X}_{n,i,j}) \mid Y_{n,i,j} = 1; \boldsymbol{\Theta} &\sim \mathcal{MVN}\left(\text{vec}(\boldsymbol{B}_{0,1}), \boldsymbol{\Sigma}_0^{ts}\right), \quad n = 0, \\ \text{vec}(\boldsymbol{X}_{n,i,j}) \mid Y_{n,i,j} = 0; \boldsymbol{\Theta} &\sim \mathcal{MVN}\left(\text{vec}(\boldsymbol{B}_0), \boldsymbol{\Sigma}_0^{ts}\right), \quad n = 0. \end{aligned} \tag{4.7}$$

where $\otimes$ is the Kronecker product operator and vec$(\cdot)$ is the vectorization operator that concatenates the matrix by its columns. To specify the prior models for the ERP signal matrices, we write $\boldsymbol{B}_{k,1} = (\boldsymbol{\beta}_{k,1,1}, \ldots, \boldsymbol{\beta}_{k,1,E})$ and $\boldsymbol{B}_0 = (\boldsymbol{\beta}_{0,1}, \ldots, \boldsymbol{\beta}_{0,E})$. For the cluster-and-channel-specific target ERP function $\boldsymbol{\beta}_{k,1,e}$, we assign a Gaussian process prior $\mathcal{GP}(0, \kappa_1)$ and a kernel variance parameter $\psi_{k,1,e}$; for the channel-specific non-target ERP function $\boldsymbol{\beta}_{0,e}$, we assign a Gaussian process prior $\mathcal{GP}(0, \kappa_0)$ and a kernel variance parameter $\psi_{0,e}$. We consider a γ-exponential kernel function to specify κ_1 and κ_0 as follows:

$$k(z_i, z_j) = \psi_0 \exp\left\{ - \left(\frac{||z_i - z_j||_2^2}{s_0} \right)^{\gamma_0} \right\}, \tag{4.8}$$

where $0 \leq \gamma_0 < 2, s_0 > 0$. In practice, we treat them as hyper-parameters and determine the optimal ones by Bayes factors (*Kass and Raftery*, 1995). For kernel variance parameters $\psi_{k,1,e}$ and $\psi_{0,e}$, we adopt Log-Normal priors with the a mean zero and a scale one. For cluster-channel-specific variance intensity parameters, we assign Half-Cauchy priors with a mean zero and a scale one. For cluster-specific temporal and spatial dependency parameters ρ_k and η_k, we assign uniform priors. For the prior weight $\boldsymbol{\pi} = (\pi_0, \ldots, \pi_{K-1})^\top$, we assign a Dirichlet distribution with a concentration parameter $\alpha \mathbf{1}_K/K$, where $\mathbf{1}_K$ is a K-dimensional vector of 1s. Finally, the prior specifications are summarized as follows:

$$
\begin{aligned}
\boldsymbol{\beta}_{k,1,e} &\sim \mathcal{GP}(0, \psi_{k,1,e}\kappa_1), \quad \boldsymbol{\beta}_{0,e} \sim \mathcal{GP}(0, \psi_{0,e}\kappa_0), \\
\psi_{k,1,e} &\sim \mathcal{LN}(0,1), \quad \psi_{0,e} \sim \mathcal{LN}(0,1), \\
\sigma_{k,e} &\sim \mathcal{HC}(0, 5.0), \quad \rho_k \sim \mathcal{U}(0,1), \quad \eta_k \sim \mathcal{U}(0,1), \\
\boldsymbol{\pi} &\sim \text{Dirichlet}(\alpha \mathbf{1}_K/K).
\end{aligned}
\tag{4.9}
$$

4.3.2 Markov Chain Monte Carlo

We apply Gibbs sampling to draw posterior samples of $\boldsymbol{\beta}_{k,1,e}$,$\boldsymbol{\beta}_{0,e}$, $\psi_{k,1,e}$,$\psi_{0,e}$,$\sigma^2_{k,e}$, ρ_k, and η_k. For the convergence check, we run multiple chains with different seed values and evaluate the convergence by the Gelman-Rubin statistic (*Gelman and Rubin*, 1992). For cluster-specific, multi-channel ERP functions $\boldsymbol{\beta}_{k,1,e}$, we establish the relationship with $\boldsymbol{\alpha}_{k,1,e}$, $\boldsymbol{\phi}_{k,1,e}$, and $\psi_{k,1,e}$ by

$$
\text{vec}(\boldsymbol{B}_{k,1}) := \begin{pmatrix} \boldsymbol{\beta}_{k,1,1} \\ \vdots \\ \boldsymbol{\beta}_{k,1,e} \\ \vdots \\ \boldsymbol{\beta}_{k,1,E} \end{pmatrix} = \left(\boldsymbol{D}_{\boldsymbol{\psi}_{k,1}} \otimes \boldsymbol{I}_{N_{\boldsymbol{\Psi}_{k,1}}} \right) (\boldsymbol{I}_E \otimes \boldsymbol{\Psi}_{k,1}) \begin{pmatrix} \boldsymbol{\alpha}_{k,1,1} \\ \vdots \\ \boldsymbol{\alpha}_{k,1,e} \\ \vdots \\ \boldsymbol{\alpha}_{k,1,E} \end{pmatrix},
$$

where $\boldsymbol{D}_{\boldsymbol{\psi}_{k,1}} = \mathrm{Diag}(\psi_{k,1,1}, \ldots, \psi_{k,1,E})$, $\boldsymbol{I}$ is the identity matrix, $N_{\boldsymbol{\Psi}_{k,1}}$ is the number of eigenvalues associated with $\boldsymbol{\Psi}_{k,1}$. We have a similar expression for $\mathrm{vec}(\boldsymbol{B}_0)$.

We also improve convergence by incorporating an iterative update procedure to the prior specification and initialization. In practice, the data of the new participant arrive in a sequential way, while the data of source participants are assumed available prior to the calibration procedure. For example, we fit the Bayesian generative method to each source participant to obtain participant-specific base classifiers, then we apply the criterion such as K-means clustering algorithm to build K clusters and use those cluster-specific model parameters for initialization. In addition, before another sequence of calibration data of the new participant arrives, we set the initial values by the posterior samples from previous sequences' calibration data.

4.3.3 Remedy for the Label Switching Issue

A common problem of parameter estimation in a Bayesian finite mixture model is known as "label switching." The label switching problem arises due to the invariance of the likelihood functions under relabelling the mixture components. The posterior distributions of parameters tend to become symmetric and multi-modal, making it difficult to summarize (*Stephens*, 2000). Although we specify the cluster 0 to match the new participant, the identification of remaining clusters in our original clustering approach is still susceptible to label switching issue. To resolve this problem, we adopt the log-likelihood function, which is a unit-less measurement and independent of the shape and magnitude of ERP functions, to rank the fitted clusters. Let $\boldsymbol{\Theta}_k$ and $\mathcal{P}_k = \{n : Z_n = k\}$ be the parameter set and the indices of the participants associated with cluster k, respectively. Let $l_k(\mathcal{P})$ be the log-likelihood values of the data belonging to the participant set $\mathcal{P}$ fitted with model parameters associated with cluster k. In particular, let $l_{k_1}(\mathcal{P}_{k_2})$ be the log-likelihood values of the source participants' data belonging to cluster k_2 fitted with the model parameters associated with cluster k_1. At the end of each MCMC iteration, we perform a two-step check as follows: First, to ensure that cluster 0 always corresponds to the cluster that matches the new participant, we compute $l_k(\{0\}), k = 0, \cdots, K-1$, the cluster-specific log-likelihood values with data from the new participant only. We denote the cluster index associated with the maximum value as $k_{\max}$. If $k_{\max} = 0$, we proceed directly to the next step; otherwise, we further compute $l_{\text{old}} = l_0(\mathcal{P}_0) + l_{k_{\max}}(\mathcal{P}_{k_{\max}})$ and $l_{\text{new}} = l_0(\mathcal{P}_{k_{\max}}) + l_{k_{\max}}(\mathcal{P}_0)$. If $l_{\text{new}} > l_{\text{old}}$, we swap the cluster indices between 0 and $k_{\max}$; otherwise, we proceed directly to the next step. Second, we rank the remaining clusters by the descending order of $l_k(\{0\}), k = 1, \ldots, K-1$. Therefore, the closer cluster k has to the new participant's data, the smaller k is.

4.3.4 Posterior Character-Level Prediction

Under the RCP design, the character-level prediction depends on selecting the correct row and column within each sequence. To simplify, let $\boldsymbol{W}, \boldsymbol{Y}$, and $\boldsymbol{X}$ be the existing stimulus-code indicators, the stimulus-type indicators, and the matrix-wise EEG signals, respectively. Let $\boldsymbol{W}_0^*, \boldsymbol{Y}_0^*, \boldsymbol{X}_0^*$, and $\boldsymbol{\Theta}_0$ be additional one sequence of the stimulus-code indicators, the stimulus-type indicators, the matrix-wise EEG signals, and the parameter set, respectively, associated with the new participant. To simplify notation, we assume that the new participant spells the same target character ω. Based on the property described in Section 4.2.1, let $\boldsymbol{y}_\omega^*$ be the possible values of stimulus-type indicators given $\boldsymbol{W}_0^*$ and the target character ω.

$$\begin{aligned}
&\Pr(\boldsymbol{Y}_0^* = \boldsymbol{y}_\omega^* \mid \boldsymbol{X}_0^*, \boldsymbol{W}_0^*, \boldsymbol{X}, \boldsymbol{W}, \boldsymbol{Y}) \\
&= \int \Pr(\boldsymbol{Y}_0^* = \boldsymbol{y}_\omega^* \mid \boldsymbol{\Theta}_0; \boldsymbol{X}_0^*, \boldsymbol{W}_0^*) \Pr(\boldsymbol{\Theta}_0 \mid \boldsymbol{X}, \boldsymbol{W}, \boldsymbol{Y}) d\boldsymbol{\Theta}_0, \\
&\Pr(\boldsymbol{Y}_0^* = \boldsymbol{y}_\omega^* \mid \boldsymbol{\Theta}_0; \boldsymbol{X}_0^*, \boldsymbol{W}_0^*) \propto \sum_{k=0}^{K-1} \pi_k \underbrace{\Pr(\boldsymbol{X}_0^* \mid \boldsymbol{\Theta}_0; \boldsymbol{y}_\omega^*)}_{\text{Equation (4.6)}} \cdot \Pr(\omega \text{ is target}),
\end{aligned} \tag{4.10}$$

where $\Pr(\omega \text{ is target}) = 1/36$ is the predictive prior on each candidate character with non-informative priors. When we repeat multiple sequences, we modify the above formula by multiplying the sequence-specific posterior conditional likelihood within each cluster.

4.4 Simulation Studies

4.4.1 Single-Channel Scenario

Setup We consider the scenario with $N = 6$ and $K = 3$. We design three pre-specified single-channel true ERP functions, the shape and magnitude of which are based on one participant from the UM-DBI Database. The simulated EEG signal segments are generated with a response window of 25 time points ($T_0 = 25$). For cluster 0, we create a typical P300 pattern where the target ERP function reaches its positive peak around the 10th time point post stimulus; for cluster 1, we consider a delayed P300 pattern where the target ERP function reaches its peak around the 17th time point post stimulus; for cluster 2, we create a typical N200 pattern where the target ERP function reaches its negative peak around the 10th time point post stimulus. We follow the equations (4.1) and (4.2) that the new participant's data are generated using the ERP functions from cluster 0 and the three clusters are evenly distributed among the remaining six source participants. We consider an autoregressive temporal structure of order 1 (i.e., AR(1)) for the background noises, where the true parameters for three clusters are $(\rho_0 = 0.6, \sigma_0 = 3.0), (\rho_1 = 0.6, \sigma_1 = 4.0)$, and $(\rho_2 = 0.6, \sigma_2 = 2.0)$. We design two cases for the single-channel scenario, denoted as SC1 and SC2. For SC1, we assume that there are matched data among source participants, and that the cluster labels for source participants 1-6 are 0, 0, 1, 1, 2, and 2, respectively. For SC2, we assume that there are no matched data among source participants, and that the cluster labels for source participants 1-6 are 1, 1, 1, 2, 2, and 2, respectively. We perform 100 replications for each case. Within each replication, we assume that each participant is spelling the same character “T” with ten sequence replications for training, and we generate additional EEG data from the new participant with 19 characters, ten sequence replications per character for testing.

Settings and Diagnostics All simulated datasets are fitted with equation (4.6). To simplify, we consider two covariance kernels κ_1 and κ_0 for target and non-target ERP functions, respectively. Both are characterized by γ-exponential kernels. The length-scale and gamma hyper-parameters of $\boldsymbol{\kappa}_1$ and κ_0 are $(0.3, 1.5)$ and $(0.4, 1.5)$, respectively. We run the MCMC with three chains, with each chain containing 3,000 burn-ins and 300 MCMC samples. The Gelman-Rubin statistics are smaller than 1.1, indicating an approximate convergence for each model fit.

Criteria We evaluate our method on clustering and prediction. For clustering, we report the average binary classification rate each source participant matches the new participant across 100 replications and produce the ERP function estimates and 90% credible bands with respect to the training sequence size. For each replication, the value of selection indicator is determined by the mode among its MCMC samples. For prediction, we report the character-level prediction accuracy of the testing data, using Bayesian mixture model with selected source participants’ data (BASIC: Mixture), swLDA with selected source participants’ data (BASIC: swLDA), Bayesian generative method with the new participant’s data only (Reference: Mixture), and swLDA with new participant’s data only (Reference: swLDA). For the purpose of notation consistency, we use Reference: Mixture to denote the Bayesian generative method although no hierarchical structure is involved. For BASIC: swLDA, the inclusion criterion is based on the selection indicators from the BASIC: Mixture method, then we refit the augmented training data with swLDA to obtain the prediction accuracy on the testing set for BASIC: swLDA.

Clustering Results The upper and lower panels of Table 4.1 show the average binary classification rate that six source participants matches the new participant for single-channel scenarios SC1 (with matched data) and SC2 (without matched data), respectively, with respect to the training sequence size of the new participant. We choose a threshold of 0.5. For SC1, our clustering method successfully identifies source participants 1-2 and excludes source participants 3-6; for SC2, our clustering method successfully excludes all source participants. The upper and lower panel of Figure 4.1 show the 95% credible bands of target ERP responses of three clusters for SC1 and SC2, respectively, with five training sequence replications of the new participant. The red curves are the true ERP response functions. We do not show the non-target ERP responses because they are not used for matching. For each cluster, the credible bands cover the entire true curve.

Table 4.1: Average percentages of selection indicator $\{Z_n\}$ for single-channel scenarios SC1 (with matched data, upper panel) and SC2 (without matched data, lower panel). For SC1, among six source participants, only participants 1 and 2 were generated from the same cluster as the new participant. For SC2, among six source participants, none were generated from the same cluster as the new participant. The numerical values are stratified with respect to the training sequence replication of the new participant. We chose a threshold of 0.5, and our method successfully performed the task of participant selection for two single-channel cases.

	Sequence Size									
Participant ID	1	2	3	4	5	6	7	8	9	10
1	0.87	0.95	0.93	0.93	0.91	0.89	0.95	0.97	0.94	0.96
2	0.89	0.96	0.93	0.93	0.90	0.87	0.95	0.97	0.95	0.96
3	0.08	0.10	0.12	0.07	0.06	0.10	0.08	0.07	0.07	0.12
4	0.08	0.10	0.12	0.07	0.06	0.10	0.08	0.07	0.07	0.12
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

	Sequence Size									
Participant ID	1	2	3	4	5	6	7	8	9	10
1	0.11	0.08	0.08	0.04	0.08	0.16	0.1	0.11	0.15	0.10
2	0.13	0.04	0.10	0.0	0.10	0.10	0.13	0.15	0.14	0.12
3	0.12	0.10	0.10	0.02	0.10	0.13	0.09	0.14	0.18	0.13
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

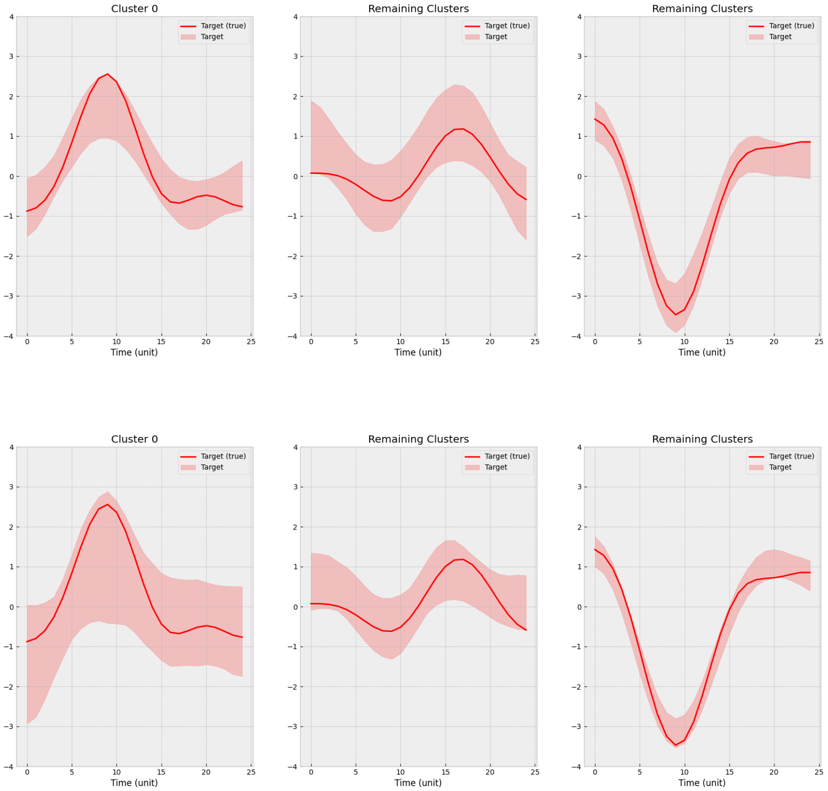


Figure 4.1: Cluster-specific target ERP function estimates with their 95% credible bands for SC1 (upper panel) and SC2 (lower panel), using five training sequence replications of the new participant. The credible bands cover the entire true curves.

