A Fully Unsupervised Online Classification Algorithm for Event-Related Potential based Brain-Computer Interfaces

Jing Jin, *Senior Member*, *IEEE*, Haoye Wang, Ian Daly, Xueqing Zhao, Shurui Li, and Andrzej Cichocki, *Life Fellow*, *IEEE*

***Abstract*— *Objective:* Brain-computer interfaces (BCIs) based on event-related potentials (ERPs) are among the most accurate and reliable BCIs. However, current mainstream classification algorithms struggle to eliminate the need for calibration and rely on expensive labeled data, limiting the practical usability of ERP-based BCIs. The development of fully unsupervised algorithms is essential for the advancement of practical applications of BCI systems. *Methods:* In this study, we propose a novel unsupervised classification method called sliding-window distribution distance maximization (sDDM). This algorithm utilizes sliding windows to highlight important temporal features and transforms the metric of inter-class differences from absolute distances to relative distribution distances in Mahalanobis space, while incorporating information on target event similarity from the BCI paradigm. Additionally, our proposed spatial dimensionality reduction strategy ensures smaller spatial dimensions and more prominent spatial features. *Results:* We compare our proposed method to other state-of-the-art unsupervised classification methods and evaluate it offline on our self-collected dataset, a public dataset recorded during the use of a P300 Speller by patients with ALS, and the BCI Competition III Dataset II. Our results demonstrate that our proposed method achieves the best spelling accuracy across all datasets, surpassing other unsupervised algorithms. We further explore its improvement effectiveness through ablation experiments. *Conclusion:* Our proposed method enhances the performance of unsupervised classification in ERP-based BCIs.**

***Index Terms*——Brain-computer interfaces, electroencephalography, event-related potential, unsupervised learning, sliding-window distribution distance maximization.**

I. Introduction

BRAIN-COMPUTER interfaces (BCIs) identify and convert brain signals into control commands, thereby establishing a direct pathway for information exchange between the human brain and computers or other electronic devices. This pathway can facilitate interaction with the external environment, particularly for individuals with mobility impairments [1], [2].

When some types of external stimuli are presented to a person, electroencephalogram (EEG) electrodes on the scalp can detect fluctuations in electro-potential associated with the stimulus event. This is known as an event-related potential (ERP) [3]. The timing of the ERP generation is synchronized with the timing of the stimulus event. For example, the P300 ERP component may be detected in the EEG approximately 300 ms after the occurrence of a deviant stimulus event [4], [5], [6]. As early as 1988, L.A. Farwell and E. Donchin introduced the ERP-based BCI (ERP-BCI) [7], which utilizes the ERP as a marker of brain activity, associating it with the timing of specific events, thus enabling the control of BCI systems through identification of the specific stimulus event the user is attending to.

To decode meaningful information from the brain signals obtained through BCIs, machine learning (ML) techniques are widely applied for feature extraction and classification of ERPs. ML techniques can be categorized into three types based on their learning approach: supervised, semi-supervised, and unsupervised [8].

Supervised learning algorithms require labeled data from either a single participant or a group of participants in order to train the classifier model to decode ERP signals. Examples

Manuscript received xxxx; revised xxxx; accepted xxxx. Date of publication xxxx; date of current version xxxx. This work was supported by Brain Science and Brain-like Intelligence Technology-National Science and Technology Major Project 2022ZD0208900 and National Natural Science Foundation of China under Grant 62176090; in part by Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX. This research is also supported by Project of Jiangsu Province Science and Technology Plan Special Fund in 2022 (Key research and development plan industry foresight, fundamental research fund for the central universities JKH01241605 and key core technologies) under Grant BE2022064-1; in part by the Lingang Laboratory under Grant No.LGL8998. *(Corresponding author: Jing Jin)*.

This work involved human subjects in its research. Approval of all ethical and experimental procedures and protocols was granted by the Bioethics Committee of East China University of Science and Technology, Shanghai, China.

Jing Jin is with the Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China and with the School of Mathematics, East China University of Science and Technology, Shanghai 200237, China (e-mail: jinjingat@gmail.com).

Haoye Wang and Xueqing Zhao are with the Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China (e-mail: haoyewang@hotmail.com; xueqingzhao2021@163.com).

Ian Daly is with the Brain-Computer Interfacing and Neural Engineering Laboratory, School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: i.daly@essex.ac.uk).

Shurui Li is with the School of Mathematics, East China University of Science and Technology, Shanghai 200237, China (shuruili1008@163.com).

Andrzej Cichocki is with the Systems Research Institute of Polish Academy of Sciences, 01-447bWarsaw, and Nicolaus Copernicus University (UMK), 87-100 Torun, Poland. (e-mail: cichockiand@gmail.com).

include traditional linear methods based on linear discriminant analysis (LDA) or support vector machine (SVM), such as stepwise LDA (SWLDA) [9], Bayesian LDA (BLDA) [10], regularized LDA [11], ToeplitzLDA [12], and ensemble of SVMs (ESVM) [13], [14], [15], as well as some more recent widely used neural network-based approaches like EEGNet [16], DeepConvNet [17], EEG-Inception [18], and ST-CapsNet [19]. However, this poses a crucial challenge for ERP-BCIs, namely the need for a calibration phase before participants can use the interface. This implies that participants must spend a certain amount of calibration time to train the classification models. For example, Jin et al. [20] employed a typical calibration model trained with LDA, which required 720 s of calibration time and achieved an average spelling accuracy of 80% across seven selected participants. In such cases, participants are required to follow the instructions provided by the system and cannot immediately control the BCI freely, which reduces the usability of BCI applications while increasing the psychological burden and fatigue experienced by users.

With a focus on reducing or eliminating calibration, many transfer learning and semi-supervised learning methods have been proposed. Transfer learning typically involves adjusting a classification model from a source domain using labeled data from the target domain to adapt to the target domain task [21], [22], [23]. In contrast, semi-supervised learning methods initially train the classification model using labeled data and then adjust it using unlabeled data [24], [25], [26], [27], [28]. However, these methods still fundamentally rely on utilizing labeled EEG data from other participants or from previous sessions recorded from the current participant. Consequently, transfer learning methods cannot eliminate calibration, and semi-supervised learning cannot immediately overcome the uncertainty of EEG features across participants or sessions. For instance, Jin et al. [22] proposed a generic model calibration method that reduced the calibration time by an average of 70.7% compared to the 276 s required by the typical calibration method. Similarly, Gao et al. [23] utilized the spelling of five characters as cross-subject calibration, requiring approximately 1 min and achieving a spelling accuracy of 93.50%. In addition, the acquisition cost of data poses limitations to the methods mentioned above. In addition to the difficulty in obtaining labeled data, these methods also require uniform specifications of input data features. This means that in supervised and semi-supervised learning methods, as well as in transfer learning methods, training and testing data must be input with entirely matching dimensional sizes, leading to non-generalizable data and increasing the cost of BCI usage. Hence, there is a necessity for the development of fully unsupervised methods for ERP-BCI systems.

