2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) | 978-1-6654-0126-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/BIBM52615.2021.9669724

Adaptive Sequence-Based Stimulus Selection in an ERP-Based Brain-Computer Interface by Thompson Sampling in a Multi-Armed Bandit Problem

1st Tianwen Ma
Dept. of Biostatistics
University of Michigan
Ann Arbor, USA
mtianwen@umich.edu

2nd Jane E. Huggins
Dept. of Physical Med. & Rehabilitation
University of Michigan
Ann Arbor, USA
janeh@umich.edu

3rd Jian Kang
Dept. of Biostatistics
University of Michigan
Ann Arbor, USA
jiankang@umich.edu

***Abstract*—A Brain-Computer Interface (BCI) is a device that interprets brain activity to help people with disabilities communicate. The P300 ERP-based BCI speller displays a series of events on the screen and searches the elicited electroencephalogram (EEG) data for target P300 event-related potential (ERP) responses among a series of non-target events. The Checkerboard (CB) paradigm is a common stimulus presentation paradigm. Although a few studies have proposed data-driven methods for stimulus selection, they suffer from intractable decision rules, large computation complexity, or error propagation for participants who perform poorly under the static paradigm. In addition, none of the methods have been applied to the CB paradigm directly. In this work, we propose a sequence-based adaptive stimulus selection method using Thompson Sampling in the multi-bandit problem with multiple actions. During each sequence, the algorithm selects a random subset of stimuli with fixed size, aiming to identify all target stimuli and to improve the spelling speed by reducing the number of unnecessary non-target stimuli. We compute "clean" stimulus-specific rewards from raw classifier scores via the Bayes rule. We perform extensive simulation studies to compare our algorithm to the static CB paradigm. We show the robustness of our algorithm by considering the constraints of practical use. For scenarios where simulated data resemble the real data the most, the spelling efficiency of our algorithm increases by more than 70%, compared to the static CB paradigm.**

***Index Terms*—Adaptive Stimulus Selection, Thompson Sampling, Checkerboard Paradigm, Brain-Computer Interface**

I. 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 [1]. One of the most popular non-invasive BCIs is the P300 ERP-based BCI speller [2] 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 [3].

In a visual P300 ERP-BCI speller, a virtual keyboard is presented to the participant (See Figure 2). 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 [2] 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. [4] applied the partially observable Markov decision process (POMDP) to compute an optimal stimulus schedule under the RC paradigm. Ma et al. [5] 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. [6] 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 static 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 [7].

In this paper, we propose a sequence-based adaptive stimulus selection method by framing the problem as a multi-armed bandit problem with multiple actions [8]. 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 the spelling speed by reducing the number of unnecessary non-target stimulus groups. We apply Thompson Sampling to achieve this goal [9]. We perform extensive simulation studies based on the CB paradigm and demonstrate the robustness of our algorithm by considering both ideal and practical scenarios.

The rest of this paper is organized as follows: Section II presents background information relevant to our algorithm. Section III introduces the proposed algorithm in detail. Section IV presents the numerical results of the simulation studies. Finally, Section VI concludes our paper with a brief discussion.

II. Background

A. The Checkerboard (CB) Paradigm

In this work, we develop our adaptive stimulus selection algorithm based upon the CB paradigm introduced by [7]. The traditional RC paradigm is susceptible to error propagation that leads to attention shifts and frustration for two primary reasons [7]. 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 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. We end up with H1R1, ..., H1R3, H1C1, ..., H1C3, H2R1, ..., H2R3, and H2C1, ..., H2C3. Each element is 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.

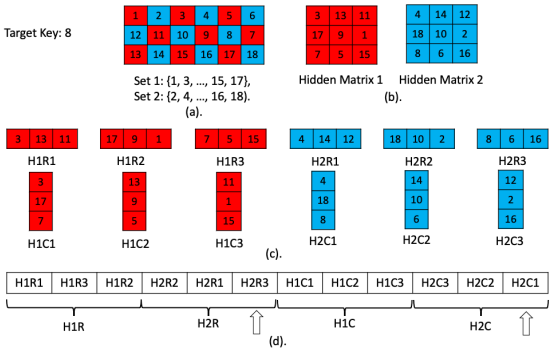

Fig. 1. An illustration of the checkerboard design. (a). A 3×6 keyboard with 18 keys labelled from 1 to 18 in a row switchback order. (b). The keyboard is split into two sets (red and blue) and placed in hidden matrices 1 and 2. (c). We extract rows and columns from two hidden matrices to form 12 stimulus 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.