Prediction Results Figures 4.2 and 4.3 show the means and standard errors of testing prediction accuracy for SC1 and SC2, respectively, by BASIC: Mixture, BASIC: swLDA, Reference: Mixture, and Reference: swLDA. The results are further stratified by up to the first four training sequence replications of the new participant. Within each row, we plot the prediction accuracy of two BASIC methods on the left side and prediction accuracy of two reference methods on the right side. For SC1, two BASIC methods show better results than their corresponding reference methods, suggesting that the selected data among source participants contribute to the prediction accuracy. For SC2, within the same row, two BASIC methods have similar values to their corresponding reference methods, suggesting that our method is robust to some extent even though no source data are selected. The prediction accuracy of two reference methods also show common increasing trend, and the method of Reference: Mixture performs better. Overall, the prediction accuracy is getting better when more training data of the new participant are included, which is within our expectation.

To quantify the data borrowing quality, we look at the number of testing sequence to reach 90% prediction accuracy for four methods. The upper and lower panels of Table 4.2 show the number of testing sequence to reach 90% accuracy with up to the first four training sequences of the new participant for SC1 and SC2, respectively. For SC1, the average testing sequences required for BASIC methods are smaller than their corresponding reference methods. For SC2, the average testing sequences required for BASIC methods are slightly larger than but close to their corresponding reference methods. It provides numerical evidence that our data borrowing framework is robust to some extent even though no participants should be selected among the source participants' pool.

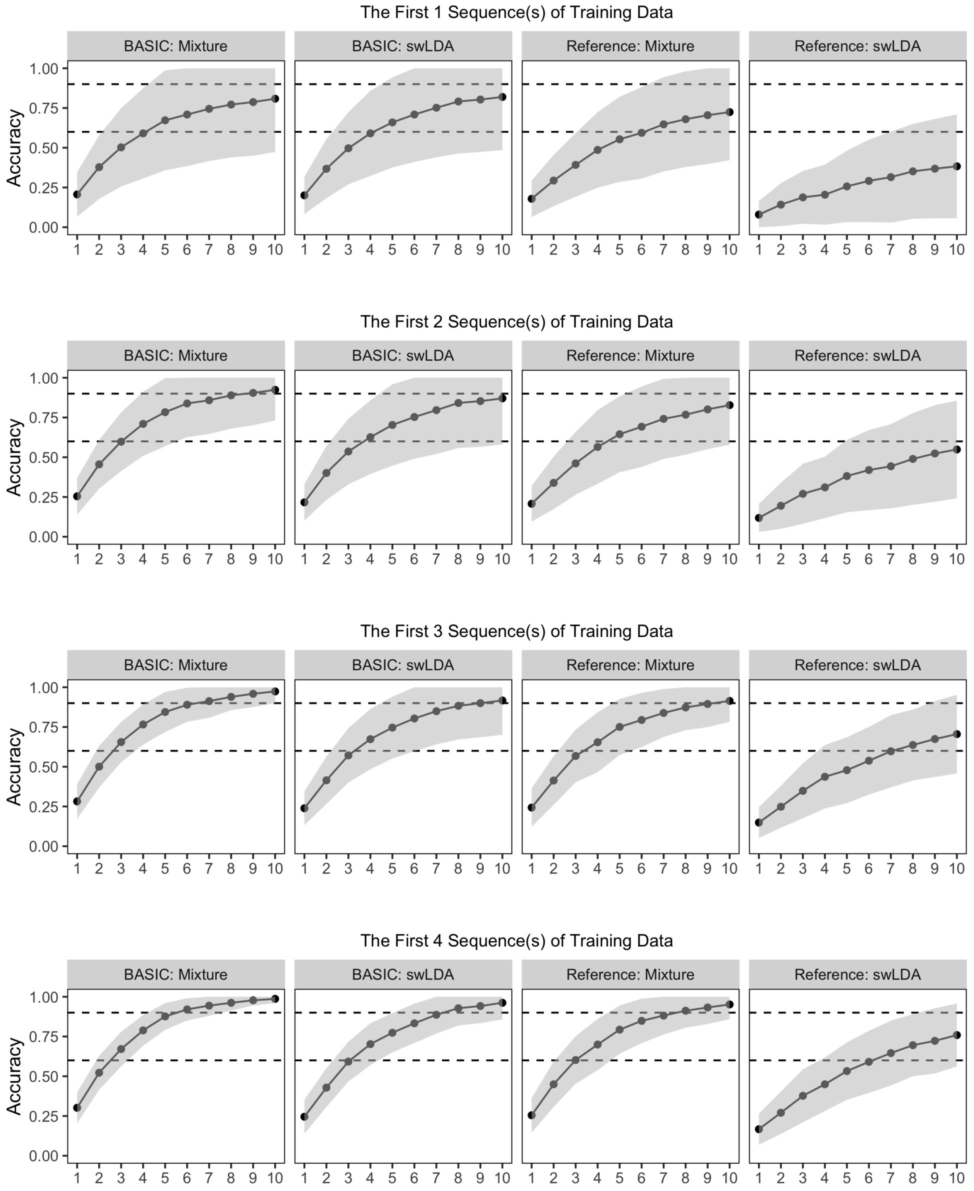


Figure 4.2: The testing prediction accuracy of SC1 by BASIC: Mixture, BASIC: swLDA, Reference: Mixture, and Reference: swLDA. The results are further stratified by up to the first four training sequence replications of the new participant. Within each row, two BASIC methods show better results than their corresponding reference methods, suggesting that the successful selection of source participants from our method contribute to the prediction accuracy. The prediction accuracy of two reference methods show common increasing trend, and the method of Reference: Mixture performs better. Overall, the prediction accuracy is better when more training data of the new participant are included.

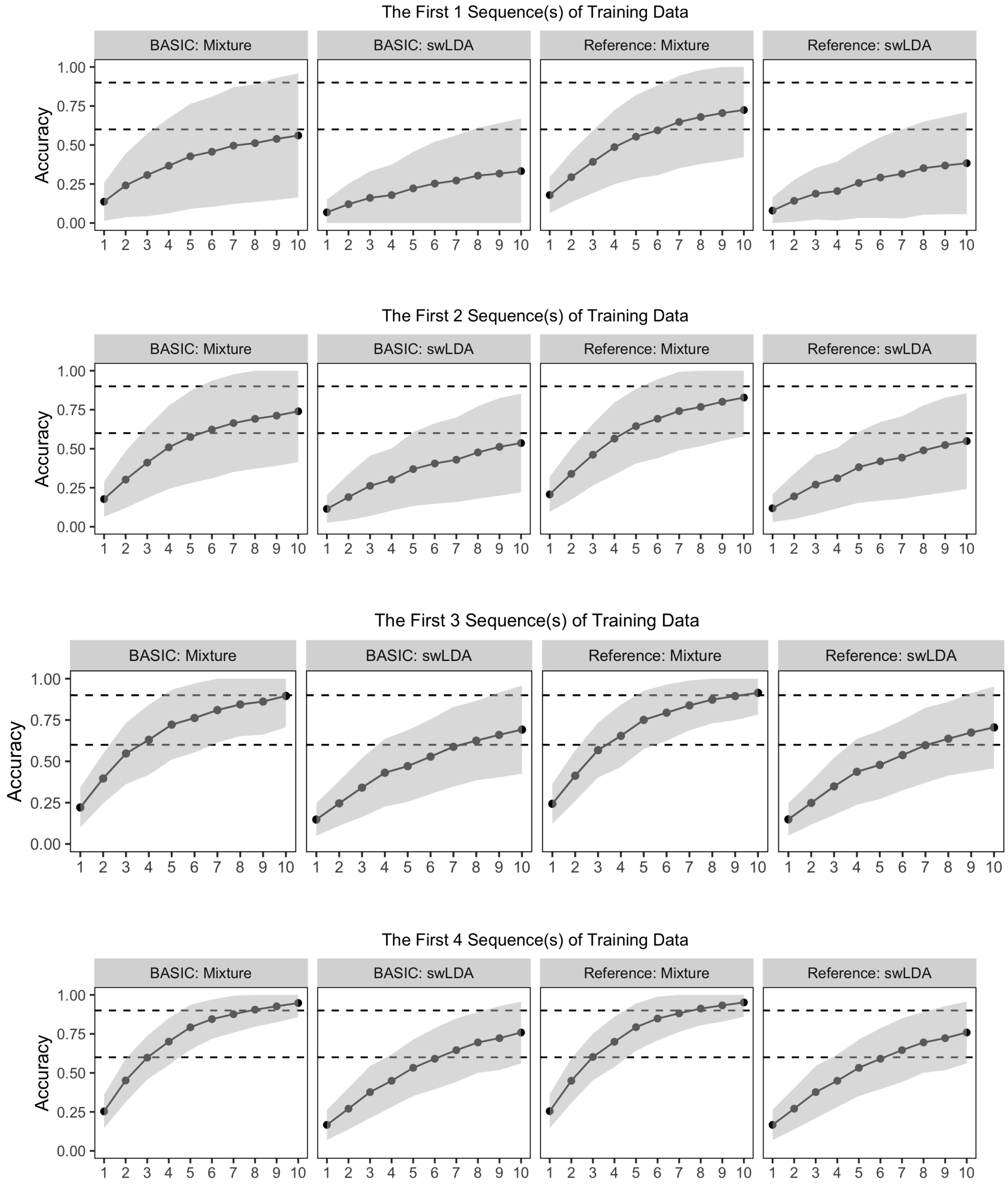


Figure 4.3: The testing prediction accuracy of SC2 by BASIC: Mixture, BASIC: swLDA, Reference: Mixture, and Reference: swLDA. The results are further stratified by up to the first four training sequence replications of the new participant. Two reference methods have similar values to their corresponding reference methods, indicating that our method is robust to some extent even though no source data are selected.

Table 4.2: The upper and lower panels show the means and standard errors of testing sequence replications to reach 90% prediction accuracy for SC1 and SC2, respectively, by BASIC: Mixture, BASIC: swLDA, Reference: Mixture, and Reference: swLDA. The upper bound for the testing sequence replication is 10. The results are also stratified by up to the first four training sequence replications of the new participant.

Training Sequence Size	90% Accuracy			
	BASIC: Mixture	BASIC: swLDA	Reference: Mixture	Reference: swLDA
1	7.83, (2.01)	8.21, (1.60)	9.16, (1.49)	9.89, (0.49)
2	7.08, (1.94)	8.05, (1.53)	8.74, (1.62)	9.86, (0.64)
3	6.70, (1.83)	7.84, (1.57)	7.89, (1.89)	9.86, (0.49)
4	6.35, (1.57)	7.73, (1.47)	7.47, (1.83)	9.65, (0.86)
Training Sequence Size	90% Accuracy			
	BASIC: Mixture	BASIC: swLDA	Reference: Mixture	Reference: swLDA
1	9.22, (1.49)	9.9, (0.48)	9.16, (1.49)	9.89, (0.49)
2	8.91, (1.56)	9.86, (0.64)	8.74, (1.62)	9.86, (0.64)
3	8.12, (1.87)	9.86, (0.49)	7.89, (1.89)	9.86, (0.49)
4	7.57, (1.92)	9.65, (0.86)	7.47, (1.83)	9.65, (0.85)

4.4.2 Multi-Channel Scenario

Setup We next consider the multi-channel scenario with $N = 6$ and $K = 3$. We design three groups of two-dimensional pre-specified true ERP functions, the shape and magnitude of which are based on three participants from the UM-DBI Database. The simulated EEG signal segments are generated with a response window of 25 time points per channel ($T_0 = 25$). Within each cluster, the shapes between two channels are similar, but the magnitude of the first channel is larger than that of the second channel. For cluster 0, both channels have separation effects between target and non-target ERP response functions (Channels 1 and 2 have positive and negative target ERP response functions, respectively; for cluster 1, only the first channel has the separation effect; for cluster 2, neither of two channels have the separation effect. We consider an autoregressive temporal structure of order 1 (i.e., AR(1)), compound symmetry spatial structure, and channel-specific variances for background noises. We determine those parameters from three participants' real data, where the true parameters for clusters 0, 1, and 2 are ($\rho_0 = 0.6, \eta_0 = 0.2, \sigma_{0,1} = 4.0, \sigma_{0,2} = 3.0$), ($\rho_1 = 0.6, \eta_1 = 0.4, \sigma_{1,1} = 3.0, \sigma_{1,2} = 2.0$), and ($\rho_2 = 0.4, \eta_2 = 0.4, \sigma_{2,1} = 2.0, \sigma_{2,2} = 1.0$), respectively. We design two cases for this scenario, denoted as MC1 and MC2. The cluster labels of source participants 1-6 for MC1 and MC2 are the same as those for SC1 and SC2, respectively. We perform 50 replications for each case. Within each replication, we assume that each participant is spelling the characters "TT," with ten sequence replications per character for training; we generate additional testing data of the same size as those in Section 4.4.1.

Table 4.3: The upper and lower panels show the average percentage of selection indicator $\{Z_n\}$ for multi-channel scenarios MC1 (with matched data) and MC2 (without matched data), respectively. We choose a threshold of 0.50. Our method successfully includes source participants 1 and 2 for MC1 but fails to exclude source participants 1-3 for MC2. However, we argue that the prediction accuracy still benefits from the partial mis-identification results.

	Sequence Size									
Participant ID	1	2	3	4	5	6	7	8	9	10
1	0.72	0.95	0.77	0.95	0.93	0.93	0.90	0.92	0.98	0.93
2	0.71	0.94	0.77	0.94	0.94	0.94	0.89	0.92	0.98	0.93
3	0.74	0.94	0.77	0.95	0.93	0.94	0.96	0.92	0.98	0.93
4	0.74	0.95	0.77	0.95	0.94	0.94	0.94	0.92	0.97	0.92
5	0.14	0.03	0.23	0.07	0.01	0.01	0.04	0.06	0.02	0.10
6	0.15	0.03	0.22	0.07	0.02	0.01	0.03	0.06	0.02	0.10

	Sequence Size									
Participant ID	1	2	3	4	5	6	7	8	9	10
1	0.95	0.96	0.97	0.97	0.96	0.97	0.97	0.97	0.97	0.97
2	0.95	0.96	0.97	0.97	0.96	0.97	0.97	0.97	0.97	0.97
3	0.95	0.96	0.97	0.97	0.96	0.97	0.97	0.97	0.97	0.97
4	0.02	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
5	0.02	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
6	0.02	0.01	0.01	0.01	0.01	0.01	0.00	0.01	0.01	0.01

Settings and Diagnostics All simulated datasets are fitted with equation (4.6). We also apply two covariance kernels κ_1 and κ_0 for target and non-target ERP functions, respectively. The two kernels are characterized by γ-exponential kernels with length-scale and gamma hyper-parameters as (0.2, 1.2) and (0.3, 1.2), respectively. We run the MCMC with three chains, with each chain containing 3,000 burn-ins and

300 MCMC samples. The Gelman-Rubin statistics are smaller than 1.1, indicating an approximate convergence for each model fit.

Criteria We apply the same evaluation criteria as in Section 4.4.1: clustering and prediction.

Clustering Results The upper and lower panels of Table 4.3 show the average binary classification rate that six source participants match the new participant for multi-channel scenarios MC1 (with matched data) and MC2 (without matched data), respectively, and the results are stratified by the training sequence sizes of the new participant. We choose a threshold of 0.5. For MC1, our clustering method successfully identify participants 1-2. For MC2, however, our clustering method does not exclude source participants 1-3. Nevertheless, we argue that the mis-identification of source participants do not affect the prediction accuracy to some extent.