Recently, Kindermann et al. [29] proposed an unsupervised classification method based on expectation maximization (EM). This algorithm is based on Bayesian least squares regression, alternately estimates the probability of alternative targets and optimizes the parameters given these probabilities [30]. Hübner et al. introduced learning from label proportions (LLP) [31] and a hybrid method of LLP and EM [32]. LLP modifies the paradigm by varying the probability of targets in different events, thereby establishing linear equations for target and non-target events and estimating the mean of targets and non-targets, ensuring convergence to the true mean after accumulating a sufficient number of epochs. However, LLP cannot be used for paradigms that do not conform to the above modification, thus limiting the range of applications it can be applied to. At the same time, this modification may lead to problems such as an increase in the number of necessary events and an increase in adjacent interference [33]. Sosulski et al. [34] proposed the unsupervised mean-difference maximization (UMM) method, which differs from classical binary classifiers in aggregating event classification results into multi-class decisions for a single spelling trial. Instead, it suggests using the data of one full trial, hypothetically selecting each possible symbol separately, and choosing the optimal hypothesis that maximizes the Mahalanobis distance between ERP target and non-target mean values. UMM achieves state-of-the-art classification accuracy among unsupervised learning algorithms. Although its performance is much lower when applied to datasets following auditory ERP protocols or those involving patient data, it has exceeded 99% accuracy across three visual ERP datasets involving healthy users. However, UMM only considers the Mahalanobis distance between two class means as a metric of inter-class difference, while lacking feature weight allocations in both temporal and spatial domains.

Therefore, in this work, we propose an improved method within the framework of UMM, called the sliding-window distribution distance maximization (sDDM) method. This method enhances the metric of inter-class differences by incorporating two pieces of information: the dispersion of the intra-class distribution and the similarity between multiple events within the target class, providing a more accurate assessment of the differences between targets and non-targets within the hypotheses. By introducing a sliding window to enhance the extraction of time-varying features, periods with significant inter-class differences are identified and given greater attention. Additionally, an unsupervised spatial dimensionality reduction strategy is proposed to reduce the spatial dimensions during the online process, thereby improving the signal-to-noise ratio of the EEG signal set.

II. Methodology

Our proposed sDDM algorithm, which we present in this paper, is a fully unsupervised learning algorithm designed for online processing and classification of EEG. We will provide an overview of the algorithm from two perspectives: the instantaneous process and the entire online decoding process.

A. Instantaneous Classification Process

In this study, the process by which the BCI makes a single decision (e.g., choosing a symbol) is referred to as a trial, while a segment of EEG signals used for binary classification of ERP is referred to as an epoch. Let all the epoch data input in a single trial be $\mathbf{X} \in \mathbb{R}^{C \times T \times S}$, where C denotes the number of channels, T denotes the number of time points per epoch, and S denotes the total number of epochs in a trial. First, $\mathbf{X}$ is zero-centered to remove the overall mean.

Let us consider a speller paradigm where the number of candidate symbols is denoted as N_s. In the stimulus sequence, there are N_e distinct stimulus events, each corresponding to a unique subset of highlighted symbols. Based on whether the target symbol is included in these subsets, the events can be classified as either target events or non-target events. Thus, all the epochs $\mathbf{X}$ in a single trial can be divided into target epochs $\mathbf{X}_1 \in \mathbb{R}^{C\times T\times S_1}$ and non-target epochs $\mathbf{X}_0 \in \mathbb{R}^{C\times T\times S_0}$ based on their classes.

To find the target symbol, all N_s symbols can be tested one by one, i.e., we assume the *m*-th symbol to be the correct symbol (referred to as the *m*-th hypothesis or hypothesis *m*, where *m* is a positive integer that ranges in value from 1 up to N_s) and evaluate the credibility of the classification under that hypothetical target. To achieve this, we iterate through all N_s possible hypotheses, and select the most credible hypothesis as the classification result. Under each hypothesis *m*, an epoch is assigned to the target epochs $\mathbf{X}_1$ if it contains the *m*-th symbol, and to the non-target epochs $\mathbf{X}_0$ otherwise.

1) Calculation of the Difference Between Classes under Hypothesis m

Under hypothesis *m*, the *a*-th kind of stimulus event that highlights the *m*-th symbol can be denoted as $f(a)$, each target epoch in $\mathbf{X}_1$ corresponds to a kind of stimulus event. When the target events consist of N_{te} kinds of event ($1 \le N_{te} < N_e$), $\mathbf{X}_1$ can be further divided into N_{te} subsets, denoted as $\mathbf{X}_{1,f(a)}$ ($a = 1,2,\dots,N_{te}$).

The squared Mahalanobis distance between the means of $\mathbf{X}_1$ and $\mathbf{X}_0$ has been proposed as a metric of the credibility of the hypothesis [34]. However, this metric ignores the following two points: 1) the intra-class distribution information for $\mathbf{X}_1$ and $\mathbf{X}_0$; and 2) the similarity between the epochs corresponding to different events (i.e., $f(1), f(2), \dots, f(N_{te})$) within $\mathbf{X}_1$. To fully utilize the above information, this paper proposes an improved inter-class difference metric:

$$D(m) = \frac{d_B^{f-f}}{d_W} \tag{1}$$

where $d_B^{f-f} \in \mathbb{R}$ is the improved inter-class squared distance, which reflects the pairwise dissimilarity among $\{f(1), f(2), \dots, f(N_{te})\}$. sDDM uses the similarity of target samples because they elicit consistent, temporally and spatially regular ERP components, making their similarity meaningful for the hypothesis. And, $d_W \in \mathbb{R}$ is the mean intra-class squared distance, which depends on the class assignment assumption m.

$$d_W = \sum_{k=0}^{1}\sum_{s=1}^{S_k} \frac{\sigma(\mathbf{X}_k^s - \bar{\mathbf{X}}_k)\boldsymbol{\Sigma}^{-1}(\sigma(\mathbf{X}_k^s - \bar{\mathbf{X}}_k))^T}{2S_k} \tag{2}$$

where S_k are the total number of epochs in class *k*, $\mathbf{X}_k^s \in \mathbb{R}^{C\times T}$ are the *s*-th epoch in class *k*, $\bar{\mathbf{X}}_k \in \mathbb{R}^{C\times T}$ are the mean of the epochs in class *k*, $\boldsymbol{\Sigma}$ is the covariance estimation matrix obtained from all the epoch data input in this trial $\mathbf{X} \in \mathbb{R}^{C\times T\times S}$, of size $CT \times CT$, and $\sigma(\cdot)$ represents the operation of flattening the matrix into a one-dimensional vector whose first dimension is 1. In addition, the bar at the top of all symbols herein denotes the mean of that subset of epochs.

$$d_B^{f-f} = \begin{cases} \sum_{a=1}^{N_{te}-1}\sum_{b=a+1}^{N_{te}} \dfrac{\sigma(\bar{\mathbf{X}}_{1,f(a)} - \bar{\mathbf{X}}_0)\boldsymbol{\Sigma}^{-1}\left(\sigma(\bar{\mathbf{X}}_{1,f(b)} - \bar{\mathbf{X}}_0)\right)^T}{N_{te}(N_{te}-1)/2} & , N_{te} > 1 \\ \sigma(\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_0)\boldsymbol{\Sigma}^{-1}\left(\sigma(\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_0)\right)^T & , N_{te} = 1 \end{cases} \tag{3}$$

where $\bar{\mathbf{X}}_{1,f(a)}$ and $\bar{\mathbf{X}}_0$ are the means of $\mathbf{X}_{1,f(a)}$ and $\mathbf{X}_0$, respectively. When $N_{te} > 1$, this new distance metric incorporates the inner product between $\bar{\mathbf{X}}_{1,f(a)} - \bar{\mathbf{X}}_0$ and $\bar{\mathbf{X}}_{1,f(b)} - \bar{\mathbf{X}}_0$. As the angle between vectors $\sigma(\bar{\mathbf{X}}_{1,f(a)} - \bar{\mathbf{X}}_0)$ and $\sigma(\bar{\mathbf{X}}_{1,f(b)} - \bar{\mathbf{X}}_0)$ increases, d_B^{f-f} decreases, which further avoids the interference of the "partial errors" situation (i.e., scenarios where both correctly classified and misclassified events coexist under the same hypothesis) compared with the simple inter-class mean squared distance $\sigma(\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_0)\boldsymbol{\Sigma}^{-1}(\sigma(\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_0))^T$ which was used originally in UMM.