B. 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 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 [9] 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 [8] 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 [10]). The number of total actions K is the number of stimulus groups that divide the entire virtual keyboard (See Section III-B). 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.

C. Bayesian Dynamic Stopping Criterion

One of the most important aspects about data-driven stimulus selection methods is the dynamic data collection. Past work in [11] 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 [12] 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.

III. Proposed Algorithm

A. Assumptions

First, 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. Second, we do not incorporate the impact of the practical constraints directly into the proposed algorithm. However, we address the modifications for practical implementations in Section IV.

B. 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.

C. 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. Let $\boldsymbol{I}(t)$ and $\boldsymbol{\mathcal{F}}_t$ be the indices of the L samples with the largest values and the corresponding subset of selected stimulus groups during sequence t, respectively. 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. Here, we introduce one way to produce 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.

$$z_{t,l} \sim \begin{cases} \mathcal{N}(\mu_1, \sigma^2), & \omega \in S_l \\ \mathcal{N}(\mu_0, \sigma^2), & \omega \notin S_l. \end{cases} \tag{1}$$

In practice, obtaining a classifier score is equivalent to a series of processes that include the brain response to stimulus groups, the feature extraction and segmentation, application to binary classifiers. We will discuss the alternative way in Section IV-C. Then, we compute the "clean" rewards for each character $n, n = 1, \cdots, N$.

$$\begin{aligned} P_{t,n} &= \frac{\prod_{l=1}^{L} \mathcal{L}_{t,l,n}(z_{t,l}) P_{t-1,n}}{\sum_{c=1}^{N} \prod_{l=1}^{L} \mathcal{L}_{t,l,c}(z_{t,l}) P_{t-1,c}}, \\ \mathcal{L}_{t,l,n}(z_{t,l}) &= \begin{cases} \mathcal{L}_0(z_{t,l}), & n \notin S_{I_l(t)}, \\ \mathcal{L}_1(z_{t,l}), & n \in S_{I_l(t)}, \end{cases} \end{aligned} \tag{2}$$

where $\mathcal{L}_1, \mathcal{L}_0$, 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

$$r_{t,k} = \sum_{n=1}^{N} P_{t,n} \cdot \mathbb{I}_{S_k}(n), \quad k = 1, \cdots, K. \tag{3}$$

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

$$\alpha_k \leftarrow \alpha_k + r_{t,k}, \quad \beta_k \leftarrow \beta_k + 1 - r_{t,k}. \tag{4}$$

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

Algorithm 1:

Input: The 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
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
  $\boldsymbol{I}(t) =$ 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 brain responses indexed by $\boldsymbol{I}(t)$.
  end
  $\boldsymbol{P}_{t+1} \overset{NBDSA}{\longleftarrow} \boldsymbol{P}_t, \{z_{t,l}\}, \mathcal{L}_0, \mathcal{L}_1$ (See Eq.2)
  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
end

D. Practical Constraints

During online BCI implementation, we accommodate two constraints. Since the CB paradigm avoids the impact of the double-flash problem, we primarily consider another systematic constraint due to the data generation mechanism. On one hand, in order to produce a classifier score, the time window of EEG signals associated with each stimulus presentation requires filtering, segmentation, and application to the binary classifier. On the other hand, the time window of EEG signals is generally shorter than the time interval between adjacent stimulus groups. Therefore, there always exists a response delay between the presentation of stimulus group during sequence t and its corresponding classifier score. We address the constraint of the response delay by applying a policy called "cross iteration update" (See Figure 3).