Prediction Results Figures 4.4 and 4.5 show testing prediction accuracy of MC1 and MC2, respectively, by BASIC: Mixture, BASIC: swLDA, Reference: Mixture, and Reference: swLDA. The results are further stratified by up to the first four training sequence replications of the new participant. Within each row, we plot the prediction accuracy of two BASIC methods on the left side and prediction accuracy of two reference methods on the right side. For MC1, two BASIC methods show better results than their corresponding reference methods although the improvements between BASIC: swLDA and Reference: swLDA are larger. For MC2, we observe a slightly better prediction accuracy between BASIC: Mixture and Reference: Mixture but a large improvement between BASIC: swLDA and Reference: swLDA. The plots of prediction accuracy of two reference methods show common increasing trends, while Reference: Bayes performs better. Overall, the prediction accuracy is better when more training data of the new participant are included.

Similarly, we look at the number of testing sequence to each 90% accuracy for four methods. The upper and lower panels of Table 4.4 show the number of testing

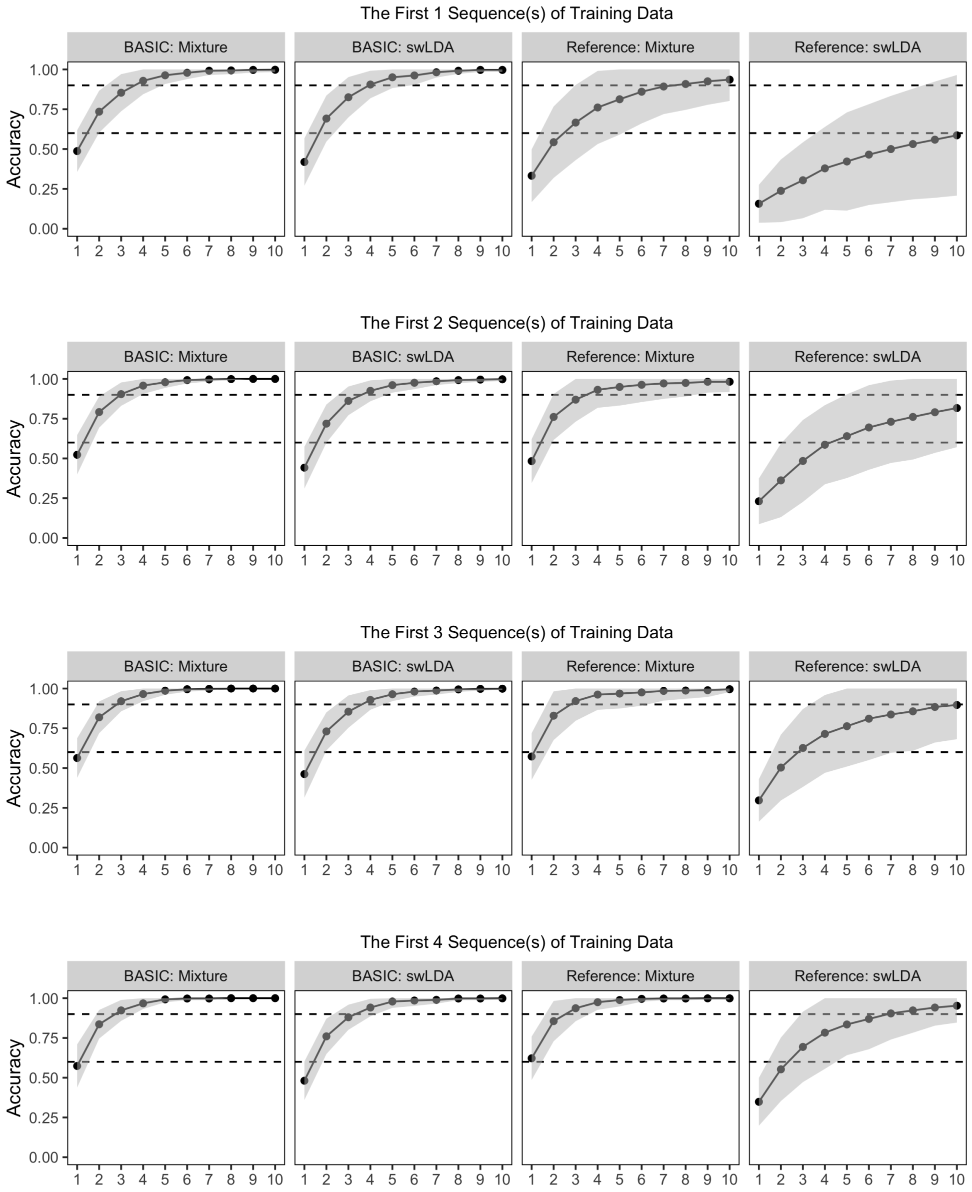


Figure 4.4: The testing prediction accuracy of MC1 by BASIC: Mixture, BASIC: swLDA, Reference: Mixture, and Reference: swLDA. Within each row, two BASIC methods show better results than their corresponding reference methods despite a larger improvement between BASIC: swLDA and Reference: swLDA. The plots indicate that the selection of source participants contribute to the prediction accuracy.

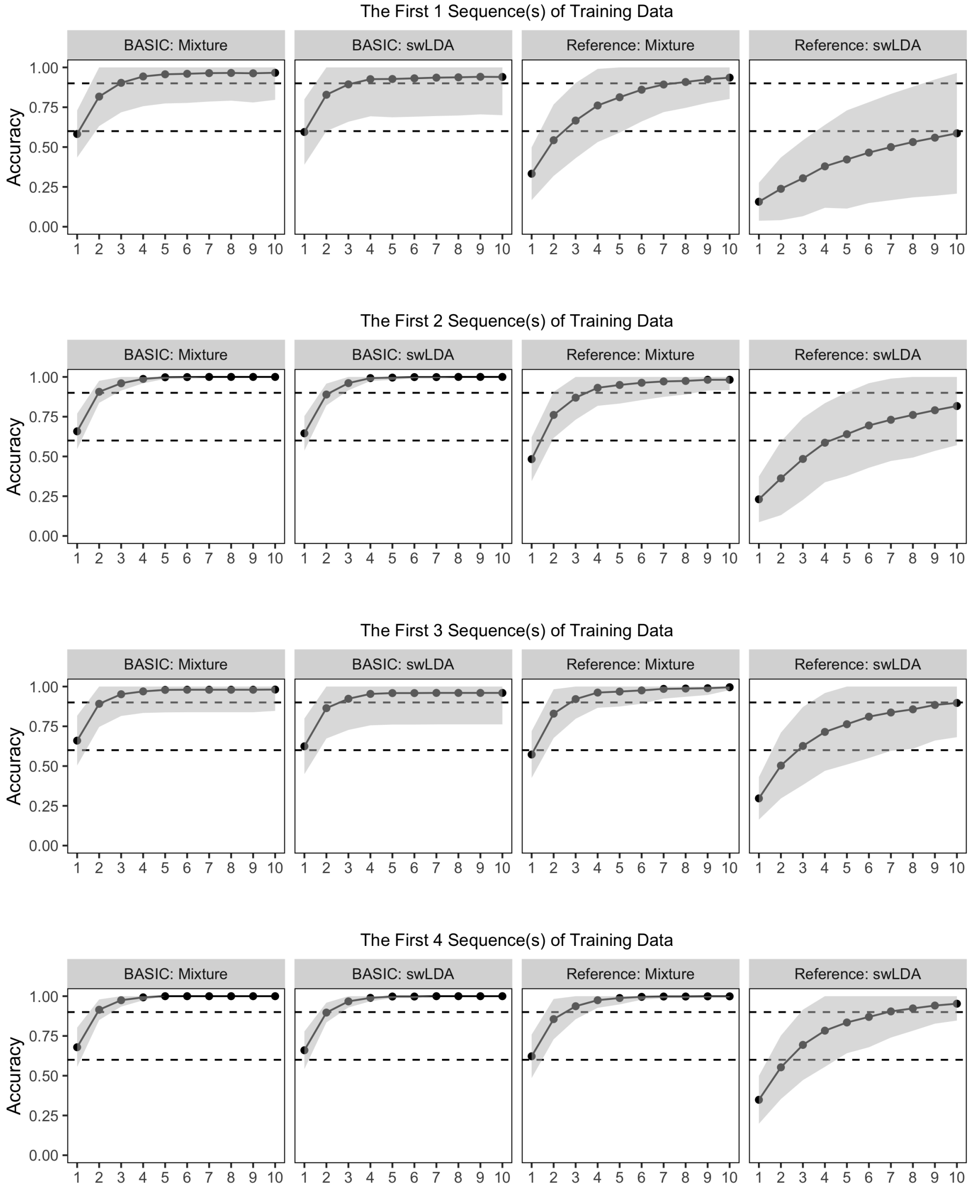


Figure 4.5: The testing prediction accuracy of MC2 by BASIC: Mixture, BASIC: swLDA, Reference: Mixture, and Reference: swLDA. We observe that BASIC: Mixture performs slightly better than Reference: Mixture, but BASIC: swLDA performs much better than Reference: swLDA. The plots show that the prediction accuracy is not affected very much even if we have partial mis-identification of participant selection.

Table 4.4: The upper and lower panels show the means and standard errors of testing sequence replications to reach 90% prediction accuracy of MC1 and MC2, respectively, by BASIC: Mixture, BASIC: swLDA, Reference: Mixture, and Reference: swLDA. The upper bound for the testing sequence replication is 10. The results are also stratified by up to the first four training sequence replications of the new participant.

Training Sequence Size	90% Accuracy			
	BASIC: Mixture	BASIC: swLDA	Reference: Mixture	Reference: swLDA
1	3.28, (1.62)	3.36, (2.08)	6.28, (2.58)	9.36, (1.29)
2	2.72, (0.89)	2.80, (0.64)	4.18, (1.99)	8.32, (2.27)
3	2.74, (1.27)	3.04, (1.60)	3.42, (1.94)	6.82, (2.37)
4	2.50, (0.61)	2.66, (0.66)	3.10, (1.04)	6.12, (2.45)
Training Sequence Size	90% Accuracy			
	BASIC: Mixture	BASIC: swLDA	Reference: Mixture	Reference: swLDA
1	4.22, (1.40)	4.54, (1.42)	6.28, (2.58)	9.36, (1.29)
2	3.64, (0.99)	4.22, (1.13)	4.18, (1.99)	8.32, (2.27)
3	3.40, (0.93)	4.18, (1.04)	3.42, (1.94)	6.82, (2.37)
4	3.36, (0.78)	3.94, (1.19)	3.10, (1.04)	6.12, (2.45)

sequence required to reach 90% accuracy with up to the first four training sequences of the new participant for MC1 and MC2, respectively. For MC1, the number of testing sequence required for BASIC methods are smaller than that for their corresponding reference methods. For MC2, the average testing sequences for BASIC: Mixture are slightly larger than those for Reference: Mixture, while the average testing sequences for BASIC: swLDA are much smaller than those for Reference: swLDA. It provides numerical evidence that our method is still robust in terms of prediction accuracy even if there exists partial false positives of participant selection.

4.5 Discussion

To reduce the calibration time while maintaining similar prediction accuracy, we propose a Bayesian SemI-supervised Classification (BASIC) method to build a participant-dependent, calibration-less framework. BASIC reduces the calibration time of a new participant by borrowing calibration data on the level of participants.

BASIC specifies the joint distribution of stimulus-specific EEG signals from all participants via a Bayesian hierarchical mixture model, and we use a generative approach as the base classifier. Our method has advantages from both inference and prediction perspectives. First, we specify the baseline cluster as the one that matches the new participant, so our method is a semi-supervised learning approach on the level of participants such that the selection indicators indicate the resemblance between source calibration data and new data on the level of participants. Second, we make predictions with the baseline cluster directly without refitting the model with the augmented data. Finally, our proposed hierarchical framework is flexible and can be extended to other base classifiers with clear and consistent parametric forms.

For single-channel scenarios SC1 (with matched data) and SC2 (without matched data), our method performs the clustering analysis correctly and achieves similar prediction accuracy. For multi-channel scenarios, our method over-selects the participants 1-3 and 3-4 for MC1 (with matched data) and MC2 (without matched data), respectively, but the prediction accuracy of the clustering methods benefits from participant selection to some extent. This counter-intuitive results could be explained as follows: Our method cannot fully distinguish cluster 0 from cluster 1 with such small sample size of the new participant because they share similar ERP function estimates of the first channel. For BASIC: Mixture, selecting source participants 1-3 messes up the ERP function estimates of the second channel, which results in the wrong calculation of log-likelihood values. Therefore, the prediction accuracy of BASIC: Mixture is not so good as that of the Reference: Mixture. However, the prediction accuracy of BASIC: swLDA is better than that of Reference: swLDA because selecting source participants 1-3 still add useful information in terms of the first channel to increase the separation effect. In addition, swLDA is a discriminant method that does not heavily rely on the ERP function estimation. In this case, it is more likely to penalize the feature selection of the second channel, and the resulting swLDA weights lead to better prediction accuracy compared to Reference: swLDA.

Although participants themselves form a natural but coarse clustering criterion (*Khazem et al.*, 2021), the qualities of the collected EEG signals are also susceptible to other factors, such as stimulus features, mental statuses, and calibration time. Since there is no feature selection procedure in our current base classifier, selecting on the level of participants may add unnecessary noises to the new participant's data even if selected source participants are overall similar to the new participant. A plausible solution to the problem would further introduce an event-specific selection indicator, where the secondary-level indicators are used to remove the "outliers" among the data pool of "matched" source participants. Thus, the posterior inferences on BASIC would produce a more powerful classifier. Or we could apply a participant-level "merge-or-keep" policy such that we would merge them together by sharing the same group index of parameters if the source participant matched the new participant; otherwise, we would keep the source-participant-specific parameter set instead.

Finally, we discuss the applicability of our BASIC framework with generative approach as the base classifier. From our extensive simulation studies, we argue that the clustering and prediction results depend on the separation effect (continuity level of target and non-target ERP functions) between target and non-target ERP functions and the background noise level. When the true target and non-target ERP functions are closer to continuous functions (with fewer jumps), it is easier to characterize them with a generative approach (i.e., Gaussian processes). In this chapter, we use the generative method as the base classifier under the BASIC framework. From equations (4.1) and (4.2), we could extend our framework to other base classifiers with specific parametric forms. The semi-supervised structure on the participant level remains unchanged, and we simply change the likelihood function with respect to the base classifier. Such extension could potentially solve the problem when EEG signals of certain participants have less continuous forms or smaller separation effects.

For example, we could use Riemannian geometry as the base classifier, and we could specify the likelihood by introducing the Riemannian Gaussian distributions (*Zanini et al.*, 2016), (*Said et al.*, 2017). Compared to our approach that incorporates the spatial dependency into the channel-specific ERP function estimates, the Riemannian geometry method starts with the sample covariance matrix and adds the referencing ERP signals to compensate for the loss of temporal information. Nevertheless, the two directions of the model specification indicate that a combination of first- and second-order statistics is necessary to form the base classifier for the P300 ERP-based BCI application.

CHAPTER V

Conclusion

5.1 Summary

In this dissertation, three challenges of current P300 ERP-BCI speller systems are addressed. We aim to improve the spelling efficiency and to create a personalized user experience for those participants with severe neuromuscular diseases from the three different aspects. Conclusions drawn from this dissertation work are summarized below.

Statistical Inferences on Brain Activity Existing works primarily focused on constructing binary classifiers to improve prediction accuracy. Very few works looked at statistical inferences under this specific P300-ERP based BCI design, especially when the data were collected with multiple sequences and overlapping components between adjacent EEG signal segments could not be disregarded. This chapter provides evidence to evaluate BCI use from the perspective of statistical inferences. We apply a new Bayesian generative framework to model the conditional distribution of multi-sequence EEG signals from real participants and explore the mechanism of brain activity in response to external stimuli by directly considering overlapping ERPs between adjacent stimuli without signal concatenation and segmentation. We develop a new GP-based prior to identify the spatial-temporally activated intervals with the split-and-merge GP (SMGP) prior and propose an information criterion for channel ranking and confirm it with existing literature. The top selected channels, including Cz, Pz, PO7, PO8, and Oz, demonstrate the spatial distributions of P300 ERP responses. In particular, the finding that channels PO8, PO7, and Oz appear the most frequently supports the finding that the performance of a P300 speller is associated with eye gaze (*Brunner et al.*, 2010). Finally, the participant-specific ERP response function estimates and channel selection help establish user-specific profiles for efficient brain-computer communications. Thus, we can incorporate user-specific channel selection into designing the EEG cap to increase spelling efficiency.

Adaptive Stimulus Selection In addition to training an efficient binary classifier, the way a sequence of stimulus groups present also affects the efficiency of real-time BCI implementations. Most existing works applied a static stimulus presentation paradigm by looping through all the events within a single sequence. Only a few studies considered history data as the criterion to dynamically determine the displayed stimulus events in the future, but they based their dynamic stimulus selection algorithms on the level of stimuli, which required intensive computations and numerical approximations due to the very short time interval between adjacent stimulus events. In this chapter, we propose a sequence-based adaptive stimulus selection algorithm based on Thompson Sampling, and we frame the problem in a multi-bandit problem with multiple selections (MAP-MS). The algorithm selects a random subset of stimuli with a fixed size during each sequence, aiming to identify all target stimulus groups and to improve spelling speed by reducing the number of unnecessary non-target stimulus groups. The algorithm is easy and straightforward to implement. We modify the rewards from raw classifier scores via the Bayes' rule and initialize the character-level probability vector with the word-level language model (LM) prior. We perform extensive simulation studies to compare our algorithm to the conventional CB paradigm. We also show the robustness of our algorithm by considering the physiological and practical constraints, such as response delay and the double flash issue, in real-time BCI implementations.