2) Sliding Windowing under Hypothesis m

In addition, the inclusion of a sliding window allows for better extraction of localized features in the time domain. Due to the temporal nature of ERPs, the differences between targets and non-targets will not be equally significant at all T time points. By incorporating sliding windows and thus obtaining the distances in each window and highlighting the weights of the windows with the most significant differences, a more reliable estimate of the differences between classes may be obtained. However, in unsupervised methods, we are unable to effectively estimate the window weights using labeled data. Therefore, we propose an automatic weighting method based on the norm sum.

First, after adding the sliding windows, we obtain an inter-class difference metric from each window using equation (1). We set the total number of windows to N_{sw}, and for the *i*-th window, epochs $\mathbf{X}^{(i)}$ and inverted covariance matrices $\{\boldsymbol{\Sigma}^{-1}\}^{(i)}$ are intercepted from the original epochs *X* and the inverted covariance matrix $\boldsymbol{\Sigma}^{-1}$ directly according to their time ranges, as shown in Fig. 1. To ensure that the window slides evenly over each time point, we applied padding operations to the epochs *X* and inverted the covariance matrix $\boldsymbol{\Sigma}^{-1}$ as shown in the figure. Note that, since the inverted covariance matrix $\boldsymbol{\Sigma}^{-1}$ no longer necessarily has a block-Toeplitz structure[12] like $\boldsymbol{\Sigma}$, the matrices $\{\boldsymbol{\Sigma}^{-1}\}^{(i)}$ chosen at different windows may not be identical. In this way, the inter-class difference metric we obtain from the *i*-th window is:

$$D_i(m) = \frac{d_B^{f-f}\left(\mathbf{X}^{(i)}, \{\boldsymbol{\Sigma}^{-1}\}^{(i)}\right)}{d_W(\mathbf{X}^{(i)}, \{\boldsymbol{\Sigma}^{-1}\}^{(i)})} \tag{4}$$

Then, after obtaining all N_{sw} values of $D_i(m) \in \mathbb{R}$, an inter-class difference metric $D(m) \in \mathbb{R}$ must be calculated based on

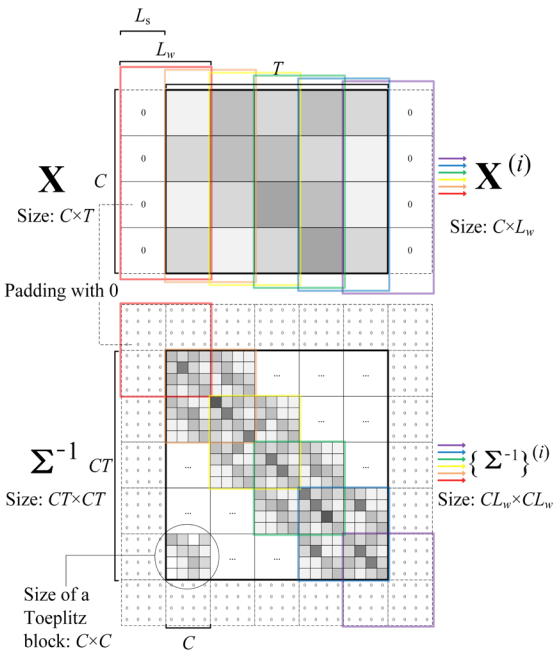


Fig 1. Application of the sliding window setting. L_w represents the window length, L_s represents the sliding step size, and the padding before and after $\mathbf{X}$ corresponds to a time length of $L_w - L_s$.

these values for the final decision:

$$D(m) = \left(\sum_{i=1}^{N_{sw}} \left(D_i(m) \right)^{\gamma} \right)^{1/\gamma} \tag{5}$$

In the above equation, the overall inter-class difference metric is calculated as the $1/\gamma$ power of the sum of the γ powers of the inter-class difference metrics of all windows, i.e., the L_γ-norm of the vector of metrics. This approach replaces the simpler method of directly computing the inter-class difference metric from the full window using equation (1).

As we know, a L_γ-norm of a n-dimensional vector $\mathbf{x} = [x_1, x_2, \dots, x_n]$ is defined as $\|x\|_\gamma = (|x_1|^\gamma + |x_2|^\gamma + \cdots + |x_n|^\gamma)^{1/\gamma}$. In practical applications, common norms include the L_1-norm (Manhattan norm) and the L_2-norm (Euclidean norm). As γ gradually increases, the L_γ-norm will increasingly emphasize the influence of larger elements in the vector. When γ tends to infinity, the L_∞-norm is obtained, representing the maximum absolute value of the vector's elements. In this paper, a reasonable γ value is used, and the L_γ-norm of $[D_1(m), D_2(m), \dots, D_{N_{sw}}(m)]$ is applied instead of simply summing all the elements of it (i.e., its L_1-norm). This choice is made because we aim to highlight the windows with the most significant differences between classes, which are typically represented by periods with the most pronounced ERP amplitudes.

3) Decision-making

After obtaining $D(m)$ for all hypotheses $m = 1,2, \dots, N_s$, we select the hypothesis m_1 that maximizes $D(m)$ as the most representative hypothesis (i.e., $\max D(m) = D(m_1)$), and output the corresponding label as the algorithm's predicted label.

B. Online Update Framework

Utilizing information from past trials is crucial for the algorithm's performance in the online updating framework. We adopt an online updating strategy, which involves estimating class means using pseudo-labels generated from past trials.

1) Online Updates for Class Means, Mean Intra-Class Distances, and Covariance Estimations

Conventionally, both classes of epochs from all trials are assumed to originate from the same distribution. This assumption allows us to leverage classified data from both current and past trials to achieve more accurate estimations of the class means, mean intra-class distances, and covariance. These refined estimates can, in turn, enhance the classification capability of our algorithm. In each hypothesis, updates are performed by incorporating the most recent estimates to refine the inter-class difference metric $D(m)$. Additionally, after each decision, estimates derived from the latest decision are integrated with the existing estimates to obtain an updated overall estimate.

The inter-class difference vector $\Delta\mathbf{X} \in \mathbb{R}^{C\times T}$ is defined as the difference vector between the two class means $\bar{\mathbf{X}}_1$ and $\bar{\mathbf{X}}_0$. Since zero-centering is applied to all input epochs and the ratio of the number of targets to non-targets is known and fixed, the estimation of $\Delta\mathbf{X}$, $\bar{\mathbf{X}}_1$, and $\bar{\mathbf{X}}_0$ are connected by simple linear transformations (e.g., $\bar{\mathbf{X}}_1 = -5\bar{\mathbf{X}}_0$, i.e., $\Delta\mathbf{X} = \bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_0 = \frac{6}{5}\bar{\mathbf{X}}_1 = -6\bar{\mathbf{X}}_0$, when the ratio of the number of targets to non-targets is $S_1/S_0 = 1/5$). We directly discuss the updating process using $\Delta\mathbf{X}$ as a representation of class mean updating.