IV. Numerical Experiment and Results

We perform simulation studies of the P300 speller character selection process to compare different configurations of our proposed algorithm to the static CB paradigm. We base the speller grid on Figure 2 [13] , where L ranges from 4 to 26. When $L = 26$, it is equivalent to the static CB paradigm because no subset selection process is involved. We design three scenarios: classifier score-based without response delay, classifier score-based with response delay, and EEG signal-based with response delay.

For the first two scenarios, we follow the framework in [6] 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,

$$d' = \frac{\mu_1 - \mu_0}{\sigma}, \tag{5}$$

where d' is the *detectability index* defined in [14]; 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 produce various cases of simulation studies. For the third scenario, we generate EEG signals by aligning simulated ERP responses according to their stimulus type indicators. We add additional Gaussian noise with variance σ_X^2 and auto-correlation parameter q to simulate the background activity. Next, we extract the EEG signal segments from the onset of the stimulus group with a fixed EEG window (e.g., 800ms) as the feature vectors. Then, we apply the the stepwise linear discriminant analysis (swLDA) method [15][16] to obtain the classifier scores z. We assume that the swLDA model has been trained from additional dataset generated by the same simulated ERP responses and noise structure. The additional dataset for each case has the same dimension. We vary the parameter σ_X^2 and subset size L to produce various cases of simulation studies. In each case, we assume one target character to type (e.g., character index of 70), and we do not incorporate any prior information into any candidate characters. For the last two scenarios, we randomly specify the selected stimulus groups for the first two sequences to account for the delay.

We define the probability threshold $p_{\max} = 0.9$ and the sequence upper threshold $T_0 = 5$. Since L is an input for the algorithm, we evaluate the spelling time in the final results for consist comparison. The spelling time is defined as the product among the number of sequences, subset size L, and the time interval between adjacent stimulus groups, where this time interval is set to 160 ms. We report the final accuracy and the spelling time to satisfy the above criteria jointly. In addition, we report the information transfer rate (ITR) as in [17] and the BCI utility (Utility) as in [18] to provide more comprehensive metrics that combine accuracy and speed. We present the metrics for $L = 5, 10, 13$, and 26. For each case, we repeat the case 500 times and report the average results across replications.

In Section IV-A, we present the results for simulation studies based on classifier scores without the response delay. In Section IV-B, we present the results for simulation studies based on classifier scores with the response delay. Finally, in Section IV-C, we present the results where simulation studies are based on EEG signal segments with the response delay.

A. Classifier Score-based without Response Delay

Assuming no response delay, we perform simulation studies to compare the performance of subset selection to the

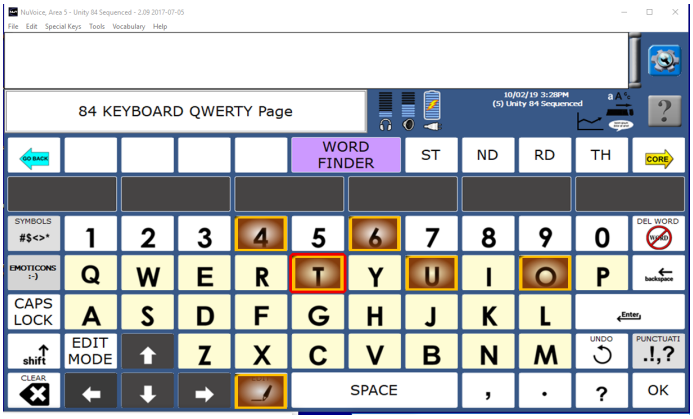

Fig. 2. An illustration of an 84-key keyboard with a combination of characters, denoted as a stimulus group, being highlighted. Under the CB paradigm, stimulus groups are not necessarily row and column arranged.

static CB paradigm. We fix $\mu_1 = 0.50$ and $\mu_0 = -0.20$, and varied σ among $\{0.20, 0.30, 0.40, 0.50, 0.60\}$ ($d' = 3.50, 2.33, 1.75, 1.40, 1.17$). Notice that the combination of $\mu_1 = 0.5, \mu_0 = -0.2$, and $\sigma = 0.3$ or equivalently, $d' = 2.33$ is based on the result of a real participant, and we vary σ to demonstrate the robustness of our method. Table I shows the mean accuracy, spelling time, ITR, and Utility metrics of Scenario A under various σ across 500 replications. As the subset size L increases from 5 to 26, the probability of correctly selecting the target character, ITR, and Utility decreases, while the spelling time to reach the stopping criterion increases. This suggests that our algorithm is more efficient than the static CB paradigm.