Participant-less Calibration Finally, to reduce the calibration time for a new participant, we propose a Bayesian SemI-supervised Classification (BASIC) method to build a participant-dependent, calibration-less framework. BASIC reduces the calibration time of a new participant by borrowing calibration data on the level of participants. BASIC specifies the joint distribution of stimulus-specific EEG signals from all participants via a Bayesian hierarchical mixture model, and we apply a generative approach as the base classifier. First, we specify the baseline cluster as the one that matches the new participant. Therefore, our method is a semi-supervised learning approach on the level of participants such that the selection indicators indicate how close source calibration data are to the new data on the level of participants. Second, we test with the baseline cluster directly without refitting the model with the augmented data. Finally, our proposed hierarchical framework is flexible and can be be extended to other base classifiers with clear parametric forms.

5.2 Future Work

Based on the research in this dissertation, future work of each aspect is shown as follows:

Statistical Inferences on Brain Activity First, we could modify the stimulus presentation paradigm from the current RCP design to the checkerboard design (*Townsend et al.*, 2010). The checkerboard design avoids the refractory effect (*Martens et al.*, 2009) in the RCP design, where participants might miss or fail to produce the second regular P300 ERP response when two target stimuli are too close. In addition, we could measure the participant-specific brain connectivity under the no-control (NoC) condition to specify the spatial covariance matrix prior to the individual analysis without estimating them during the calibration procedure. Similarly, we could apply a more flexible spatial correlation structure for estimation. Instead of using a simple compound-symmetry structure, a multi-block compound symmetry structure would be preferred to estimate within-block, intra-block correlation parameters, and the scalar parameter σ^2. Finally, we could develop a framework for a multi-participant analysis to incorporate time-invariant demographic covariates such as age and neuromuscular disease history by modifying the priors.

Adaptive Stimulus Selection First, since there were no real data available to evaluate the online stimulus selection algorithm, we only reported numerical results based on simulated datasets to compare different configurations of our proposed algorithm to the conventional CB paradigm. An ideal future work would be to test our algorithm on data from real participants. However, this might indicate that we would need to develop new software to determine the stimulus presentation paradigm and perform the conventional binary classification simultaneously. In addition, we would perform a theoretical analysis of the optimal regret bound for the proposed method. We would establish the results based on the existing proof (*Komiyama et al.*, 2015). In particular, we would first show that the conditional expectation of the regret given the classification scores had an optimal upper bound. Then, we would show that the double expectation also had an optimal bound under certain regularity conditions.

Participant-less Calibration First, although *Khazem et al.* argued that participants themselves formed a natural but coarse clustering criterion, the quality of the collected EEG signals are susceptible to other factors, such as calibration time. Since no feature selection procedure is applied in our current base classifier, participant-level selection may add unnecessary noise to the new participant's data even if selected source participants are overall similar to the new participant. A plausible solution would be to introduce event-specific selection indicators, which are used to remove "outliers" among the data pool of "matched" source participants. Thus, posterior inferences on BASIC produce a more powerful classifier. Alternatively, we could apply a "merge-or-keep" policy such that if the source participant matches the new participant, we would merge them together by sharing the same group of parameters; otherwise, we would keep the source-participant-specific parameter set. Second, we would apply other base classifiers that are robust to noise, such as the Riemannian geometry classifier. In that case, we could specify the likelihood by introducing Riemannian Gaussian distributions (*Zanini et al.*, 2016), (*Said et al.*, 2017). Finally, since the current data generative mechanism for simulation studies ignores the overlapping component between adjacent ERP responses, we would extend our framework to the situation where the EEG signal segments are extracted from the continuous EEG measurement allowing for the overlapping issue.

APPENDICES

APPENDIX A

Supplement for Chapter II

This supplementary document includes details of the MCMC algorithm in Section A.1, additional results of the simulation study in Section A.2, results of the sensitivity analysis of Participant A in Section A.3, and additional results of Participants B, E, and J in Section A.4.

A.1 Details of the MCMC Algorithm

We present the MCMC algorithm to update the following parameters $\boldsymbol{\alpha}, \boldsymbol{\zeta}, \sigma_s^2, \rho_s$, and ρ_t conditional on the stimulus type indicators $\boldsymbol{Y}_{l,i}$ and matrix-wise EEG signals $\boldsymbol{X}_{l,i}$.

A.1.1 Update $\boldsymbol{\alpha}$

The distribution of $\boldsymbol{\alpha}$ given $\boldsymbol{Y}_{l,i}, \boldsymbol{X}_{l,i}$ and all other parameters follows the multivariate Gaussian distribution. Let $\mathcal{N}, G_{l,i}$ and $\Sigma_{\boldsymbol{\alpha}}$ be the notation of the normal distribution, the linear map associated with $\boldsymbol{Y}_{l,i}$, and the channel-specific prior covariance. If we write the prior that $vec(\boldsymbol{\alpha}) \sim \mathcal{N}(vec(\mathbf{0}), I_E \otimes \Sigma_{\boldsymbol{\alpha}})$, then

$$
\begin{aligned}
& vec(\boldsymbol{\alpha}) \sim N\left\{\Lambda_{\boldsymbol{\alpha}}^{-1}\boldsymbol{\eta}_{\boldsymbol{\alpha}}, \Lambda_{\boldsymbol{\alpha}}^{-1}\right\}, \\
& \qquad \Lambda_{\boldsymbol{\alpha}} = \sum_{l,i} Diag\left(G_{l,i}S(\boldsymbol{\zeta}_e)\right)^T \left(C_s \otimes C_t\right)^{-1} Diag\left(G_{l,i}S(\boldsymbol{\zeta}_e)\right) + (I_E \otimes \Sigma_\alpha)^{-1}, \\
& \qquad \boldsymbol{\eta}_{\boldsymbol{\alpha}} = \sum_{l,i} Diag\left(G_{l,i}S(\boldsymbol{\zeta}_e)\right)^T \left(C_s \otimes C_t\right)^{-1} vec(\boldsymbol{X}_{l,i}).
\end{aligned}
\tag{A.1}
$$

A.1.2 Update $\boldsymbol{\zeta}$

The linear map $A(\boldsymbol{\alpha}_e)$ is

$$
\begin{aligned}
& \boldsymbol{\beta}_e = A(\boldsymbol{\alpha}_e)(\boldsymbol{\zeta}_e) = A_{1,e}\boldsymbol{\zeta}_e + A_{2,e}, \\
& A_{1,e} = \begin{pmatrix}
\boldsymbol{\alpha}_{1,e}(1) - \boldsymbol{\alpha}_{0,e}(1) & 0 & 0 \\
0 & \ddots & 0 \\
0 & 0 & \boldsymbol{\alpha}_{1,e}(T_z) - \boldsymbol{\alpha}_{0,e}(T_z) \\
0 & 0 & 0 \\
0 & \ddots & 0 \\
0 & 0 & 0
\end{pmatrix}, \quad A_{2,e} = \begin{pmatrix}
\boldsymbol{\alpha}_{0,e}(1) \\
\vdots \\
\boldsymbol{\alpha}_{0,e}(T_z) \\
\boldsymbol{\alpha}_{0,e}(1) \\
\vdots \\
\boldsymbol{\alpha}_{0,e}(T_z))
\end{pmatrix},
\end{aligned}
\tag{A.2}
$$

where $A_{1,e}$ is a matrix of dimension $2T_z \times T_z$, and $A_{2,e}$ is a $2T_z$-dim vector. The distribution of $\boldsymbol{\zeta}$ given $\boldsymbol{Y}_{l,i}, \boldsymbol{X}_{l,i}$ and all other parameters follows the multivariate truncated Gaussian distribution. Let $\mathcal{TN}$ be the notation of the truncated normal distribution and we write the prior that $vec(\boldsymbol{\zeta}) \sim \mathcal{TN}_{[\mathbf{0},\mathbf{1}]}(vec(\mathbf{0.5}), I_E \otimes \Sigma_{\boldsymbol{\zeta}})$, then

$$
\begin{aligned}
& vec(\boldsymbol{\zeta}) \sim \mathcal{TN}_{[\mathbf{0},\mathbf{1}]}\{\Lambda_{\boldsymbol{\zeta}}^{-1}\boldsymbol{\eta}_{\boldsymbol{\zeta}}, \Lambda_{\boldsymbol{\zeta}}^{-1}\}, \\
& \quad \Lambda_{\boldsymbol{\zeta}} = \sum_{l,i} Diag\left(G_{l,i}\boldsymbol{A}_{1,e}\right)^T \left(C_s \otimes C_t\right)^{-1} Diag\left(G_{l,i}\boldsymbol{A}_{1,e}\right) + (I_E \otimes \Sigma_{\boldsymbol{\zeta}})^{-1}, \\
& \quad \boldsymbol{\eta}_{\boldsymbol{\zeta}} = \sum_{l,i} Diag\left(G_{l,i}\boldsymbol{A}_{1,e}\right)^T \left(C_s \otimes C_t\right)^{-1} \left(vec(\boldsymbol{X}_{l,i}) - vec(G_{l,i}\boldsymbol{A}_2)\right) + vec(\mathbf{0.5}).
\end{aligned}
\tag{A.3}
$$

A.1.3 Update C_s

We decompose $C_s = \sigma_s^2 \tilde{C}_s$, where σ_s^2 is the scalar parameter and $\tilde{C}_s$ is the correlation matrix assuming compound symmetry structure with the parameter ρ_s. The distribution of σ_s^2 given $\boldsymbol{Y}_{l,i}, \boldsymbol{X}_{l,i}$ and other parameters follows the inverse gamma distribution, while we use the adaptive rejection sampling method to sample ρ_s. Let Γ^{-1}, L, I, T, E be the notation of the inverse gamma distribution, the number of characters of interest, the number of sequences per character, the length of signal segments, and channel dimension, respectively. We write the prior that $\sigma_s^2 \sim \Gamma^{-1}(\alpha_s, \beta_s)$, $\nu_s > E-1, \Psi_s$ is the positive definite matrix, $\sigma_s^2 \sim \Gamma^{-1}(a_s, b_s)$, where $\alpha_s, \beta_s > 0$, then

(A.4)

$$\begin{aligned}
&\sigma_X^2 \sim \Gamma^{-1}\left\{a + \frac{LT}{2}, b + \frac{1}{2}\sum_{l}\left(\boldsymbol{X}_l - G_l\boldsymbol{\beta}\right)^T C^{-1}(\rho)\left(\boldsymbol{X}_l - G_l\boldsymbol{\beta}\right)\right\},\\
&C_s \sim Wishart^{-1}\left(\nu_s + LIT, \Psi_s + \sum_{l,i}\left(\boldsymbol{X}_{l,i} - \boldsymbol{M}_{l,i}\right)^T C_t^{-1}(\boldsymbol{X}_{l,i} - \boldsymbol{M}_{l,i})\right).\\
&\sigma_s^2 \sim \Gamma^{-1}\left(\alpha_s + \frac{LITE}{2}, \beta_s + \frac{1}{2}\mathrm{tr}\left(\tilde{C}_s^{-1}\sum_{l,i}\left(\boldsymbol{X}_{l,i} - \boldsymbol{M}_{l,i}\right)^T C_t^{-1}\left(\boldsymbol{X}_{l,i} - \boldsymbol{M}_{l,i}\right)\right)\right).
\end{aligned}$$

A.1.4 Update ρ_t

We assume ρ_t generates the temporal correlation matrix of the $AR(2)$ structure and follows the discrete uniform distribution such that the correlation matrix associated with ρ_t is invertible. For the $AR(2)$ structure, the restriction of ρ_t is equivalent to $||\rho_t||_1 < 1$, where $||\cdot||_1$ is the L_1-norm. Therefore, given $\boldsymbol{X}_{l,i}, \boldsymbol{Y}_{l,i}$ and other parameters, we select the optimal ρ_t by maximizing the conditional log-likelihood such that

$$\rho_{t,opt} = \underset{\boldsymbol{\rho}_t}{\operatorname{argmax}} \sum_{l,i} \log \mathcal{L}(\boldsymbol{X}_{l,i} \mid \boldsymbol{\alpha}, \boldsymbol{\zeta}, C_s, \rho_t; \boldsymbol{Y}_{l,i}). \tag{A.5}$$

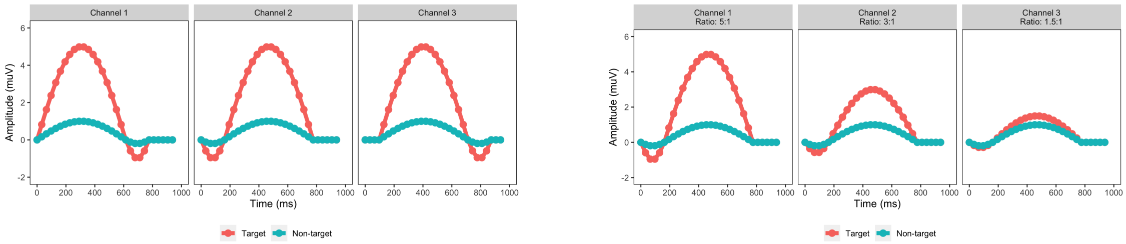


Figure A.1: **Left Panel**: A figure of 3-dimensional mean response functions with different temporal separation effects. **Right Panel**: A figure of 3-dimensional mean response functions with channel-specific SNR values. The maximum amplitude ratios between target and non-target stimuli are 5 : 1, 3 : 1, and 1.5 : 1, respectively.

A.2 Additional Simulations and Results

We provide additional results of the simulation study. For single-channel simulation studies, Tables A.1 and A.2 summarize the cumulative prediction accuracy over the number of sequence replications under five scenarios described in Section 5.2 comparing the SMGP method with $\sigma_x^2 = 10, \rho_t = (0.5, 0)$ (Table A.1) and $\sigma_x^2 = 20, \rho_t = (0.5, 0)$ (Table A.2) to other ML methods. Both point estimates and standard errors over 100 datasets are calculated. Our method has the highest and most precise prediction accuracy under five scenarios, which indicates that our analysis is robust to moderate model mis-specifications. For multi-channel simulation studies, Figure A.1 shows two groups of multi-dimensional mean response functions. The first group of mean response functions focuses on detecting temporal activation regions, while the second group focuses on channel ranking. Figure A.2 shows the estimated ERP functions of target and non-target stimuli for the true generative scenario with parameters $\sigma_x^2 = 20$ and $\sigma_x^2 = 40$, respectively. Table A.3 summarizes the ISWR, IMWR of the SMGP method, the ESWR, EEWR of the swLDA method and prediction accuracy over the number of sequences with $\sigma_x^2 = 40, \rho_t = (0.5, 0)$,and $\rho_s = 0.5$ comparing the SMGP method to other ML methods. Both point estimates and standard errors over 100 datasets are provided.

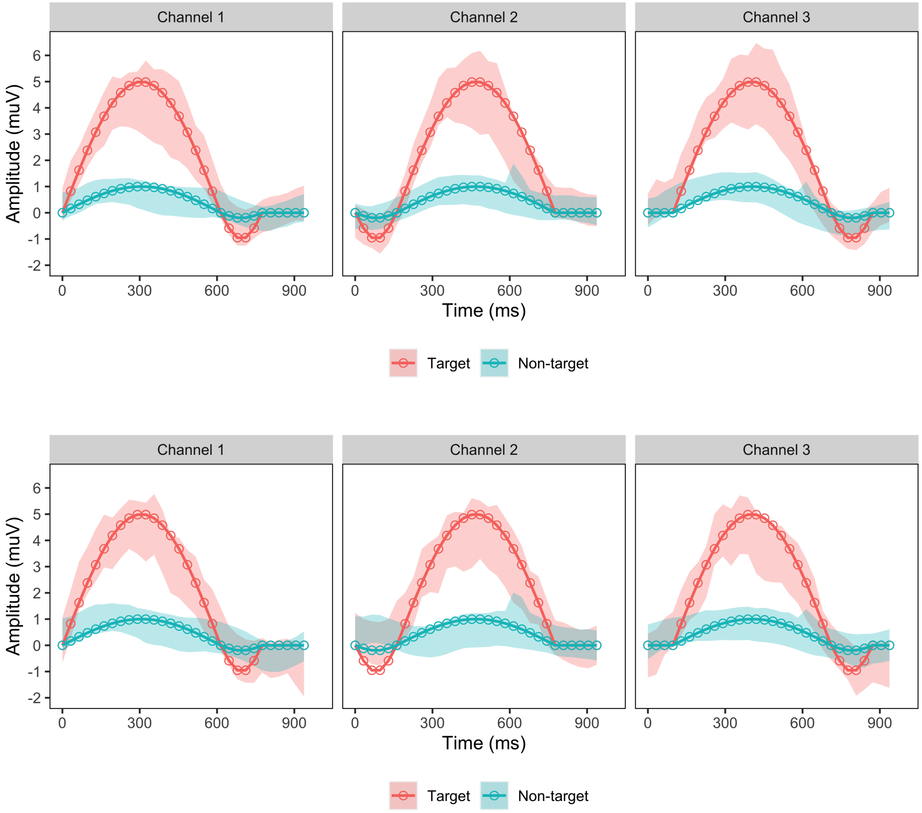


Figure A.2: The upper and lower rows show the 95% credible bands of simulated ERP functions to target and non-target stimuli for three channels for the true generative scenario with $\zeta_0 = 0.5, \sigma_x^2 = 20$ and $\zeta_0 = 0.5, \sigma_x^2 = 40$, respectively. The dots and curves are the true underlying values. The credible bands cover the entire the true curves for three channels.