Let $\widehat{\Delta\mathbf{X}}^n \in \mathbb{R}^{C\times T}$ represents the overall estimate of $\Delta\mathbf{X}$ obtained from all the decisions of the first n trials, and $\Delta\mathbf{X}^n \in \mathbb{R}^{C\times T}$ represents the $\Delta\mathbf{X}$ obtained from the decision of the n-th trial or from one of the hypotheses in the n-th trial. Similarly, we denote the estimates of $\{\bar{\mathbf{X}}_{1,f(a)}, \bar{\mathbf{X}}_1, \bar{\mathbf{X}}_0, d_W\}$ generated from the first n trials as $\{\hat{\mathbf{X}}^n_{1,f(a)}, \hat{\mathbf{X}}^n_1, \hat{\mathbf{X}}^n_0, \hat{d}^n_W\}$, and the estimates generated from the n-th trial as $\{\bar{\mathbf{X}}^n_{1,f(a)}, \bar{\mathbf{X}}^n_1, \bar{\mathbf{X}}^n_0, d^n_W\}$.

In the online updating process, after obtaining the classification result of the n-th trial as a new pseudo-label, $\Delta\mathbf{X}$ will be updated from $\widehat{\Delta\mathbf{X}}^{n-1}$ to $\widehat{\Delta\mathbf{X}}^n$. The inter-class difference vector $\widehat{\Delta\mathbf{X}}^n$ is the weighted vector sum of the inter-class difference vector $\widehat{\Delta\mathbf{X}}^{n-1}$ for all past trial epochs and the inter-class difference vector $\Delta\mathbf{X}^n$ under the current trial epochs. The weight of $\Delta\mathbf{X}^n$ is calculated based on the classification results, representing the confidence in the classification results of the current trial. According to the descending order of the corresponding $D(m)$, for the n-th trial denoted as $D^{\sim n}(m)$, all the possible hypotheses $m = 1,2, \dots, N_s$ are reordered into a new sequence $[m_1, m_2, \dots, m_{N_s}]$ (i.e., $D^{\sim n}(m_1) > D^{\sim n}(m_2) > \cdots > D^{\sim n}(m_{N_s})$). The confidence level of the n-th trial is calculated according to the following formula:

$$w(n) = \min\left(1, \frac{D^{\sim n}(m_1) - D^{\sim n}(m_2)}{std\left(D^{\sim n}(m_2), \dots, D^{\sim n}(m_{N_s})\right)}\right) \tag{6}$$

The term $std(\cdot)$ denotes the standard deviation and the range of

values of $w(n)$ is [0,1]. The update of $\Delta\mathbf{X}$ is shown in equation (7):

$$\widehat{\Delta\mathbf{X}}^n = \begin{cases} \dfrac{(\sum_{j=1}^{n-1} w(j))\widehat{\Delta\mathbf{X}}^{n-1} + w(n)\Delta\mathbf{X}^n}{\sum_{j=1}^{n} w(j)}, & n > 1 \\ \Delta\mathbf{X}^n, & n = 1 \end{cases} \tag{7}$$

Similarly, $\overline{\mathbf{X}}_1$ and $\overline{\mathbf{X}}_0$ will be updated in the same manner.

When $n > 1$, in order to utilize the mean estimates obtained from the previous $n-1$ trials, for the m-th hypothesis, $\overline{\mathbf{X}}_{1,f(a)}$, $\overline{\mathbf{X}}_1$, and $\overline{\mathbf{X}}_0$ will all be updated in the form of equation (7). These represent the mean estimates of $\overline{\mathbf{X}}_{1,f(a)}$, $\overline{\mathbf{X}}_1$, and $\overline{\mathbf{X}}_0$ derived from the first n trials under the given hypothesis, and are then substituted into equations (2) and (3). Since $\mathbf{X}_{1,f(\mathrm{a})}$ is included in $\mathbf{X}_1$, $\overline{\mathbf{X}}_{1,f(a)}$ should be updated in the same way as $\overline{\mathbf{X}}_1$, i.e., use $\widehat{\mathbf{X}}_1^{n-1}$ as the mean from the first $n-1$ trials.

Due to the computational complexity of calculating the mean intra-class distances, it is not feasible to recalculate it on all epochs from past trials one by one according to the latest obtained class means. Therefore, we updated the mean intra-class squared distance $d_W \in \mathbb{R}$ in the same way as the class means, setting $\tilde{d}_W^n$ to be the weighted sum of the mean intra-class squared distance d_W^n under the current trial hypothesis and the mean intra-class squared distance $\tilde{d}_W^{n-1}$ for all past trials, with the confidence mentioned above used as weights.

However, in our proposed method, as distance calculations occur within each window, the aforementioned updated features are also processed according to the time range of the respective window, for example, $\{\widehat{\mathbf{X}}_1^{n-1}, \widehat{\mathbf{X}}_0^{n-1}, \hat{d}_W^{n-1}\}^{(i)}$ is required for the i-th window. What is particularly notable is that $\{\hat{d}_W^{n-1}\}^{(i)}$ needs to be stored separately for each window.

A more precise covariance estimate can also be generated by utilizing all the epochs from all trials. $\mathbf{\Sigma}^n \in \mathbb{R}^{CT\times CT}$ will be calculated directly by pooling all the data from all n trials, rather than using only the data from the n-th trial.

2) Online Procedure of sDDM

The online execution process for the sDDM algorithm is illustrated in Algorithm 1, where N_{trials} denotes the total number of online trials. In all paradigms considered in this study, the total number of candidate symbols N_{s} is set to 36. The class means, mean inter-class squared distance, and mean intra-class squared distance output by the algorithm for the previous trial will be used as inputs for the algorithm in the subsequent trial.