The results of Scenario A can be considered as an upper bound of the performance evaluation. In Scenarios B and C, we evaluate the robustness of our algorithm in response to the response delay from two data generative mechanisms.

B. Classifier Score-based with Response Delay

Since the CB paradigm avoids the "double flash" problem mentioned above, we consider the response delay for the practical implementation. We perform simulation studies to compare the performance of subset selection to the static CB paradigm with the same parameter set in Section IV-A. We assume that the data collection, classifier score generation, and posterior sampling associated with sequence t will be completed by the end of sequence $(t+1)$, denoted as "cross iteration update" (See Figure 3). Thus, we randomly initialize stimulus groups for the first two sequences to compensate for the response delay. For sequence $t, t > 2$, we generate the stimulus groups based on the Thompson sampling results of the posterior samples $\boldsymbol{\alpha}_{t-2}$ and $\boldsymbol{\beta}_{t-2}$. Table II shows the mean accuracy, spelling time, ITR, and Utility of Scenario B under various σ across 500 replications. Similar patterns to Table I are observed. The ITR, Utility, accuracy, and spelling time in Scenario B are, on average, lower than those in Scenario A because by assumption, we record the probability of the sequence prior to the one that reaches the stopping criterion.

TABLE I

A TABLE OF THE EVALUATION METRICS FOR THE CLASSIFIER SCORE-BASED SCENARIO WITHOUT RESPONSE DELAY. METRICS INCLUDE ACCURACY, SPELLING TIME, ITR, AND UTILITY. WE VARY THE DETECTABILITY INDEX d' AND THE SUBSET SIZE L. THE RESULTS FOR $L = 26$ ARE THE BASELINE WHERE NO ADAPTIVE STIMULUS SELECTION IS INVOLVED.

d'	Subset Size L	ITR	Utility	Accuracy	Time (ms)
3.50	5	0.95	0.94	0.97	5920
3.50	10	0.74	0.74	0.99	6880
3.50	13	0.65	0.64	0.99	7680
3.50	26	0.50	0.50	0.99	9440
2.33	5	0.69	0.69	0.96	7520
2.33	10	0.53	0.53	0.97	9120
2.33	13	0.49	0.49	0.97	10080
2.33	26	0.35	0.34	0.97	13600
1.75	5	0.47	0.47	0.92	10400
1.75	10	0.38	0.38	0.93	12480
1.75	13	0.34	0.34	0.93	13440
1.75	26	0.23	0.23	0.87	17600
1.20	5	0.35	0.35	0.87	13120
1.20	10	0.26	0.25	0.83	16000
1.20	13	0.24	0.23	0.81	16000
1.20	26	0.14	0.13	0.67	19200
1.17	5	0.26	0.25	0.79	15520
1.17	10	0.18	0.17	0.70	17600
1.17	13	0.14	0.13	0.63	19200
1.17	26	0.08	0.07	0.47	20800

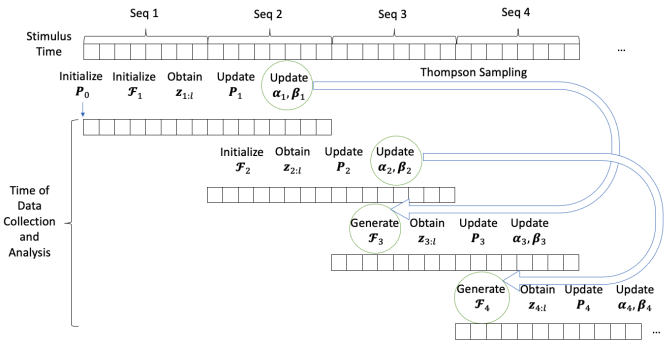

Fig. 3. A figure of the simulated data generation mechanism with the response delay. The upper panel is the time frame for stimulus presentation, while the lower panel is the time frame of data collection and analysis for each sequence. Let $\boldsymbol{\mathcal{F}}_t$ be the set of stimulus groups for sequence t. 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 t would will be completed by the end of sequence $(t+1)$, denoted as "cross iteration update". For sequence $t, t > 2$, we generate the stimulus groups based on posterior samples of TS, $\boldsymbol{\alpha}_{t-2}, \boldsymbol{\beta}_{t-2}$. The process is not terminated until the stopping criterion is reached.