Table A.1: Cumulative prediction accuracy for the single-channel simulation study under five scenarios with $\sigma_x^2 = 10, \rho_t = (0.5, 0)$ comparing the SMGP method to other ML methods. The split threshold of SMGP method is $\zeta_0 = 0.5$. At most 30% of feature vector is selected for swLDA. Point estimates and standard errors averaged over 100 datasets are reported. The SMGP results are marked bold.

		Testing Sequence Replications		
Scenarios	**Methods**	3	4	5
True Generative	**SMGP**	**0.84 (0.09)**	**0.93 (0.06)**	**0.97 (0.04)**
	Neural Network	0.65 (0.10)	0.77 (0.10)	0.84 (0.09)
	SVM	0.67 (0.11)	0.76 (0.10)	0.82 (0.09)
	Logistic	0.71 (0.11)	0.83 (0.08)	0.88 (0.08)
	Random Forest	0.73 (0.11)	0.83 (0.09)	0.9 (0.07)
	swLDA	0.77 (0.11)	0.87 (0.08)	0.92 (0.07)
	XGBoost	0.65 (0.13)	0.75 (0.11)	0.82 (0.09)
Mis-specified Noise	**SMGP**	**0.63 (0.12)**	**0.76 (0.10)**	**0.84 (0.08)**
	Neural Network	0.41 (0.13)	0.50 (0.13)	0.6 (0.14)
	SVM	0.45 (0.11)	0.55 (0.12)	0.62 (0.12)
	Logistic	0.53 (0.12)	0.64 (0.12)	0.72 (0.11)
	Random Forest	0.56 (0.13)	0.68 (0.12)	0.76 (0.10)
	swLDA	0.59 (0.12)	0.70 (0.11)	0.77 (0.10)
	XGBoost	0.49 (0.13)	0.60 (0.12)	0.68 (0.11)
Shorter Window	**SMGP**	**0.79 (0.10)**	**0.89 (0.07)**	**0.95 (0.05)**
	Neural Network	0.65 (0.10)	0.76 (0.10)	0.84 (0.10)
	SVM	0.66 (0.12)	0.76 (0.11)	0.84 (0.08)
	Logistic	0.73 (0.11)	0.83 (0.09)	0.89 (0.07)
	Random Forest	0.73 (0.11)	0.82 (0.09)	0.90 (0.07)
	swLDA	0.79 (0.10)	0.88 (0.08)	0.94 (0.05)
	XGBoost	0.64 (0.11)	0.76 (0.11)	0.83 (0.08)
Longer Window	**SMGP**	**0.82 (0.08)**	**0.90 (0.07)**	**0.98 (0.04)**
	Neural Network	0.61 (0.11)	0.72 (0.10)	0.80 (0.09)
	SVM	0.60 (0.12)	0.71 (0.10)	0.79 (0.10)
	Logistic	0.68 (0.11)	0.80 (0.10)	0.88 (0.08)
	Random Forest	0.70 (0.12)	0.81 (0.09)	0.87 (0.08)
	swLDA	0.75 (0.11)	0.85 (0.09)	0.92 (0.07)
	XGBoost	0.61 (0.11)	0.72 (0.10)	0.80 (0.11)
Mis-specified Signal	**SMGP**	**0.72 (0.1)**	**0.84 (0.08)**	**0.91 (0.07)**
	Neural Network	0.56 (0.11)	0.66 (0.12)	0.75 (0.11)
	SVM	0.59 (0.11)	0.69 (0.11)	0.76 (0.09)
	Logistic	0.64 (0.12)	0.77 (0.09)	0.82 (0.09)
	Random Forest	0.63 (0.13)	0.75 (0.09)	0.83 (0.08)
	swLDA	0.66 (0.12)	0.78 (0.10)	0.85 (0.09)
	XGBoost	0.57 (0.12)	0.68 (0.11)	0.75 (0.10)

Table A.2: Cumulative prediction accuracy for the single-channel simulation study under five scenarios with $\sigma_x^2 = 20, \rho_t = (0.5, 0)$ comparing the SMGP method to other ML methods. The split threshold of the SMGP method is $\zeta_0 = 0.5$. Point estimates and standard errors averaged over 100 datasets are reported. The SMGP method results are marked in bold.

		Testing Sequence Replications		
Scenarios	**Methods**	3	4	5
True Generative	**SMGP**	**0.54 (0.11)**	**0.66 (0.11)**	**0.76 (0.09)**
	Neural Network	0.36 (0.10)	0.45 (0.11)	0.52 (0.12)
	SVM	0.35 (0.10)	0.44 (0.11)	0.53 (0.13)
	Logistic	0.45 (0.11)	0.56 (0.11)	0.65 (0.11)
	Random Forest	0.44 (0.11)	0.54 (0.11)	0.64 (0.10)
	swLDA	0.49 (0.10)	0.60 (0.11)	0.69 (0.10)
	XGBoost	0.36 (0.11)	0.46 (0.11)	0.55 (0.13)
Mis-specified Noise	**SMGP**	**0.37 (0.12)**	**0.47 (0.12)**	**0.55 (0.11)**
	Neural Network	0.19 (0.10)	0.23 (0.12)	0.27 (0.14)
	SVM	0.24 (0.09)	0.29 (0.09)	0.34 (0.10)
	Logistic	0.33 (0.11)	0.42 (0.12)	0.47 (0.12)
	Random Forest	0.32 (0.11)	0.42 (0.11)	0.48 (0.12)
	swLDA	0.34 (0.13)	0.44 (0.13)	0.50 (0.10)
	XGBoost	0.27 (0.11)	0.35 (0.12)	0.41 (0.11)
Shorter Response Window Length	**SMGP**	**0.52 (0.13)**	**0.64 (0.12)**	**0.73 (0.10)**
	Neural Network	0.34 (0.13)	0.44 (0.12)	0.52 (0.13)
	SVM	0.36 (0.12)	0.45 (0.12)	0.52 (0.10)
	Logistic	0.46 (0.12)	0.58 (0.12)	0.67 (0.11)
	Random Forest	0.44 (0.11)	0.56 (0.11)	0.65 (0.12)
	swLDA	0.49 (0.12)	0.62 (0.11)	0.73 (0.11)
	XGBoost	0.39 (0.12)	0.47 (0.12)	0.55 (0.14)
Longer Response Window Length	**SMGP**	**0.56 (0.12)**	**0.68 (0.1)**	**0.8 (0.09)**
	Neural Network	0.35 (0.12)	0.44 (0.12)	0.52 (0.11)
	SVM	0.35 (0.12)	0.42 (0.12)	0.49 (0.10)
	Logistic	0.45 (0.12)	0.57 (0.12)	0.65 (0.11)
	Random Forest	0.43 (0.11)	0.55 (0.12)	0.64 (0.10)
	swLDA	0.51 (0.11)	0.61 (0.11)	0.72 (0.10)
	XGBoost	0.37 (0.10)	0.46 (0.11)	0.53 (0.11)
Mis-specified Signal	**SMGP**	**0.44 (0.11)**	**0.54 (0.11)**	**0.65 (0.11)**
	Neural Network	0.28 (0.10)	0.35 (0.10)	0.41 (0.12)
	SVM	0.30 (0.09)	0.39 (0.11)	0.45 (0.13)
	Logistic	0.39 (0.11)	0.49 (0.12)	0.56 (0.10)
	Random Forest	0.36 (0.10)	0.46 (0.11)	0.53 (0.12)
	swLDA	0.41 (0.11)	0.50 (0.11)	0.59 (0.11)
	XGBoost	0.31 (0.10)	0.39 (0.11)	0.46 (0.12)

Table A.3: **Upper Panel**: Cumulative prediction accuracy for the multi-channel simulation study under the true generative mechanism with $\sigma_x^2 = 40, \rho_t = (0.5, 0)$, and $\rho_s = 0.5$ comparing the SMGP method to other ML methods. The split threshold of the SMGP method is $\zeta_0 = 0.5$. Point estimates and standard errors averaged over 100 datasets are reported. Results of the SMGP method are marked in bold. Overall, the SMGP method has the highest and most precise prediction accuracy. **Lower Panel**: The ISWR and the IMWR of the SMGP method and the ESWR and the EEWR of the swLDA method for the multi-channel simulation study under the true generative mechanism with $\sigma_x^2 = 40, \rho_t = (0.5, 0)$, and $\rho_s = 0.5$. Channel-specific point estimates and standard errors averaged over 100 datasets are reported.

	Testing Sequence Replications		
Methods	3	4	5
SMGP	**0.67 (0.11)**	**0.79 (0.09)**	**0.87 (0.08)**
Neural Network	0.55 (0.12)	0.66 (0.11)	0.75 (0.10)
SVM	0.54 (0.11)	0.64 (0.12)	0.73 (0.10)
Logistic	0.53 (0.11)	0.62 (0.11)	0.71 (0.11)
Random Forest	0.50 (0.12)	0.62 (0.12)	0.69 (0.11)
swLDA	0.59 (0.11)	0.71 (0.11)	0.80 (0.10)
XGBoost	0.45 (0.12)	0.56 (0.11)	0.65 (0.11)

	SMGP		**swLDA**	
Channels	**ISWR**	**IMWR**	**ESWR**	**EEWR**
1	0.98 (0.04)	0.6 (0.13)	0.32 (0.07)	0.76 (0.09)
2	0.97 (0.04)	0.57 (0.14)	0.29 (0.07)	0.76 (0.09)
3	0.97 (0.04)	0.62 (0.14)	0.24 (0.07)	0.83 (0.08)

A.3 Sensitivity Analysis

We performed two sensitivity analyses to see how channel ranking, prediction accuracy, ERP function estimates, and split-and-merge time windows changed with respect to the kernel hyper-parameters and the bandpass filters.

First, we applied different hyper-parameters of the γ-exponential kernel to Participant A. We assigned 0.4, 0.5, and 0.6 to the scale parameter s_0 and $1.7, 1.8$, and 1.9 to the gamma parameter γ_0. Based on our information criterion, channels PO8, PO7, Oz, P4, and Cz were always selected under nine variations of hyper-parameters. Figures A.3 and A.4 show the P300 ERP function estimates and significant temporal intervals by varying thresholds of median split probabilities for channels Cz and PO8, respectively. Channel Cz is the most clinically relevant, while channel PO8 is selected the most frequently with the largest value. The thresholds included 0.6, 0.75, and 0.9. Overall, the combination of s_0 and γ_0 did not affect the ERP function estimates significantly. For channel Cz, we observed the split window with the threshold of 0.90 when s_0 and γ_0 were in the middle of the hyper-parameter space. Table A.4 shows the prediction accuracy with channels PO7, PO8, Oz, P4, and Cz at seven sequence replications under nine variations of kernel hyper-parameters. It suggests that a combination of moderate s_0 and γ_0 can achieve 100% correct accuracy.

We also examined the channel ranking with respect to different kernel hyper-parameters. Table A.5 shows the top six selected channels with respect to nine variations of kernel hyper-parameters. Overall, the results did not change much. Channels PO7, PO8, Oz, P3, P4, and Cz were the top six selected channels.

Next, we applied different bandpass filter parameters to all participant data. We assigned 0.4, 0.5, and 0.6 to the lower bound of the bandpass filter and 5.5, 6, and 6.5 to the upper bound of the bandpass filter. Table A.6 shows the top six selected channels with respect to nine variations of the bandpass filter. Overall, the results did not change much. Channels PO7, PO8, Oz, P3, P4, and Cz were the top six

Table A.4: Prediction accuracy of Participant A using channels PO8, PO7, Oz, P4, and Cz at seven sequence replications under nine variations of kernel hyper-parameters.

$\mathbf{s_0}$ γ_0	0.4	0.5	0.6
1.7	1.00	1.00	1.00
1.8	1.00	1.00	0.95
1.9	1.00	1.00	0.95

selected channels among nine variations of bandpass filter parameters.

Table A.5: The top six selected channels among ten participants with nine variations of kernel hyper-parameters.

Scale s_0	**Gamma** γ_0		**Top 1**	**Top 2**	**Top 3**	**Top 4**	**Top 5**	**Top 6**
0.4	1.7	Channel	PO7	PO8	Oz	P4	Cz	P3
		Frequency	10	10	10	7	6	6
0.4	1.8	Channel	PO7	PO8	Oz	P3	P4	Cz
		Frequency	10	10	10	7	7	5
0.4	1.9	Channel	PO7	PO8	Oz	P3	P4	Cz
		Frequency	10	10	10	7	7	5
0.5	1.7	Channel	PO7	PO8	Oz	P3	P4	Cz
		Frequency	10	10	10	7	7	5
0.5	1.8	Channel	PO7	PO8	Oz	P3	P4	Cz
		Frequency	10	10	10	7	7	5
0.5	1.9	Channel	PO7	PO8	Oz	P3	P4	Cz
		Frequency	10	10	10	7	7	5
0.6	1.7	Channel	PO7	PO8	Oz	P3	P4	Cz
		Frequency	10	10	10	7	7	5
0.6	1.8	Channel	PO7	PO8	Oz	P3	P4	Cz
		Frequency	10	10	10	7	7	5
0.6	1.9	Channel	PO7	PO8	Oz	P3	P4	Cz
		Frequency	10	10	10	7	7	5

A.4 Additional Results of EEG-BCI Data Analysis

We provide additional results of Participants B, E, and J. Figures A.6-A.8 show their ERP function estimates with 95% credible bands and channel-specific significant temporal intervals by varying thresholds of median split probabilities. The results were produced based on the 16-channel joint model fitting. The thresholds of median

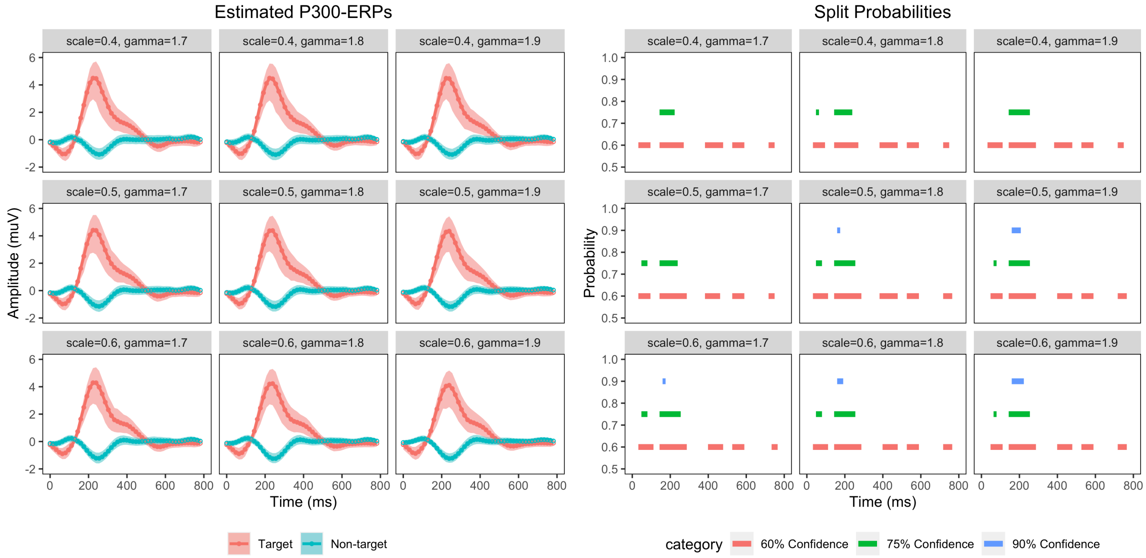


Figure A.3: **Left Panel**: A figure of ERP function estimates of target and non-target stimuli. **Right Panel**: A figure of significant temporal intervals by varying thresholds of median split probabilities 0.60, 0.75, and 0.90. All plots were produced from channel Cz, Participant A under nine variations of kernel hyper-parameters.