Algorithm 1 Pseudocode for the sDDM

for n **in** [1:N_{trials}] **do**
  Input: Epoch data $\mathbf{X}^n \in \mathbb{R}^{C\times T\times S}$ of the n-th trial, where the total number of epochs is S.
  if n>1 **then**
    Additional input: The confidence level [$w(1)$, $w(2)$, ..., $w(n-1)$] of the previous $n-1$ trials and the features of the previous $n-1$ trials $\{\widehat{\mathbf{X}}_1^{n-1}, \widehat{\mathbf{X}}_0^{n-1}, \hat{d}_W^{n-1}\}$ for full window, which are needed to perform the updates.
  end if
  Estimate the covariance matrix $\mathbf{\Sigma}^n \in \mathbb{R}^{CT\times CT}$ for the n-th trial.
  for i **in** [1:N_{sw}] **do**
    According to the method shown in Fig. 1, $\mathbf{X}^{(i)}$ and $\{\mathbf{\Sigma}^{-1}\}^{(i)}$ of the n-th trial are obtained for the i-th window, as well as $\{\widehat{\mathbf{X}}_1^{n-1}, \widehat{\mathbf{X}}_0^{n-1}, \hat{d}_W^{n-1}\}^{(i)}$ for the i-th window.
    for m **in** [1:N_{s}] **do**
      Assume that the m-th hypothesis correspond to the correct target symbol.
      In the i-th window, estimate $\{\overline{\mathbf{X}}_{1,f(a)}^n, \overline{\mathbf{X}}_1^n, \overline{\mathbf{X}}_0^n, d_W^n\}^{(i)}$ based on the corresponding labels of the m-th hypothesis.
      In the i-th window, according to the updating form of $\Delta\mathbf{X}$ in equation (7), use the features of the n-th trial $\{\overline{\mathbf{X}}_{1,f(a)}^n, \overline{\mathbf{X}}_1^n, \overline{\mathbf{X}}_0^n, d_W^n\}^{(i)}$ and the features of the previous $n-1$ trials $\{\widehat{\mathbf{X}}_1^{n-1}, \widehat{\mathbf{X}}_0^{n-1}, \hat{d}_W^{n-1}\}^{(i)}$, to obtain the estimation features of the previous n trials $\{\widehat{\mathbf{X}}_{1,f(a)}^n, \widehat{\mathbf{X}}_1^n, \widehat{\mathbf{X}}_0^n, \hat{d}_W^n\}^{(i)}$.
      Obtain the inter-class difference metric for the m-th hypothesis in the i-th window $D_i(n,m)$ by equation (4).
    end for
  end for
  for m **in** [1:N_{s}] **do**
    Obtain the inter-class difference metric for the m-th hypothesis $D(n,m)$ by equation (5).
  end for
  Sort all possible hypotheses $m = 1,2,\ldots,N_{\mathrm{s}}$ in the descending order of $[D^{\sim n}(1), D^{\sim n}(2), \ldots, D^{\sim n}(N_{\mathrm{s}})]$ to obtain $[m_1, m_2, \ldots, m_{N_{\mathrm{s}}}]$, i.e., $D^{\sim n}(m_1) > D^{\sim n}(m_2) > \cdots > D^{\sim n}(m_{N_{\mathrm{s}}})$.
  Consider the m_1-th symbol as a result of the decision of the n-th trial.
  The confidence level $w(n)$ is calculated according to equation (6).
  According to the label corresponding to the decision, obtain $\{\widehat{\mathbf{X}}_1^n, \widehat{\mathbf{X}}_0^n, \hat{d}_W^n\}$ for full window by equation (7).
  Output: The labels correspond to m_1, $w(n)$, and $\{\widehat{\mathbf{X}}_1^n, \widehat{\mathbf{X}}_0^n, \hat{d}_W^n\}$ for full window.
end for

C. Spatial Downscaling Method in Online Processes

The distribution of ERP features in EEG channels is characterized by local sparsity. When the channels cover the entire scalp, applying channel selection methods to emphasize ERP-related components in the EEG input is an effective strategy for enhancing classification accuracy. Therefore, we designed an online spatial dimensionality reduction strategy for unsupervised algorithms such as sDDM and UMM. In the initial set of trials, due to the lack of reliable label information, it is challenging to extract effective channel weighting information from

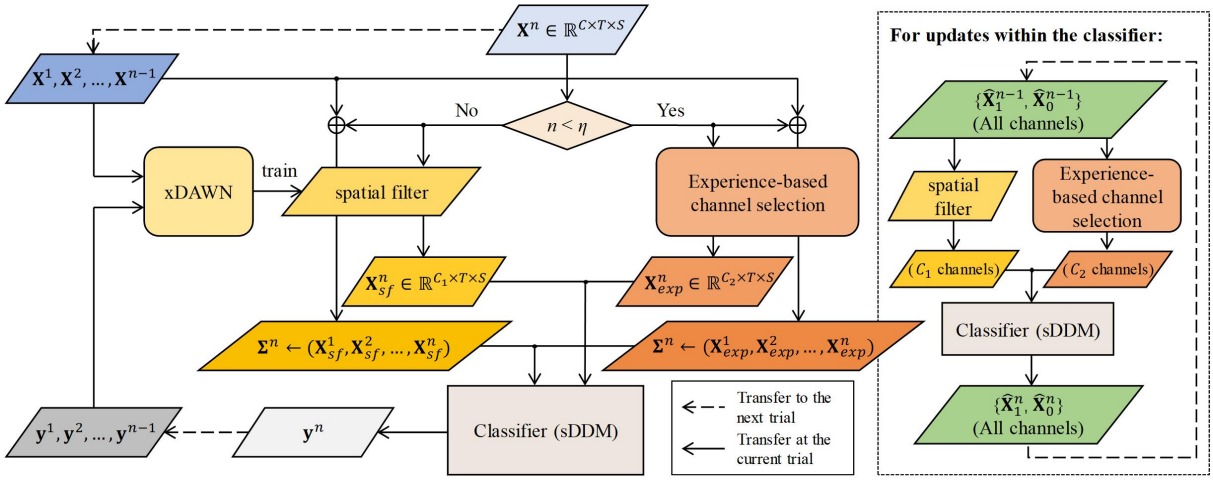


Fig 2. Online spatial dimensionality reduction strategy. $\mathbf{X}_{sf}^n$ and $\mathbf{X}_{exp}^n$ represent the results of $\mathbf{X}^n$ after being processed by the spatial filter and channel selection, respectively. C_1 and C_2 are their corresponding channel dimensions. To maintain the continuity of the class mean updates in the unsupervised algorithm between the two spatial processing methods, we adopted the approach shown in the right-hand inset. Specifically, the class means for all channels are saved after each trial.

the epochs. Combining existing neurophysiological knowledge and past practical experience, manual channel selection can be performed on the initial set of trials during the online classification process to preliminarily improve the signal-to-noise ratio of the input data. Subsequently, after obtaining a sufficient amount of reliably labeled data across all channels, we train xDAWN spatial filters [35] to maximize the signal differences between different categories, thereby enhancing classification performance. Fig. 2 illustrates the online spatial dimensionality reduction strategy employed in this study, with the utilization of xDAWN spatial filters after the η-th trial.

III. Experiments and Results

A. Datasets and Preprocessing

In our experiments, all algorithms were tested on our self-collected dataset, a publicly available dataset recorded during the use of a P300 Speller by patients with ALS [36] and the BCI Competition III Dataset II [37]. These datasets all use the visual BCI protocol based on the P300.

Dataset 1: Our self-collected dataset was originally used to compete in the ERP task of the BCI Brain Control Competition at the 2023 World Robot Contest. This study involved human participants, and the experimental procedures (Document No.: ECUST-2022-054) were approved by the local Institutional Review Board. The dataset consists of EEG recorded from 42 participants (L01-L42), all of whom were tasked with completing random symbol spelling tasks across 32 trials. Each trial comprises 5 iterations of randomly sequenced stimulation events, with the interval between event sequences set to 850 ms. Each sequence contains 12 events with symbols defined by the binomial coefficient [38]. The target events depict a face appearing and disappearing on the target symbol, with each appearance and disappearance lasting 75 ms, resulting in a total event duration of 150 ms, i.e., inter-stimulus interval (ISI) is 75 ms and stimulus onset asynchrony (SOA) is 150 ms. The paradigm interface displays a 6×6 symbol matrix, including letters A-Z, numbers 1-9, and the underscore, totaling 36 spelling targets. EEG data were collected using a wireless EEG acquisition system NSW364 (Neuracle, NeuSen W series) with 59 EEG channels and were sampled at 1000 Hz. They are available at: https://doi.org/10.5281/zenodo.18035041. Then, data were down-sampled to 250 Hz, followed by bandpass filtering in the 0.1-16 Hz range.