However, the results between Scenarios A and B do not differ very much.

C. EEG Signal-based with Response Delay

Finally, we extend Scenario B where we start from simulated EEG time series. We apply the same updating policy as in Section IV-B. We assume that the simulated EEG signals have an additive signal-and-noise effect. For the signal component, we start with simulated ERP responses to target and non-target stimuli by the type of stimuli (See Figure 4), and align them by the rule of convolution. For the noise component, we assume

TABLE II

A TABLE OF THE EVALUATION METRICS FOR THE CLASSIFIER SCORE-BASED SCENARIO WITH RESPONSE DELAY. METRICS INCLUDE ACCURACY, SPELLING TIME, ITR, AND UTILITY. WE VARY THE DETECTABILITY INDEX d' AND THE SUBSET SIZE L. THE RESULTS FOR $L = 26$ ARE THE BASELINE WHERE NO ADAPTIVE STIMULUS SELECTION IS INVOLVED.

d'	Subset Size L	ITR	Utility	Accuracy	Time (ms)
3.50	5	0.80	0.79	0.98	6240
3.50	10	0.60	0.59	0.98	8000
3.50	13	0.54	0.54	0.98	8640
3.50	26	0.35	0.35	0.99	13280
2.33	5	0.59	0.59	0.97	8160
2.33	10	0.43	0.43	0.97	10720
2.33	13	0.38	0.38	0.97	11840
2.33	26	0.26	0.26	0.95	17600
1.75	5	0.42	0.42	0.95	11040
1.75	10	0.31	0.30	0.93	14560
1.75	13	0.28	0.27	0.91	15680
1.75	26	0.18	0.17	0.80	19200
1.20	5	0.33	0.33	0.91	13600
1.20	10	0.23	0.23	0.86	17600
1.20	13	0.19	0.19	0.80	19200
1.20	26	0.10	0.09	0.55	20800
1.17	5	0.25	0.24	0.82	16000
1.17	10	0.15	0.13	0.65	19200
1.17	13	0.12	0.10	0.59	19200
1.17	26	0.05	0.04	0.37	20800

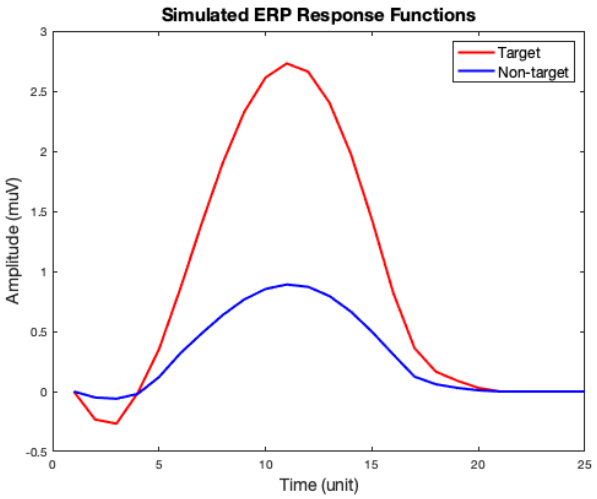

Fig. 4. A figure of simulated ERP response functions to target (red) and non-target (blue) in Scenario C. Both ERP response functions contain 25 time points, with the unit time interval representing 32 ms. The peak ratio between target and non-target stimuli is around 3. We assume that the time interval between adjacent stimuli are 160ms, and the extracted EEG response window has the same length as the simulated ERP response functions.

it follows a Gaussian distribution with an auto-correlation structure $AR(q)$ and the variance σ_X^2. For this study, we only generate one-dimension simulated data (one electrode), so no spatial correlation is considered. Then, we extract the EEG signal segments from the onset of each stimulus with the fixed response window (i.e., 800 ms) 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 from the training set. 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 $\{0.1, 2, 5.5, 12.5, 20\}$ to match d' approximately for the consistent comparison. Table III shows the means and standard deviations of accuracy, spelling time, ITR, and TTI of Scenario C under various σ_X^2 across 500 replications. Notice that the actual d' may deviate from the d' in Scenarios A and B due to the randomness in the training set.