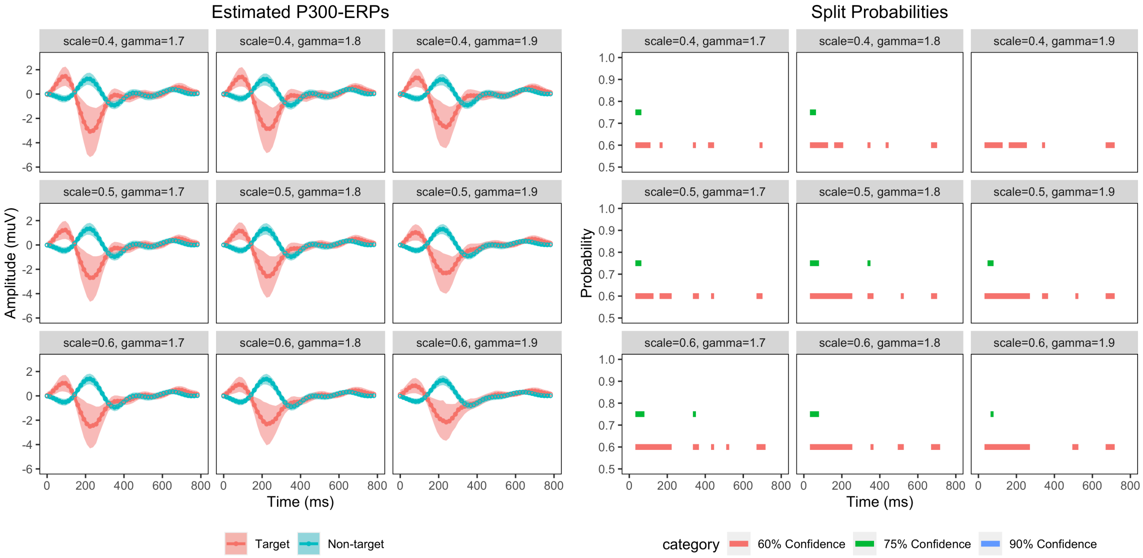


Figure A.4: **Left Panel**: A figure of ERP function estimates of target and non-target stimuli. **Right Panel**: A figure of significant temporal intervals by varying thresholds of median split probabilities 0.60, 0.75, and 0.90. All plots were produced from channel PO8, Participant A under nine variations of kernel hyper-parameters.

Table A.6: The top six selected channels among ten participants with nine variations of the bandpass filter.

Bandpass Filter		**Top 1**	**Top 2**	**Top 3**	**Top 4**	**Top 5**	**Top 6**
[0.4, 5.5]	Channel	PO7	PO8	Oz	P3	P4	Cz
	Frequency	10	10	10	6	6	5
[0.4, 6]	Channel	PO7	PO8	Oz	P3	P4	Cz
	Frequency	10	10	10	6	6	4
[0.4, 6.5]	Channel	PO7	PO8	Oz	P3	P4	Cz
	Frequency	10	10	10	7	7	5
[0.5, 5.5]	Channel	PO7	PO8	Oz	P3	P4	Cz
	Frequency	10	10	10	7	7	5
[0.5, 6]	Channel	PO7	PO8	Oz	P3	P4	Cz
	Frequency	10	10	10	7	7	4
[0.5, 6.5]	Channel	PO7	PO8	Oz	P4	P3	Cz
	Frequency	10	10	10	8	7	5
[0.6, 5.5]	Channel	PO7	PO8	Oz	P3	P4	Cz
	Frequency	10	10	10	7	7	5
[0.6, 6]	Channel	PO7	PO8	Oz	P3	P4	Cz
	Frequency	10	10	10	7	7	5
[0.6, 6.5]	Channel	PO7	PO8	Oz	P4	P3	Cz
	Frequency	10	10	10	8	7	5

split probabilities were $0.6, 0.75$, and 0.9. We attach the plot for Participant A in Figure A.5 for convenient comparison. We arranged channel-specific plots by their spatial locations to better visualize the patterns of brain activity. In particular, the upper and lower rows represent the front and back of the head. A "z" (zero) refer to a channel placed on the mid-line (sagittal) plane of the skull. Channels with even numbers $(2, 4, 6, 8)$ refer to the electrodes placed on the right side of the head, whereas channels with odd numbers $(1, 3, 5, 7)$ refer to those on the left.

The ERP function estimates among these participants shared similar patterns. First, the target ERPs of frontal and central channels (channel names with "F" and "C") of the majority of participants shared the negative drop between 100ms and 150ms and reached their first positive peaks around 300ms. The target ERP functions gradually declined to zero and collapsed with non-target ERP functions between 600 ms and 800 ms for the majority of participants, which again showed

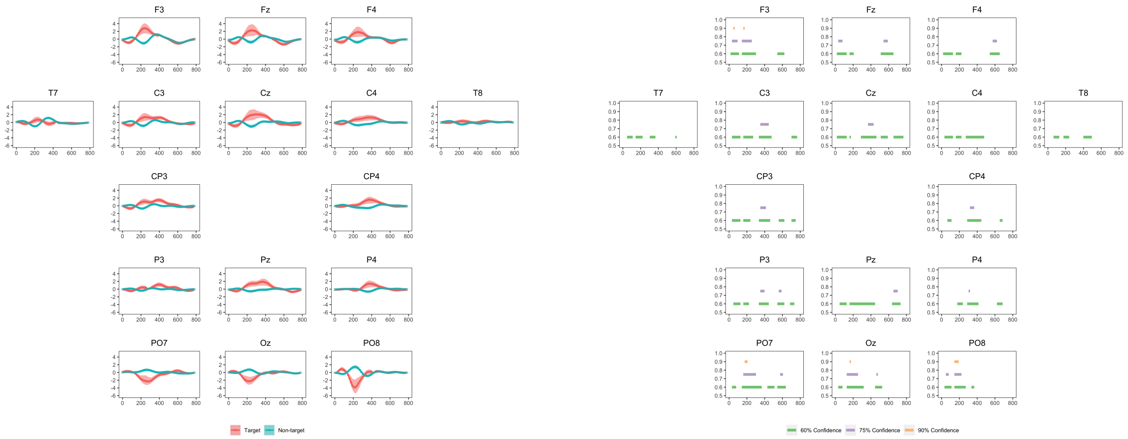


Figure A.5: A copy of Figure 3 in the main text. **Left Panel**: A figure of channel-specific ERP function estimates of target and non-target stimuli with the 95% credible bands of Participant A. **Right Panel**: A figure of channel-specific significant temporal intervals by varying thresholds of median split probabilities of Participant A.

that our split-and-merge prior worked well for these participants. In addition, the target ERP functions of the parietal-occipital, and occipital channels (channel names with "PO" and "O") also only reached their negative peaks between 200 ms and 250 ms without first reaching a positive peak.

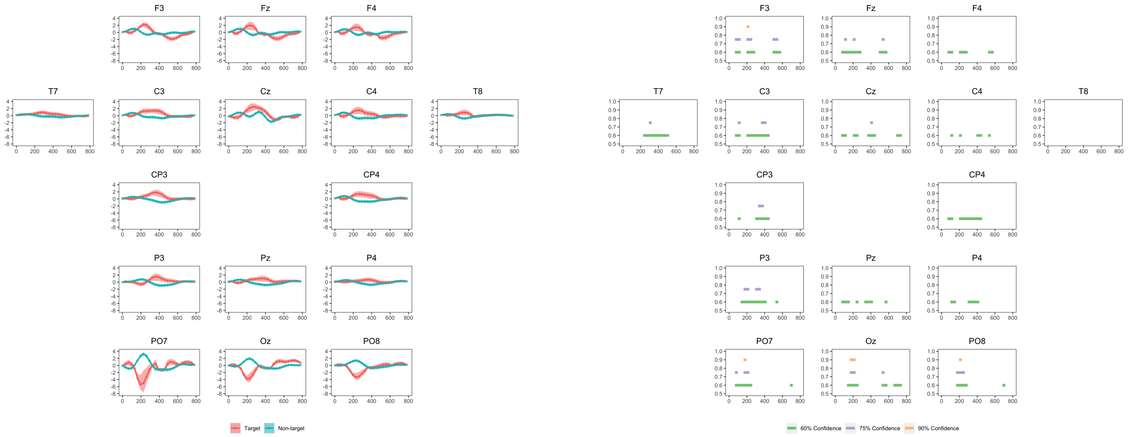


Figure A.6: **Left Panel**: A figure of channel-specific ERP function estimates of target and non-target stimuli with the 95% credible bands of Participant B. **Right Panel**: A figure of channel-specific significant temporal intervals by varying thresholds of median split probabilities of Participant B.

For cross-participant evaluation, we compared the neural activity between par-

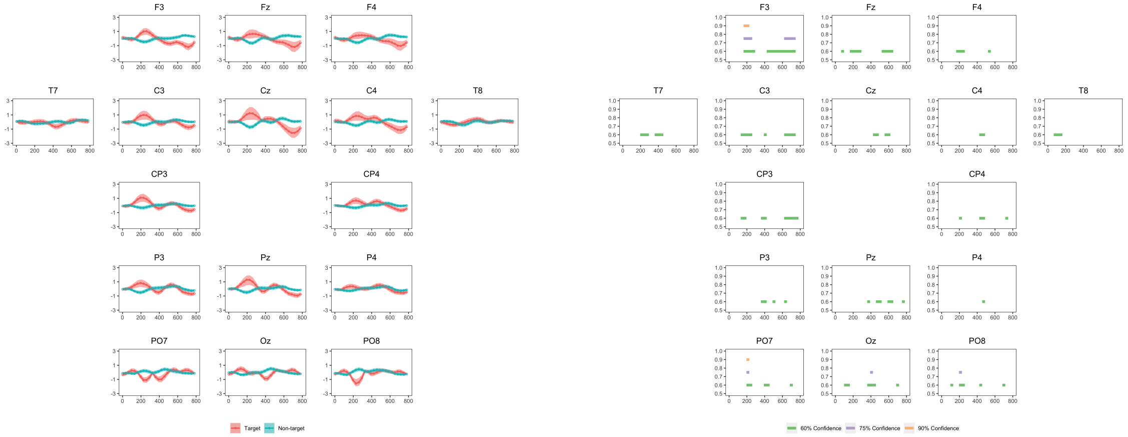


Figure A.7: **Left Panel**: A figure of channel-specific ERP function estimates of target and non-target stimuli with the 95% credible bands of Participant E. **Right Panel**: A figure of channel-specific significant temporal intervals by varying thresholds of median split probabilities of Participant E.

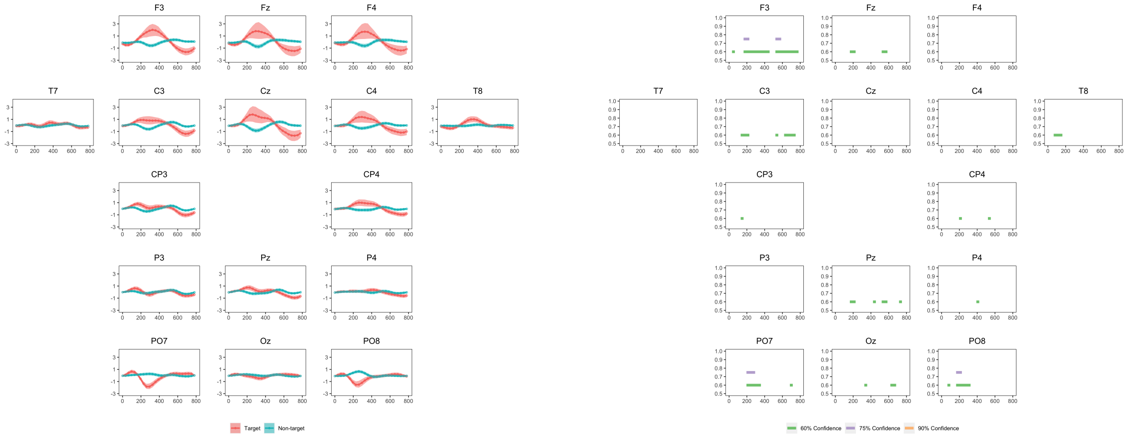


Figure A.8: **Left Panel**: A figure of channel-specific ERP function estimates of target and non-target stimuli with the 95% credible bands of Participant J. **Right Panel**: A figure of channel-specific significant temporal intervals by varying thresholds of median split probabilities of Participant J.

Table A.7: Cumulative prediction accuracy of Participant E comparing the SMGP method with $\zeta_0 = 0.4$ to other ML methods at seven sequence replications for the top six selected channels and all 16 channels. Channel ranking was based on the 16-channel joint fitting result of the SMGP method and the proposed information criterion.

Channels	**SMGP**	**CNN**	**SVM**	**Logistic**	**RF**	**swLDA**	**XGBoost**
PO8, PO7	**0.68**	0.84	0.89	0.95	0.89	0.89	0.89
PO8, PO7, Cz	**0.74**	0.79	0.84	0.95	0.89	0.95	0.89
PO8, PO7, Cz, Oz	**0.68**	0.58	1.00	1.00	0.89	1.00	0.89
PO8, PO7, Cz, Oz, P4	**0.74**	0.74	1.00	1.00	0.89	1.00	0.95
PO8, PO7, Cz, Oz, P4, Pz	**0.79**	0.84	1.00	1.00	0.95	1.00	0.95
All Channels	**0.89**	0.58	0.84	1.00	0.84	1.00	0.95

ticipants with ALS and healthy controls as well as between younger and older participants. Figure A.9 shows the ERP function estimates of channels Cz and PO8 of Participants A, B, E, and J. For both channels Cz and PO8, the peak amplitude of Participant E was smaller than the other three participants, suggesting that the signal of the P300 ERPs for Participant E at channel Cz was weak; however, the difference between target and non-target stimuli was still significant. For channel Cz, the target ERP functions merged with the non-target functions around 500 ms post-stimulus for younger participants (A and B), while the target ERP functions were significantly below the non-target functions at the end of the EEG response window for senior participants (E and J). We did not observe this pattern at channel PO8.

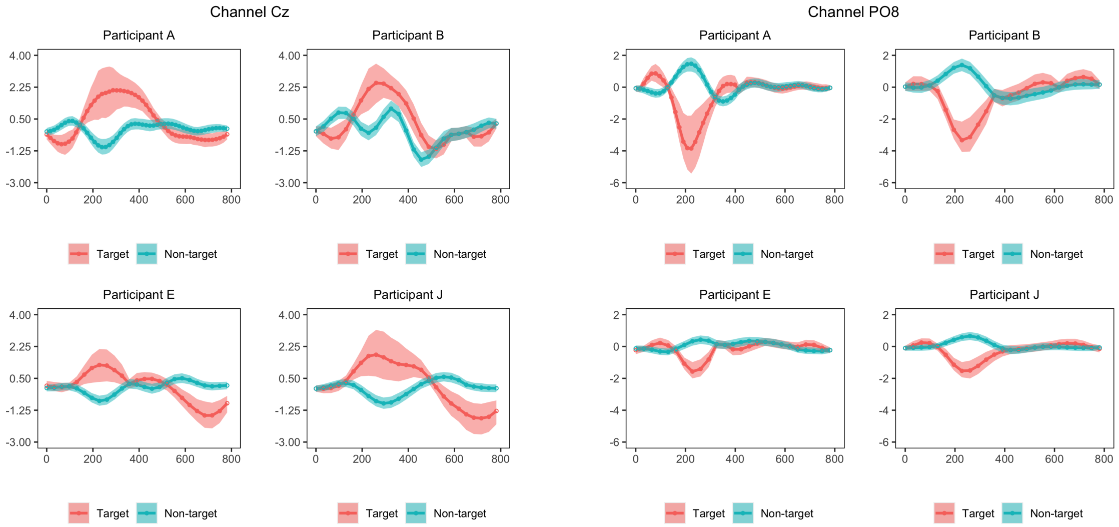


Figure A.9: The ERP function estimates of target and non-target stimuli with 95% credible bands of Participants A, B, E, and J at channel Cz (**Left Panel**) and channel PO8 (**Right Panel**). Participants A and B were young female healthy controls, while Participants E and J were senior males, of whom only E was diagnosed with ALS.

APPENDIX B

Supplement for Chapter IV

This supplementary document includes the MCMC algorithm for single-channel scenario in Section B.1 and true ERP functions in Section B.2.