Dataset 2: This dataset was recorded during the use of a P300 speller by patients with ALS and follows the classical 6×6 oddball paradigm with row/column stimulus, with an ISI and an SOA of 125 ms and 250 ms, respectively. A total of 8 individuals with ALS (S01-S08, 3 females, average age 58±12 years) participated in the experiment, spelling 7 words of 5 symbols each. Each trial contained 10 iterations of the event sequence with 12 events in each sequence, with data recorded from 8 selected EEG channels. Data were filtered using a Butterworth filter in the 0.1-10 Hz range.

Dataset 3: The BCI Competition III Dataset II includes data from 2 participants (A and B), utilizing the classical Oddball 6×6 row/column stimulus paradigm. Rows and columns stimuli appear randomly for 100 ms, with an ISI of 75 ms and an SOA of 175 ms. A total of 12 events were repeated 15 times in each trial. Each participant's data includes 85 trials in the training phase and 100 trials in the testing phase. In this experiment, the two phases were combined into a single extended online session consisting of 185 trials. All data were recorded using 64 channels and were filtered using a Butterworth filter with a frequency range of 0.5-16 Hz.

After bandpass filtering, all datasets were down-sampled to 20 Hz, and data within 0-800 ms after each stimulus were extracted as epochs. Z-score normalization was applied to each channel.

B. Experimental Results

All experiments in this study are based on a simulated online process, where the process of spelling each symbol is considered to be a trial, and in each trial, a data packet is input into the signal processing module, containing only unlabeled data epochs and the order in which the events are randomly presented. Before the start of the simulated online process, all classification algorithms are provided with the encoding rules of the

paradigm in advance, but do not receive any labels or data. All experiments were conducted using MATLAB 2020b on an HP Laptop with a 13th Gen Intel(R) Core(TM) i9-13900HX CPU @ 2.20GHz, 16 GB RAM, and a 64-bit Windows 11 OS.

We first validated the effectiveness of the spatial dimensionality reduction strategy on Dataset 1. In this study, η was set to 8 for all algorithms, meaning that the first 8 trials utilized the empirical channel selection method. The selected channels included: Fz, Cz, Pz, Oz, P3, P4, PO7, and PO8. We then chose the EM algorithm and the UMM algorithm as benchmarks against which we compared our proposed method. In this study, we independently reproduced the UMM algorithm and utilized the publicly available implementation of the EM algorithm from [32]. Additionally, we selected two control scenarios for comparison: one referred to as "full window", where the proposed method is applied without the sliding window (Control 1), and the other termed "original distance", where only the original inter-class squared distance $\sigma(\overline{\mathbf{X}}_1 - \overline{\mathbf{X}}_0)\boldsymbol{\Sigma}^{-1}(\sigma(\overline{\mathbf{X}}_1 - \overline{\mathbf{X}}_0))^T$ is used (Control 2), to evaluate the effectiveness of the two improvements. Both UMM and our proposed method utilized the means based on equation (7), and the covariance estimation used for calculating the squared Mahalanobis distance was transformed into the block-Toeplitz structure[12].

The evaluation metric employed in this study was spelling accuracy, defined as the proportion of correctly selected symbols among all chosen symbols. This metric provides an estimate of the probability of correctly spelling the intended symbols during the experiment. Since the number of symbols in all paradigms is 36, the expected chance level is 1/36 = 2.78%. Unless otherwise specified, the term "accuracy" in the subsequent text refers to spelling accuracy. Since the variables considered in this study do not necessarily follow a normal distribution, the statistical significance between different variables was assessed using a non-parametric method, the Wilcoxon signed-rank test, to calculate *p*-values. To account for multiple comparisons, the *p*-values were adjusted using the Bonferroni correction. Statistical analyses were performed only on Dataset 1, as the small number of participants in Datasets 2 and 3 does not support statistical testing.

In this section, apart from Dataset 2, which had only 8 channels of data, the proposed algorithm and UMM applied the aforementioned spatial dimensionality reduction strategy on Datasets 1 and 3. Since the online process of Dataset 3 contains 185 trials, only the most recent 30 trials were retained in training the xDAWN spatial filter to avoid consuming too much time in its training. Due to the necessity of inheriting projection vectors from the last trial, the EM algorithm is not suitable for employing this strategy; therefore, we only apply the aforementioned experience-based selected channels with this method for all the datasets. The number of EM-steps used by the EM algorithm is set to 5.

1) Validation of Spatial Downscaling Strategies

Initially, the effectiveness of the xDAWN-based spatial dimensionality reduction strategy was validated on UMM and our proposed method. The results on Dataset 1 are presented in Table I, where "all channels" denotes the utilization of all channels in all trials without any processing, "selected channels" indicates the usage of empirically selected channels, and the suffix "-xDAWN" indicates the adoption of xDAWN spatial filters trained with pseudo labels after the η-th trial.

TABLE I
AVERAGE SIMULATED ONLINE SPELLING ACCURACY (%) USING DIFFERENT SPATIAL PROCESSING ON DATASET 1

Spatial processing	UMM	Proposed
All channels	28.13 (±32.35)	72.92 (±38.74)
Selected channels	86.09 (±20.17)	88.84 (±16.93)
Selected channels-xDAWN	85.57 (±24.86)	92.41 (±14.91)

Spelling accuracies are reported as "mean (±standard deviation)".

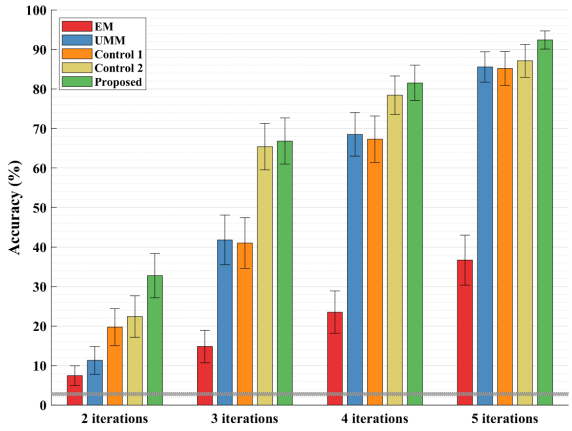


Fig 3. Average simulated online spelling accuracy using the first 2, 3, 4, and 5 iterations on Dataset 1. The error bars indicate the standard error for each method. The expected chance level for this dataset, along with its standard error range, is highlighted in the gray-shaded area (2.78±0.45%).

UMM and our proposed method exhibit significant improvements when applied with "Selected channels-xDAWN" compared to using all channels. When this strategy was used together with our proposed method, its performance was superior to that obtained when combined with UMM, yielding an improvement of approximately 3.6% compared with the "selected channels" condition (p = 0.002), while no significant change was observed for UMM. This improvement primarily arises from the higher reliability of the initial pseudo-labels in our proposed method, which enables the spatial filters to be trained with more accurate labels. Furthermore, the higher reliability of the initial pseudo-labels allows our proposed method to better compensate for the negative impact of using all channels compared with UMM.