TABLE III

A TABLE OF THE EVALUATION METRICS FOR THE EEG SIGNAL-BASED SCENARIO WITH RESPONSE DELAY. METRICS INCLUDE ACCURACY, SPELLING TIME, ITR, AND UTILITY. WE VARY THE NOISE VARIANCE σ_X^2 TO MATCH THE DETECTABILITY INDEX d' AND THE SUBSET SIZE L. THE RESULTS FOR $L = 26$ ARE THE BASELINE WHERE NO ADAPTIVE STIMULUS SELECTION IS INVOLVED. THE ACTUAL d' MAY BE DIFFERENT FROM THE VALUES IN SCENARIOS A AND B DUE TO THE RANDOMNESS IN THE TRAINING SET.

d'	Subset Size L	ITR	Utility	Accuracy	Time (ms)
3.51	5	0.87	0.86	0.99	5920
3.48	10	0.68	0.68	0.99	7200
3.48	13	0.61	0.60	1.00	7840
3.47	26	0.38	0.37	1.00	12480
2.39	5	0.62	0.62	0.97	7680
2.44	10	0.46	0.46	0.97	9920
2.22	13	0.42	0.42	0.96	10560
2.24	26	0.27	0.26	0.96	16000
1.65	5	0.46	0.46	0.96	9920
1.63	10	0.33	0.32	0.95	13600
1.78	13	0.29	0.29	0.94	15040
1.76	26	0.18	0.17	0.81	19200
1.20	5	0.34	0.34	0.92	12960
1.25	10	0.20	0.19	0.80	17600
1.39	13	0.19	0.18	0.79	19200
1.27	26	0.09	0.07	0.52	20800
1.03	5	0.28	0.28	0.88	15040
1.04	10	0.18	0.17	0.75	19200
1.02	13	0.14	0.12	0.67	19200
1.09	26	0.05	0.04	0.37	20800

V. DISCUSSION

Although the ITR and Utility increase by more than 70% if we compare the $L = 5$ to $L = 26$ for the scenarios where parameter set resembles the real data the most (i.e., $d' = 2.33$), we find that smallest L does not necessarily always produce the best prediction accuracy, especially for the scenario of $d' = 3.50$. One explanation that smallest L does not produce the best prediction accuracy is that it may be difficult to initialize both target stimulus groups for the first two sequences. Therefore, applying a statistical language modeling to initialize $\boldsymbol{P}_0$ prior to the experiment will

compensate for the disadvantage. In addition, although the CB paradigm completely removes the impact of the double-flash problem, using too small L may make it difficult to create enough time gap between adjacent target stimuli in practice. One possible modification is to choose a moderate size of L (i.e., $L = 10$ in our setting) so that a minimum of certain stimulus groups are, on average, selected from four hidden sets to maintain a reasonable target-to-target interval (TTI). Finally, when evaluating the applicability of these scenarios to actual use, we consider that Scenarios A and B are based on the classifier scores, and Scenario B considers the response delay. Scenario C is based on synthesized EEG data with the response delay. Although the distributions of classifier scores conditional on the type of stimulus groups from synthesized EEG signals are similar to the ones from real participants, the synthesized EEG signals in Scenario C may neglect certain aspects of real EEG data. Therefore, we will follow the steps in [19] and consider using previously recorded EEG signals in a simulation-based environment for future work.