B.1 MCMC Algorithm for Single-channel Scenario

B.1.1 Model Formula

Let $\boldsymbol{\Theta} := \{\boldsymbol{\Theta}_k\}_{k=0}^{K-1} = \{\boldsymbol{\beta}_{k,1}, \boldsymbol{\beta}_{k,0}, \sigma_k, \rho_k\}_{k=0}^{K-1}$. Let $\boldsymbol{\mathcal{H}} := \{\boldsymbol{\mathcal{H}}_k\}_{k=0}^{K-1}$ be the set of hyper-parameters for $\boldsymbol{\Theta}$. For $n = 1, \ldots, N$,

$$\begin{aligned} Z_n \mid \boldsymbol{\pi} &\sim \text{Discrete}(\boldsymbol{\pi}), \\ \boldsymbol{X}_{n,i,j} \mid Y_{n,i,j}; Z_n = k, \boldsymbol{\Theta} &\sim \mathcal{MVN}(\boldsymbol{M}_{n,i,j,k}, \sigma_k^2 \text{Corr}(\rho_k)), \quad i = 1, \ldots, I, \quad j = 1, \ldots, 12, \\ \boldsymbol{M}_{n,i,j,k} &= \boldsymbol{\beta}_{k,1} Y_{n,i,j} + \boldsymbol{\beta}_{k,0}(1 - Y_{n,i,j}), \\ \boldsymbol{\pi} &\sim \text{Dirichlet}\left(\frac{\alpha}{K}, \ldots, \frac{\alpha}{K}\right), \end{aligned}$$

where $\text{Corr}(\rho_k)$ is an exponential-decay correlation matrix characterized by ρ_k. For $n = 0$,

$$\boldsymbol{X}_{n,i,j} \mid Y_{n,i,j}; \boldsymbol{\Theta}_0 \sim \mathcal{MVN}(\boldsymbol{M}_{n,i,j,0}, \sigma_0^2 \text{Corr}(\rho_k)), \quad i = 1, \ldots, I, \quad j = 1, \ldots, 12.$$

B.1.2 Joint Distribution and Prior Specifications

The joint distribution is

$$
\begin{aligned}
&\Pr(\{\boldsymbol{X}_{n,i,j}\}, \boldsymbol{\Theta}, \{Z_n\}, \boldsymbol{\pi} \mid \{Y_{n,i,j}\}; \boldsymbol{\mathcal{H}}, \alpha) \\
&= \underbrace{\left(\prod_{k=0}^{K-1} \Pr(\boldsymbol{\Theta}_k; \boldsymbol{\mathcal{H}}_k)\right) \Pr(\boldsymbol{\pi}; \alpha) \cdot}_{\text{Prior Distribution}} \underbrace{\left(\prod_{n=1}^{N} \Pr(\{\boldsymbol{X}_{n,i,j}\} \mid \{Y_{n,i,j}\}, Z_n, \boldsymbol{\Theta}_{Z_n}) \Pr(Z_n \mid \boldsymbol{\pi})\right)}_{\text{Source Participants' Likelihood}} \\
&\cdot \underbrace{\Pr\left(\{\boldsymbol{X}_{0,i,j}\} \mid \{Y_{0,i,j}\}, \boldsymbol{\Theta}_0\right)}_{\text{New Participant's Likelihood}},
\end{aligned}
$$

The joint posterior distribution of parameters is proportional to

$$
\begin{aligned}
&\Pr(\boldsymbol{\Theta}, \{Z_n\}, \boldsymbol{\pi} \mid \{Y_{n,i,j}\}; \boldsymbol{\mathcal{H}}, \alpha) \\
&\propto \left(\prod_{n=1}^{N} \Pr(\{\boldsymbol{X}_{n,i,j}\} \mid \{Y_{n,i,j}\}, Z_n, \boldsymbol{\Theta}_{Z_n}) \Pr(Z_n \mid \boldsymbol{\pi})\right) \cdot \Pr(\boldsymbol{X}_{0,i,j} \mid \{Y_{0,i,j}\}, \boldsymbol{\Theta}_0) \\
&\cdot \prod_{k=0}^{K-1} \Pr(\boldsymbol{\Theta}_k; \boldsymbol{\mathcal{H}}_k) \Pr(\boldsymbol{\pi}; \alpha).
\end{aligned}
$$

For each cluster-specific parameter set $\boldsymbol{\Theta}_k,\ k = 0, \ldots, K-1$,

$$
\boldsymbol{\beta}_{k,1} \sim \mathcal{GP}(\boldsymbol{0}, \psi_{k,1}\boldsymbol{\kappa}_1), \quad \boldsymbol{\beta}_{k,0} \sim \mathcal{GP}(\boldsymbol{0}, \psi_{k,0}\boldsymbol{\kappa}_0), \quad \rho_k \sim \mathcal{U}(0,1), \quad \sigma_k \sim \mathcal{HC}(0, 5.0).
$$

By Mercer's Theorem, we further obtain that

$$
\begin{aligned}
\boldsymbol{\beta}_{k,1} = \psi_{k,1}\boldsymbol{\Psi}_1\boldsymbol{\alpha}_{k,1}, \quad \boldsymbol{\alpha}_{k,1} \sim \mathcal{MVN}(\boldsymbol{0}, \mathrm{Diag}(\boldsymbol{\lambda}_1)), \quad \psi_{k,1} \sim \mathcal{LN}(0,1), \\
\boldsymbol{\beta}_{k,0} = \psi_{k,0}\boldsymbol{\Psi}_0\boldsymbol{\alpha}_{k,0}, \quad \boldsymbol{\alpha}_{k,0} \sim \mathcal{MVN}(\boldsymbol{0}, \mathrm{Diag}(\boldsymbol{\lambda}_0)), \quad \psi_{k,0} \sim \mathcal{LN}(0,1),
\end{aligned}
$$

where $\boldsymbol{\Psi}_1, \boldsymbol{\lambda}_1$ and $\boldsymbol{\Psi}_0, \boldsymbol{\lambda}_0$ are eigen-functions and eigen-vectors for target and non-target ERP response functions, respectively.

B.1.3 Markov Chain Monte Carlo

We apply the Gibbs Sampler method to draw posterior samples. For parameters of interest, within each cluster k, we update $\boldsymbol{\alpha}_{k,1}, \boldsymbol{\alpha}_{k,0}, \psi_{k,1}, \psi_{k,0}, \sigma_k$, and ρ_k, then we update latent indicators $\{Z_n\}_{n=1}^{N}$ and weight probability vector $\boldsymbol{\pi}$. Among the set of continuous variables, only $\boldsymbol{\alpha}_{k,1}$ and $\boldsymbol{\alpha}_{k,0}$ have the closed forms for the posterior distribution. We resort to the Metropolis-Hastings algorithm to update $\psi_{k,1}, \psi_{k,0}, \sigma_k$, and ρ_k. We use "rest" to denote the remaining parameters for simplicity.

B.1.3.1 Update $\boldsymbol{\alpha}_{k,1}, \boldsymbol{\alpha}_{k,0}$

For coefficients of the γ-exponential kernel, we specify the priors that $\boldsymbol{\alpha}_{k,1} \sim \mathcal{MVN}(\mathbf{0}, \mathrm{Diag}(\boldsymbol{\lambda}_1))$ and $\boldsymbol{\alpha}_{k,0} \sim \mathcal{MVN}(\mathbf{0}, \mathrm{Diag}(\boldsymbol{\lambda}_0))$. For $k > 0$,

$$\begin{aligned}
\boldsymbol{\alpha}_{k,1} \mid \text{rest} &\sim \mathcal{MVN}(\boldsymbol{\Lambda}_{\alpha_{k,1}}^{-1}\boldsymbol{\eta}_{\alpha_{k,1}}, \boldsymbol{\Lambda}_{\alpha_{k,1}}^{-1}),\\
\boldsymbol{\Lambda}_{\alpha_{k,1}} &= \sum_{Z_n=k}\sum_{Y_{n,i,j}=1} (\psi_{k,1}\boldsymbol{\Psi}_1)^\top \left(\sigma_k^2 \mathrm{Corr}(\rho_k)\right)^{-1} (\psi_{k,1}\boldsymbol{\Psi}_1) + \mathrm{Diag}(\boldsymbol{\lambda}_k^{-1}),\\
\boldsymbol{\eta}_{\alpha_{k,1}} &= \sum_{Z_n=k}\sum_{Y_{n,i,j}=1} (\psi_{k,1}\boldsymbol{\Psi}_1)^\top \left(\sigma_k^2 \mathrm{Corr}(\rho_k)\right)^{-1} \boldsymbol{X}_{n,i,j},\\
\boldsymbol{\alpha}_{k,0} \mid \text{rest} &\sim \mathcal{MVN}(\boldsymbol{\Lambda}_{\alpha_{k,0}}^{-1}\boldsymbol{\eta}_{\alpha_{k,0}}, \boldsymbol{\Lambda}_{\alpha_{k,0}}^{-1}),\\
\boldsymbol{\Lambda}_{\alpha_{k,0}} &= \sum_{Z_n=k}\sum_{Y_{n,i,j}\neq 1} (\psi_{k,0}\boldsymbol{\Psi}_0)^\top \left(\sigma_k^2 \mathrm{Corr}(\rho_k)\right)^{-1} (\psi_{k,0}\boldsymbol{\Psi}_0) + \mathrm{Diag}(\boldsymbol{\lambda}_k^{-1}),\\
\boldsymbol{\eta}_{\alpha_{k,0}} &= \sum_{Z_n=k}\sum_{Y_{n,i,j}\neq 1} (\psi_{k,0}\boldsymbol{\Psi}_0)^\top \left(\sigma_k^2 \mathrm{Corr}(\rho_k)\right)^{-1} \boldsymbol{X}_{n,i,j},
\end{aligned}$$

For $k = 0$,

$$
\begin{aligned}
\boldsymbol{\alpha}_{k,1} \mid \text{rest} &\sim \mathcal{MVN}(\boldsymbol{\Lambda}_{\alpha_{k,1}}^{-1}\boldsymbol{\eta}_{\alpha_{k,1}}, \boldsymbol{\Lambda}_{\alpha_{k,1}}^{-1}),\\
\boldsymbol{\Lambda}_{\alpha_{k,1}} &= \sum_{\{n:Z_n=k\}\cup\{0\}} \sum_{Y_{n,i,j}=1} (\psi_{k,1}\boldsymbol{\Psi}_1)^\top \left(\sigma_k^2 \text{Corr}(\rho_k)\right)^{-1} (\psi_{k,1}\boldsymbol{\Psi}_1) + \text{Diag}(\boldsymbol{\lambda}_k^{-1}),\\
\boldsymbol{\eta}_{\alpha_{k,1}} &= \sum_{\{n:Z_n=k\}\cup\{0\}} \sum_{Y_{n,i,j}=1} (\psi_{k,1}\boldsymbol{\Psi}_1)^\top \left(\sigma_k^2 \text{Corr}(\rho_k)\right)^{-1} \boldsymbol{X}_{n,i,j},\\
\boldsymbol{\alpha}_{k,0} \mid \text{rest} &\sim \mathcal{MVN}(\boldsymbol{\Lambda}_{\alpha_{k,0}}^{-1}\boldsymbol{\eta}_{\alpha_{k,0}}, \boldsymbol{\Lambda}_{\alpha_{k,0}}^{-1}),\\
\boldsymbol{\Lambda}_{\alpha_{k,0}} &= \sum_{\{n:Z_n=k\}\cup\{0\}} \sum_{Y_{n,i,j}\neq 1} (\psi_{k,0}\boldsymbol{\Psi}_0)^\top \left(\sigma_k^2 \text{Corr}(\rho_k)\right)^{-1} (\psi_{k,0}\boldsymbol{\Psi}_0) + \text{Diag}(\boldsymbol{\lambda}_k^{-1}),\\
\boldsymbol{\eta}_{\alpha_{k,0}} &= \sum_{\{n:Z_n=k\}\cup\{0\}} \sum_{Y_{n,i,j}\neq 1} (\psi_{k,0}\boldsymbol{\Psi}_0)^\top \left(\sigma_k^2 \text{Corr}(\rho_k)\right)^{-1} \boldsymbol{X}_{n,i,j},
\end{aligned}
$$

B.1.3.2 Update $\psi_{k,1}, \psi_{k,0}$

For the kernel variance parameters, we specify the priors that $\psi_{k,1}, \psi_{k,0} \sim \mathcal{LN}(0, 1)$. Let $f_{\mathcal{LN}}(x; 0, 1)$ be the density function for the Log-Normal prior. For $k > 0$,

$$
\begin{aligned}
f(\psi_{k,1} \mid \text{rest}) &\propto \prod_{Z_n=k} \prod_{Y_{n,i,j}=1} \boldsymbol{\phi}\left(\boldsymbol{X}_{n,i,j}; \psi_{k,1}\boldsymbol{\alpha}_{k,1}, \sigma_k^2 \text{Corr}(\rho_k)\right) f_{\mathcal{LN}}(\psi_{k,1}; 0, 1),\\
f(\psi_{k,0} \mid \text{rest}) &\propto \prod_{Z_n=k} \prod_{Y_{n,i,j}\neq 1} \boldsymbol{\phi}\left(\boldsymbol{X}_{n,i,j}; \psi_{k,0}\boldsymbol{\alpha}_{k,0}, \sigma_k^2 \text{Corr}(\rho_k)\right) f_{\mathcal{LN}}(\psi_{k,1}; 0, 1),
\end{aligned}
$$

For $k = 0$,

$$
\begin{aligned}
f(\psi_{k,1} \mid \text{rest}) &\propto \prod_{\{n:Z_n=k\}\cup\{0\}} \prod_{Y_{n,i,j}=1} \boldsymbol{\phi}\left(\boldsymbol{X}_{n,i,j}; \psi_{k,1}\boldsymbol{\alpha}_{k,1}, \sigma_k^2 \text{Corr}(\rho_k)\right) f_{\mathcal{LN}}(\psi_{k,1}; 0, 1),\\
f(\psi_{k,0} \mid \text{rest}) &\propto \prod_{\{n:Z_n=k\}\cup\{0\}} \prod_{Y_{n,i,j}\neq 1} \boldsymbol{\phi}\left(\boldsymbol{X}_{n,i,j}; \psi_{k,0}\boldsymbol{\alpha}_{k,0}, \sigma_k^2 \text{Corr}(\rho_k)\right) f_{\mathcal{LN}}(\psi_{k,1}; 0, 1),
\end{aligned}
$$

B.1.3.3 Update σ_k

For the noise variance parameter, we specify the prior that $\sigma_k \sim \mathcal{HC}(0, 5)$. Let $f_{\mathcal{HC}}(x; 0, 5)$ be the density function for the Half-Cauchy prior. Let $\boldsymbol{M}_{n,i,j,k} = \psi_{k,1}\boldsymbol{\alpha}_{k,1}Y_{n,i,j}+$

$\psi_{k,0}\boldsymbol{\alpha}_{k,0}(1-Y_{n,i,j})$, then

$$f(\sigma_k \mid \text{rest}) \propto \prod_{Z_n=k}\prod_{i,j}\boldsymbol{\phi}\left(\boldsymbol{X}_{n,i,j};\boldsymbol{M}_{n,i,j,k},\sigma_k^2\text{Corr}(\rho_k)\right)\cdot f_{\mathcal{HC}}(\sigma_k;0,5),\quad k>0,$$
$$f(\sigma_k \mid \text{rest}) \propto \prod_{\{n:Z_n=k\}\cup\{0\}}\prod_{i,j}\boldsymbol{\phi}\left(\boldsymbol{X}_{n,i,j};\boldsymbol{M}_{n,i,j,k},\sigma_k^2\text{Corr}(\rho_k)\right)\cdot f_{\mathcal{HC}}(\sigma_k;0,5),\quad k=0.$$

B.1.3.4 Update ρ_k

For the temporal correlation parameter, we specify the prior that $\rho_k \sim \mathcal{U}(0,1)$. The exponential-decay correlation matrix is constructed by

$$\text{Corr}(\rho_k) = \begin{pmatrix} 1 & \rho_k & \rho_k^2 & \cdots & \rho_k^{T_0-1} \\ \rho_k & 1 & \rho_k & \cdots & \rho_k^{T_0-2} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \rho_k^{T_0-1} & \cdots & \rho_k^2 & \rho_k & 1 \end{pmatrix},$$

where T_0 is the EEG response window length. Then,

$$f(\rho_k \mid \text{rest}) \propto \prod_{Z_n=k}\prod_{i,j}\boldsymbol{\phi}\left(\boldsymbol{X}_{n,i,j};\boldsymbol{M}_{n,i,j,k},\sigma_k^2\text{Corr}(\rho_k)\right),\quad k>0,$$
$$f(\rho_k \mid \text{rest}) \propto \prod_{\{n:Z_n=k\}\cup\{0\}}\prod_{i,j}\boldsymbol{\phi}\left(\boldsymbol{X}_{n,i,j};\boldsymbol{M}_{n,i,j,k},\sigma_k^2\text{Corr}(\rho_k)\right),\quad k=0.$$

B.1.3.5 Update Z_n

For the latent indicator Z_n, $n=1,\ldots,N$, we specify the prior that $Z_n \sim \text{Discrete}(\boldsymbol{\pi})$, and we obtain that

$$\Pr(Z_n=k \mid \text{rest}) \propto \pi_k \prod_{i,j}\boldsymbol{\phi}(\boldsymbol{X}_{n,i,j};\boldsymbol{\beta}_{k,1}Y_{n,i,j}+\boldsymbol{\beta}_{k,0}(1-Y_{n,i,j}),\sigma_k^2\text{Corr}(\rho_k)).$$

B.1.3.6 Update $\boldsymbol{\pi}$

For the weight probability $\boldsymbol{\pi}$, we specify the prior that $\boldsymbol{\pi} \sim \text{Dirichlet}(\frac{\alpha}{K}, \dots, \frac{\alpha}{K})$, and we obtain that

$$\Pr(\boldsymbol{\pi} \mid \text{rest}) \sim \text{Dirichlet}\left(\left\{\frac{\alpha}{K} + m_k\right\}_{k=0}^{K-1}\right), \quad m_k = \sum_{n=1}^{N} \mathbb{1}\{Z_n = k\}.$$

B.2 True ERP Functions for Simulation Studies

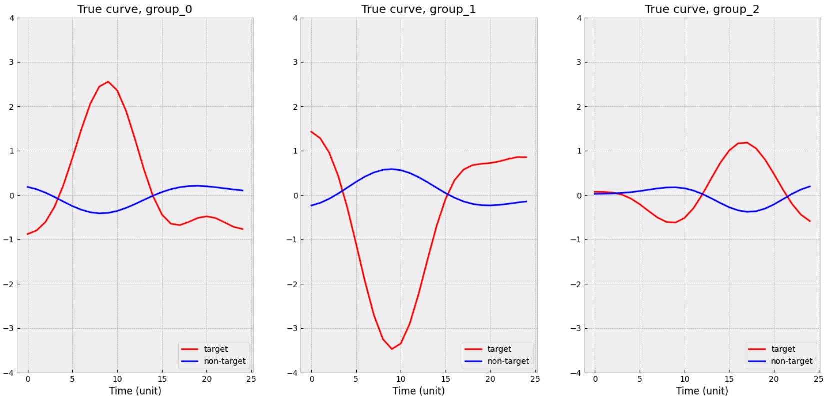


Figure B.1: Three true ERP functions for single-channel simulation studies.