2) Comparison Experiment

We conducted a simulated online experiment using data from all 42 participants in Dataset 1 and evaluated the average classification accuracy results produced by each method using the first 2, 3, 4, and 5 iterations of all 5 iterations of event sequences. Considering correction under multiple testing, a total of eight pairwise comparisons were conducted under the same conditions across the five algorithms. Therefore, after Bonferroni correction, the significance threshold for each *p*-value was adjusted to 0.00625. The results are presented in Fig. 3.

Our proposed method exhibited consistently superior performance across all numbers of iterations used. In the first 2 iterations, the superiority of UMM over EM was not statistically significant (first 2 iterations EM-UMM: p = 0.562; first 3, 4, and 5 iterations EM-UMM: $p \leq 0.001$). In contrast, our proposed method consistently outperformed EM in the first 2, 3, 4, and 5

TABLE II
SIMULATED ONLINE SPELLING ACCURACY (%) OF PARTICIPANTS FROM DATASET 1 USING CUMULATIVE DATA FROM ALL 5 ITERATIONS

Participant	EM	UMM	Control 1	Control 2	Proposed
L01	03.13	87.50	00.00	93.75	78.13
L02	00.00	100.00	100.00	100.00	100.00
L03	93.75	96.88	100.00	96.88	100.00
L04	00.00	00.00	75.00	00.00	87.50
L05	00.00	96.88	96.88	100.00	100.00
L06	78.13	100.00	100.00	100.00	100.00
L07	90.63	96.88	90.63	96.88	90.63
L08	00.00	100.00	96.88	100.00	96.88
L09	100.00	87.50	90.63	87.50	93.75
L10	75.00	100.00	93.75	90.63	87.50
L11	00.00	96.88	93.75	93.75	93.75
L12	62.50	96.88	93.75	87.50	84.38
L13	00.00	90.63	03.13	100.00	100.00
L14	00.00	90.63	87.50	87.50	90.63
L15	00.00	71.88	00.00	00.00	75.00
L16	00.00	93.75	93.75	93.75	93.75
L17	00.00	90.63	90.63	87.50	90.63
L18	00.00	78.13	03.13	06.25	06.25
L19	81.25	96.88	96.88	90.63	96.88
L20	00.00	15.63	93.75	90.63	87.50
L21	93.75	100.00	96.88	100.00	100.00
L22	00.00	93.75	100.00	100.00	100.00
L23	00.00	00.00	87.50	93.75	100.00
L24	81.25	46.88	71.88	93.75	87.50
L25	46.88	100.00	100.00	100.00	100.00
L26	93.75	100.00	100.00	100.00	100.00
L27	53.13	90.63	93.75	15.63	90.63
L28	00.00	90.63	96.88	93.75	93.75
L29	00.00	96.88	96.88	100.00	96.88
L30	87.50	96.88	96.88	100.00	100.00
L31	68.75	100.00	100.00	96.88	96.88
L32	00.00	100.00	100.00	100.00	100.00
L33	93.75	100.00	93.75	100.00	100.00
L34	00.00	65.63	87.50	100.00	100.00
L35	87.50	100.00	100.00	100.00	100.00
L36	90.63	100.00	100.00	100.00	100.00
L37	00.00	81.25	87.50	93.75	93.75
L38	84.38	90.63	90.63	90.63	87.50
L39	68.75	96.88	100.00	96.88	100.00
L40	00.00	78.13	81.25	84.38	84.38
L41	00.00	100.00	100.00	100.00	100.00
L42	06.25	78.13	96.88	96.88	96.88
Mean (±STD)	36.68 (±41.08)	85.57 (±24.86)	85.19 (±27.88)	87.13 (±26.94)	**92.41** (±14.91)

STD: standard deviation. Average spelling accuracies below 50% are highlighted with underscores.

TABLE III
AVERAGE SIMULATED ONLINE SPELLING ACCURACY (%) AND SIGNIFICANCE TEST RESULTS ON DATASET 1 USING CUMULATIVE DATA OF ALL 5 ITERATIONS, AFTER EXCLUDING RESULTS NOT EXCEEDING THE CHANCE LEVEL

Method	EM	UMM	Control 1	Control 2	Proposed
No. of participants	21	40	40	40	42
Mean (±STD)	73.36 (±26.78)	89.30 (±16.49)	89.46 (±21.13)	91.49 (±19.32)	**92.41** (±14.91)
p (-EM)	/	0.025	**0.001**	**0.002**	**< 0.001**
p (-UMM)	/	/	0.970	0.195	0.496
p (-Control 1)	/	/	/	0.399	0.112
p (-Control 2)	/	/	/	/	0.943

STD: standard deviation.

iterations (all numbers of iterations EM-Proposed: $p < 0.001$), and achieved statistically significant improvements over UMM in the first 2 and 3 iterations (first 2 iterations UMM-Proposed: $p = 0.001$; first 3 iterations UMM-Proposed: $p = 0.001$; first 4 iterations UMM-Proposed: $p = 0.012$; first 5 iterations UMM-Proposed: $p = 0.247$). In the initial 5 iterations, our proposed method achieved the highest average accuracy, exceeding 92%.

Throughout all numbers of iterations used, Control 2 showed significantly higher accuracy than UMM and Control 1 only in some cases (first 3 iterations UMM-Control 2: $p = 0.001$; first 3 iterations Control 1-Control 2: $p = 0.001$). No significant differences were observed between UMM, Control 1, and Control 2 beyond these cases. Additionally, our proposed method demonstrated significantly higher accuracy than Control 1 in the first 3 and 4 iterations (first 3 iterations Control 1-Proposed: $p < 0.001$; first 4 iterations Control 1-Proposed: $p = 0.002$).

Table II presents the simulated online accuracy performance from all 42 participants in Dataset 1 over all 5 iterations of experiments, where data highlighted with an underscore indicate accuracies below 50% for that participant when using the respective method.

Our proposed method achieved the highest average accuracy among all compared methods. In the results from the EM algorithm, more than half of the participants (24 out of 42) had accuracies below 50%, indicating its ineffectiveness in classifying ERP tasks on this dataset. UMM, Control 1, and Control 2 each had 4 underperforming participants, while our proposed method only had 1 such participant. We can observe, from Table II, that the majority of highlighted accuracies tend towards 0, indicating a strong negative correlation between the number of underperforming participants and the average accuracy. This phenomenon is attributed to the issue of pseudo-label contamination (discussed later), which remains the primary obstacle to further enhancing the performance of our proposed method.

For reference, we recalculated the average accuracies and Wilcoxon signed-rank test results for each method after excluding those below the chance level, as shown in Table III. The proposed method still achieved the highest mean accuracy, confirming that the improvement remains valid under normal conditions. Meanwhile, the performance gaps between methods became smaller, and the pairwise differences among the last four methods were no longer statistically significant, suggesting that the superior robustness of the proposed method partly stems from its ability to avoid pseudo-label contamination.

Additionally, we conducted simulated online experiments on the data from all 8 ALS patients from Dataset 2, using the first 5, 7, and 10 iterations out of all 10 iterations of event sequences. The average accuracy results for the first 5, 7, and 10 iterations are illustrated in Fig. 4, and individual participants' average classification accuracies for the first 10 iterations are presented in Table IV.