VI. Conclusion

We propose a sequence-based adaptive stimulus selection algorithm based on Thompson Sampling under the framework of a multi-bandit problem with multiple selections. The algorithm selects a random subset of stimuli with fixed size during each sequence, aiming to identify all target stimulus groups and to improve the spelling speed by reducing the number of unnecessary non-target stimulus groups. For the scenario where parameter set resembles the real data the most, both ITR and Utility increase by 70.4% and 76.9%, respectively. We compute "clean" rewards from raw classifier scores via the Bayes rule to further improve the spelling efficiency. We perform extensive simulation studies to compare our algorithm to the static CB paradigm. We also show the robustness of our algorithm by considering the physiological and practical constraints in real-time BCI implementations. In future work, we will test our algorithm on data from real participants.

References

[1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain–computer interfaces for communication and control," *Clinical Neurophysiology*, vol. 113, no. 6, pp. 767–791, 2002.

[2] L. A. Farwell and E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalography and Clinical Neurophysiology*, vol. 70, no. 6, pp. 510–523, 1988.

[3] F. A. Rodden and B. Stemmer, "A Brief Introduction to Common Neuroimaging Techniques," in *Handbook of the Neuroscience of Language*. Elsevier, 2008, pp. 57–67.

[4] J. Park and K.-E. Kim, "A pomdp approach to optimizing p300 speller bci paradigm," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 20, no. 4, pp. 584–594, 2012.

[5] R. Ma, N. Aghasadeghi, J. Jarzebowski, T. Bretl, and T. P. Coleman, "A stochastic control approach to optimally designing hierarchical flash sets in p300 communication prostheses," *IEEE Transactions on Neural Systems and Rehabilitation engineering*, vol. 20, no. 1, pp. 102–112, 2011.

[6] D. Kalika, L. M. Collins, C. S. Throckmorton, and B. O. Mainsah, "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)*. IEEE, 2017, pp. 1405–1410.

[7] G. Townsend, B. K. LaPallo, C. B. Boulay, D. J. Krusienski, G. Frye, C. Hauser, N. E. Schwartz, T. M. Vaughan, J. R. Wolpaw, and E. W. Sellers, "A novel p300-based brain-computer interface stimulus presentation paradigm: Moving beyond rows and columns," *Clinical Neurophysiology*, vol. 121, no. 7, pp. 1109–1120, 2010.

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

[9] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3/4, pp. 285–294, 1933.

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

[11] A. Lenhardt, M. Kaper, and H. J. Ritter, "An adaptive p300-based online brain–computer interface," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 16, no. 2, pp. 121–130, 2008.

[12] C. S. Throckmorton, K. A. Colwell, D. B. Ryan, E. W. Sellers, and L. M. Collins, "Bayesian approach to dynamically controlling data collection in p300 spellers," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 21, no. 3, pp. 508–517, 2013.

[13] "Nuvoice prc application and support software (pass)," 1996. [Online]. Available: https://www.prentrom.com/support/PASS

[14] T. G. Birdsall, "The theory of signal detectability: Roc curves and their character," MICHIGAN UNIV ANN ARBOR COOLEY ELECTRONICS LAB, Tech. Rep., 1973.

[15] E. Donchin, K. M. Spencer, and R. Wijesinghe, "The mental prosthesis: Assessing the speed of a p300-based brain-computer interface," *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 2, pp. 174–179, 2000.

[16] D. J. Krusienski, E. W. Sellers, F. Cabestaing, S. Bayoudh, D. J. McFarland, T. M. Vaughan, and J. R. Wolpaw, "A comparison of classification techniques for the p300 speller," *Journal of Neural Engineering*, vol. 3, no. 4, p. 299, 2006.

[17] J. R. Wolpaw, H. Ramoser, D. J. McFarland, and

G. Pfurtscheller, "Eeg-based communication: improved accuracy by response verification," *IEEE Transactions on Rehabilitation Engineering*, vol. 6, no. 3, pp. 326–333, 1998.

[18] B. Dal Seno, M. Matteucci, and L. T. Mainardi, "The utility metric: a novel method to assess the overall performance of discrete brain–computer interfaces," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, no. 1, pp. 20–28, 2009.

[19] J. Mladenovic, J. Frey, M. Joffily, E. Maby, F. Lotte, and J. Mattout, "Active inference as a unifying, generic and adaptive framework for a p300-based bci," *Journal of Neural Engineering*, vol. 17, no. 1, p. 016054, 2020.