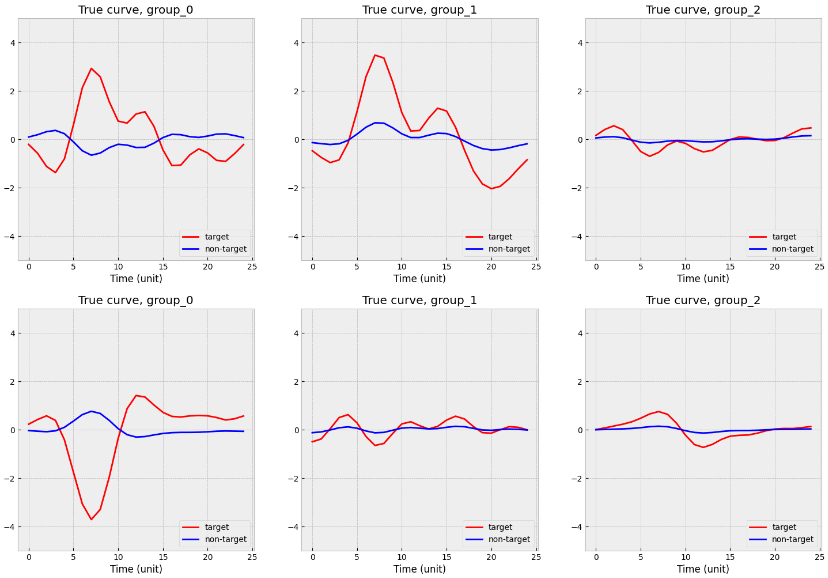


Figure B.2: Three two-dimensional true ERP functions for multi-channel simulation studies.

BIBLIOGRAPHY

BIBLIOGRAPHY

Adair, J., A. Brownlee, F. Daolio, and G. Ochoa (2017), Evolving training sets for improved transfer learning in brain computer interfaces, in *International Workshop on Machine Learning, Optimization, and Big Data*, pp. 186–197, Springer.

An, X., X. Zhou, W. Zhong, S. Liu, X. Li, and D. Ming (2020), Weighted subject-semi-independent erp-based brain-computer interface, in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 2969–2972, IEEE.

Barachant, A., and M. Congedo (2014), A plug&play p300 bci using information geometry, *arXiv preprint arXiv:1409.0107*.

Barachant, A., S. Bonnet, M. Congedo, and C. Jutten (2011), Multiclass brain–computer interface classification by riemannian geometry, *IEEE Transactions on Biomedical Engineering*, *59*(4), 920–928.

Birdsall, T. G. (1973), The theory of signal detectability: Roc curves and their character, *Tech. rep.*, MICHIGAN UNIV ANN ARBOR COOLEY ELECTRONICS LAB.

Bozinovski, S., and A. Fulgosi (1976), The influence of pattern similarity and transfer learning upon training of a base perceptron b2, in *Proceedings of Symposium Informatica*, pp. 3–121.

Brunner, P., S. Joshi, S. Briskin, J. R. Wolpaw, H. Bischof, and G. Schalk (2010), Does the 'p300' speller depend on eye gaze?, *Journal of Neural Engineering*, *7*(5), 056,013.

Cai, Q., J. Kang, and T. Yu (2020), Bayesian network marker selection via the thresholded graph laplacian gaussian prior, *Bayesian Analysis*, *15*(1), 79.

Cecotti, H., and A. Graser (2010), Convolutional neural networks for p300 detection with application to brain-computer interfaces, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(3), 433–445.

Congedo, M., A. Barachant, and A. Andreev (2013), A new generation of brain-computer interface based on riemannian geometry, *arXiv preprint arXiv:1310.8115*.

Dal Seno, B., M. Matteucci, and L. T. Mainardi (2009), The utility metric: a novel method to assess the overall performance of discrete brain–computer interfaces, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *18*(1), 20–28.

Dawid, A. P. (1981), Some matrix-variate distribution theory: notational considerations and a bayesian application, *Biometrika*, *68*(1), 265–274.

Donchin, E., K. M. Spencer, and R. Wijesinghe (2000), The mental prosthesis: assessing the speed of a p300-based brain-computer interface, *IEEE Transactions on Rehabilitation Engineering*, *8*(2), 174–179.

D'Avanzo, C., S. Schiff, P. Amodio, and G. Sparacino (2011), A bayesian method to estimate single-trial event-related potentials with application to the study of the p300 variability, *Journal of Neuroscience Methods*, *198*(1), 114–124.

Farwell, L. A., and E. Donchin (1988), talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials, *Electroencephalography and Clinical Neurophysiology*, *70*(6), 510–523.

Folstein, J. R., and C. Van Petten (2008), Influence of cognitive control and mismatch on the n2 component of the erp: A review, *Psychophysiology*, *45*(1), 152–170.

Gelman, A., and D. B. Rubin (1992), Inference from iterative simulation using multiple sequences, *Statistical Science*, *7*(4), 457–472.

Gilks, W. R., and P. Wild (1992), adaptive rejection sampling for gibbs sampling, *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *41*(2), 337–348.

Hoffmann, U. (2007), Bayesian machine learning applied in a brain-computer interface for disabled users, *Tech. rep.*, EPFL.

Hoffmann, U., J.-M. Vesin, T. Ebrahimi, and K. Diserens (2008), An efficient p300-based brain-computer interface for disabled subjects, *Journal of Neuroscience Methods*, *167*(1), 115–125.

Jasper, H. H. (1958), The ten-twenty electrode system of the international federation, *Electroencephalography and Clinical Neurophysiology*, *10*, 370–375.

Johnson, G. D., and D. J. Krusienski (2009), Ensemble swlda classifiers for the p300 speller, in *International Conference on Human-Computer Interaction*, pp. 551–557, Springer.

Kalika, D., L. M. Collins, C. S. Throckmorton, and B. O. Mainsah (2017), Adaptive stimulus selection in erp-based brain-computer interfaces by maximizing expected discrimination gain, in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1405–1410, IEEE.

Kang, J., B. J. Reich, and A.-M. Staicu (2018), Scalar-on-image regression via the soft-thresholded gaussian process, *Biometrika*, *105*(1), 165–184.

Kaper, M., P. Meinicke, U. Grossekathoefer, T. Lingner, and H. Ritter (2004), Bci competition 2003-data set iib: support vector machines for the p300 speller paradigm, *IEEE Transactions on Biomedical Engineering*, *51*(6), 1073–1076.

Kass, R. E., and A. E. Raftery (1995), Bayes factors, *Journal of the American Statistical Association*, *90*(430), 773–795.

Khazem, S., S. Chevallier, Q. Barthélemy, K. Haroun, and C. Noûs (2021), Minimizing subject-dependent calibration for bci with riemannian transfer learning, in *2021 10th International IEEE/EMBS Conference on Neural Engineering (NER)*, pp. 523–526, IEEE.

Komiyama, J., J. Honda, and H. Nakagawa (2015), Optimal regret analysis of thompson sampling in stochastic multi-armed bandit problem with multiple plays, in *International Conference on Machine Learning*, pp. 1152–1161, PMLR.

Krusienski, D. J., E. W. Sellers, D. J. McFarland, T. M. Vaughan, and J. R. Wolpaw (2008), Toward enhanced p300 speller performance, *Journal of Neuroscience Methods*, *167*(1), 15–21.

Lenhardt, A., M. Kaper, and H. J. Ritter (2008), An adaptive p300-based online brain–computer interface, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *16*(2), 121–130.

Lenzerini, M. (2002), Data integration: A theoretical perspective, in *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 233–246.

Leoni, J., S. C. Strada, M. Tanelli, K. Jiang, A. Brusa, and A. M. Proverbio (2021), Automatic stimuli classification from erp data for augmented communication via brain-computer interfaces, *Expert Systems with Applications*, p. 115572.

Li, F., Y. Xia, F. Wang, D. Zhang, X. Li, and F. He (2020), Transfer learning algorithm of p300-eeg signal based on xdawn spatial filter and riemannian geometry classifier, *Applied Sciences*, *10*(5), 1804.

Li, Y., and S. K. Ghosh (2015), Efficient sampling methods for truncated multivariate normal and student-t distributions subject to linear inequality constraints, *Journal of Statistical Theory and Practice*, *9*(4), 712–732.

Luck, S. J. (2014), *An introduction to the event-related potential technique*, 7–9 pp., MIT press.

Ma, R., N. Aghasadeghi, J. Jarzebowski, T. Bretl, and T. P. Coleman (2011), A stochastic control approach to optimally designing hierarchical flash sets in p300 communication prostheses, *IEEE transactions on neural systems and rehabilitation engineering*, *20*(1), 102–112.

Martens, S., N. Hill, J. Farquhar, and B. Schölkopf (2009), Overlap and refractory effects in a brain-computer interface speller based on the visual p300 event-related potential, *Journal of Neural Engineering*, *6*(2), 026,003.

McCann, M. T., D. E. Thompson, Z. H. Syed, and J. E. Huggins (2015), Electrode subset selection methods for an eeg-based p300 brain-computer interface, *Disability and Rehabilitation: Assistive Technology*, *10*(3), 216–220.

Mowla, M. R., J. E. Huggins, B. Natarajan, and D. E. Thompson (2018), P300 latency estimation using least mean squares filter, in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1976–1979, IEEE.

Okumuş, H., and Ö. Aydemır (2017), Random forest classification for brain computer interface applications, in *2017 25th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE.

Onishi, A. (2020), Convolutional neural network transfer learning applied to the affective auditory p300-based bci, *Journal of Robotics and Mechatronics*, *32*(4), 731–737.

Onishi, A., and K. Natsume (2014), Overlapped partitioning for ensemble classifiers of p300-based brain-computer interfaces, *PloS one*, *9*(4), e93,045.

Park, J., and K.-E. Kim (2012), A pomdp approach to optimizing p300 speller bci paradigm, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *20*(4), 584–594.

Pavarini, S. C. I., et al. (2018), On the use of the p300 as a tool for cognitive processing assessment in healthy aging: A review, *Dementia & neuropsychologia*, *12*(1), 1–11.

Polich, J., L. Howard, and A. Starr (1985), effects of age on the p300 component of the event-related potential from auditory stimuli: peak definition, variation, and measurement, *Journal of Gerontology*, *40*(6), 721–726.

PRC-Saltillo, O., Wooster (2009), Nuvoice software with unity language encoding.

Qiao, X., S. Guo, and G. M. James (2019), Functional graphical models, *Journal of the American Statistical Association*, *114*(525), 211–222.

Rakotomamonjy, A., and V. Guigue (2008), Bci competition iii: dataset ii-ensemble of svms for bci p300 speller, *IEEE transactions on biomedical engineering*, *55*(3), 1147–1154.

Rasmussen, C. E. (2003), Gaussian processes in machine learning, in *Summer school on machine learning*, pp. 63–71, Springer.

Rivet, B., A. Souloumiac, V. Attina, and G. Gibert (2009), xdawn algorithm to enhance evoked potentials: application to brain–computer interface, *IEEE Transactions on Biomedical Engineering*, *56*(8), 2035–2043.

Rodden, F. A., and B. Stemmer (2008), A brief introduction to common neuroimaging techniques, in *Handbook of the Neuroscience of Language*, pp. 57–67, Elsevier.

Rodrigues, P. L. C., C. Jutten, and M. Congedo (2018), Riemannian procrustes analysis: transfer learning for brain–computer interfaces, *IEEE Transactions on Biomedical Engineering*, *66*(8), 2390–2401.

Russo, D., B. Van Roy, A. Kazerouni, I. Osband, and Z. Wen (2017), A tutorial on thompson sampling, *arXiv preprint arXiv:1707.02038*.

Said, S., L. Bombrun, Y. Berthoumieu, and J. H. Manton (2017), Riemannian gaussian distributions on the space of symmetric positive definite matrices, *IEEE Transactions on Information Theory*, *63*(4), 2153–2170.

Schalk, G., D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw (2004), Bci2000: A general-purpose brain-computer interface (bci) system, *IEEE Transactions on Biomedical Engineering*, *51*(6), 1034–1043.

Stephens, M. (2000), Dealing with label switching in mixture models, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *62*(4), 795–809.

Thompson, D. E., K. L. Gruis, and J. E. Huggins (2014), A plug-and-play brain-computer interface to operate commercial assistive technology, *Disability and Rehabilitation: Assistive Technology*, *9*(2), 144–150.

Thompson, W. R. (1933), On the likelihood that one unknown probability exceeds another in view of the evidence of two samples, *Biometrika*, *25*(3/4), 285–294.

Throckmorton, C. S., K. A. Colwell, D. B. Ryan, E. W. Sellers, and L. M. Collins (2013), Bayesian approach to dynamically controlling data collection in p300 spellers, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *21*(3), 508–517.

Townsend, G., et al. (2010), A novel p300-based brain-computer interface stimulus presentation paradigm: Moving beyond rows and columns, *Clinical Neurophysiology*, *121*(7), 1109–1120.

van Dinteren, R., M. Arns, M. L. Jongsma, and R. P. Kessels (2014), P300 development across the lifespan: A systematic review and meta-analysis, *PloS one*, *9*(2), e87,347.

Viana, S. S., D. M. Batista, and D. B. Melges (2014), Logistic regression models: Feature selection for p300 detection improvement, in *XXIV Brazilian Congress on Biomedical Engineering–CBEB*, vol. 2014, pp. 979–982.

Völker, M., R. T. Schirrmeister, L. D. Fiederer, W. Burgard, and T. Ball (2018), Deep transfer learning for error decoding from non-invasive eeg, in *2018 6th International Conference on Brain-Computer Interface (BCI)*, pp. 1–6, IEEE.

Wolpaw, J. R., H. Ramoser, D. J. McFarland, and G. Pfurtscheller (1998), Eeg-based communication: improved accuracy by response verification, *IEEE transactions on Rehabilitation Engineering*, *6*(3), 326–333.

Wolpaw, J. R., N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan (2002), Brain-computer interfaces for communication and control, *Clinical neurophysiology*, *113*(6), 767–791.

Wolpaw, J. R., et al. (2018), Independent home use of a brain-computer interface by people with amyotrophic lateral sclerosis, *Neurology*, *91*(3), e258–e267.

Xu, M., J. Liu, L. Chen, H. Qi, F. He, P. Zhou, X. Cheng, B. Wan, and D. Ming (2015), Inter-subject information contributes to the erp classification in the p300 speller, in *2015 7th International IEEE/EMBS Conference on Neural Engineering (NER)*, pp. 206–209, IEEE.

Xu, M., J. Liu, L. Chen, H. Qi, F. He, P. Zhou, B. Wan, and D. Ming (2016), Incorporation of inter-subject information to improve the accuracy of subject-specific p300 classifiers, *International journal of neural systems*, *26*(03), 1650,010.

Xu, N., X. Gao, B. Hong, X. Miao, S. Gao, and F. Yang (2004), Bci competition 2003-data set iib: Enhancing p300 wave detection using ica-based subspace projections for bci applications, *IEEE Transactions on Biomedical Engineering*, *51*(6), 1067–1072.

Zanini, P., M. Congedo, C. Jutten, S. Said, and Y. Berthoumieu (2016), Parameters estimate of riemannian gaussian distribution in the manifold of covariance matrices, in *2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pp. 1–5, IEEE.