In the first 5 iterations, Control 1 achieved the highest mean accuracy, while UMM demonstrated the lowest value. Our proposed method showed the best performance in the first 7 and 10 iterations. When considering the first 10 iterations from the perspective of individual participants, our proposed method had only 1 underperforming participant, the same as Control 2, and achieved the best classification accuracy in 5 out of 8 participants. Control 1, UMM, and EM had 2, 2, and 4 underperforming participants, respectively.

Finally, we conducted a simulated online experiment comprising 185 trials from Dataset 3. The results are illustrated in Fig. 5 and Table V. As shown in Table V, neither participant A

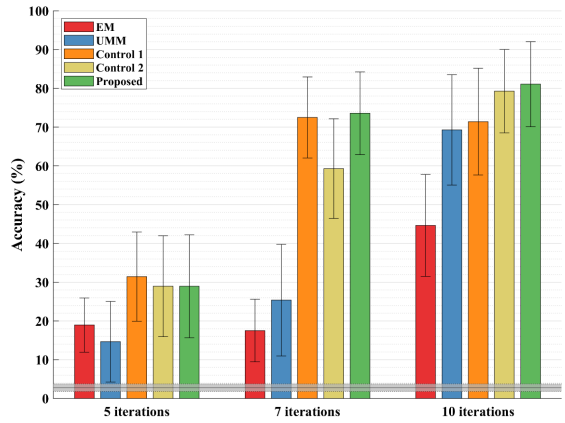


Fig 4. Average simulated online spelling accuracy using the first 5, 7, and 10 iterations from Dataset 2. The error bars indicate the standard error for each method. The expected chance level for this dataset, along with its standard error range, is highlighted in the gray-shaded area (2.78±0.98%).

TABLE IV
SIMULATED ONLINE SPELLING ACCURACY (%) OF ALL 8 PARTICIPANTS FROM DATASET 2 USING CUMULATIVE DATA FROM ALL 10 ITERATIONS

Participant	EM	UMM	Control 1	Control 2	Proposed
S01	80.00	82.86	00.00	82.86	85.71
S02	77.14	94.29	94.29	91.43	91.43
S03	91.43	91.43	91.43	85.71	85.71
S04	00.00	00.00	08.57	00.00	00.00
S05	00.00	00.00	88.57	85.71	91.43
S06	68.57	91.43	94.29	94.29	97.14
S07	40.00	94.29	94.29	94.29	97.14
S08	00.00	100.00	100.00	100.00	100.00
Mean	44.64	69.29	71.43	79.29	**81.07**
(±STD)	(±37.20)	(±40.25)	(±38.94)	(±30.43)	(±31.03)

STD: standard deviation. Average spelling accuracies below 50% are highlighted with underscore.

nor participant B exhibited an accuracy below 50% for the first 15 iterations. Under this condition, where pseudo-label contamination is absent, our proposed method achieved the highest average accuracy, particularly demonstrating the best average accuracy for participant A, who displayed lower overall accuracy.

3) Time Cost of Simulated Online Execution

Online BCI execution demands algorithms to promptly return decision results, thus providing closed-loop feedback for users. Therefore, the time cost of algorithms used within BCI systems needs to be within a certain maximum time, typically considered as the time required for a sequence in the paradigm. We present the average time costs of the different algorithms we evaluate in this study and their corresponding maximum time allowed on Datasets 1 and 2 in Table VI. All results are calculated using the maximum number of iterations. The time costs of these algorithms are significantly lower than the maximum allowable time in the paradigm, indicating their compliance with the time requirements for online execution. However, our proposed method does take considerably longer to complete than all the other methods.

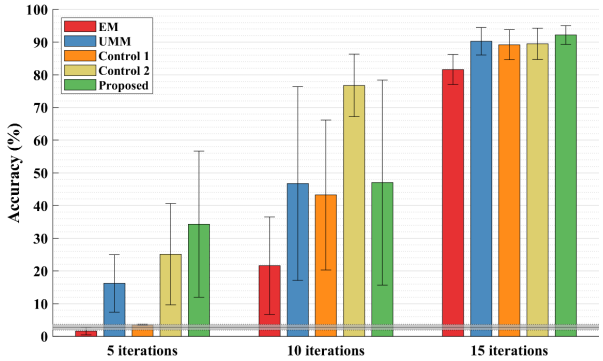


Fig 5. Average simulated online spelling accuracy using the first 5, 10, and 15 iterations on Dataset 3. The error bars indicate the standard error for each method. The expected chance level for this dataset, along with its standard error range, is highlighted in the gray-shaded area (2.78±0.85%).

TABLE V
SIMULATED ONLINE SPELLING ACCURACY (%) OF BOTH PARTICIPANTS FROM DATASET 3 USING CUMULATIVE DATA FROM ALL 15 ITERATIONS

Participant	EM	UMM	Control 1	Control 2	Proposed
A	75.14	84.32	82.70	82.70	88.11
B	88.11	96.22	95.68	96.22	96.22
Mean	81.62	90.27	89.19	89.46	**92.16**

Average spelling accuracies below 50% are highlighted with underscore.

IV. DISCUSSION

A. Learning Curves

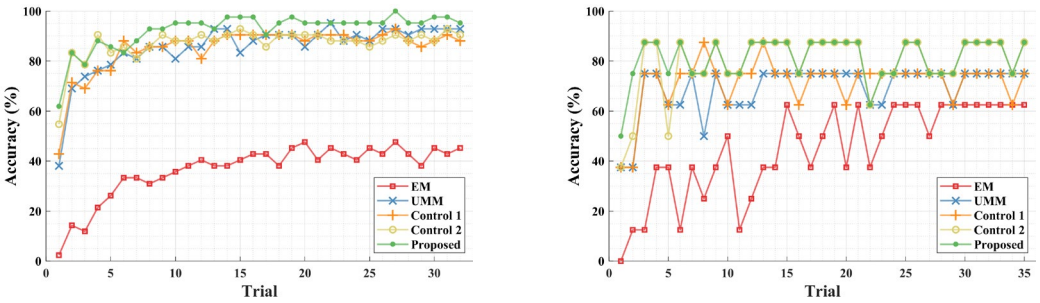
Although unsupervised algorithms eliminate the need for a calibration phase, in practical online scenarios, these algorithms typically require several trials to gradually improve classification to satisfactory levels. Therefore, the learning speed of unsupervised algorithms is also an important metric. Fig. 6 illustrates the learning curves of the various unsupervised methods we consider in this study on Datasets 1 and 2.

As depicted in Fig. 6 (a), both UMM and our proposed method exhibit relatively rapid learning speeds. Our proposed method achieves over 60% initial accuracy as early as the first trial and maintains accuracy above 90% after the eighth trial. In contrast, the EM method consistently performs at a lower accuracy level throughout the entire process, failing to surpass 50% accuracy and falling short of approaching the performance of UMM and our proposed method. Similarly, in Fig. 6 (b), both UMM and our proposed method demonstrate advantages over EM, despite the larger amount of data per trial favoring EM's performance. This result indicates that methods based on the UMM architecture (i.e., UMM and our proposed method) demonstrate faster learning speeds, enabling them to promptly provide reliable classification results to users in practical online BCI scenarios. The improvement in performance of our proposed method over UMM is evident in the two control scenarios

TABLE VI
TIME COST AND THE MAXIMUM TIME ALLOWED ON EACH DATASET

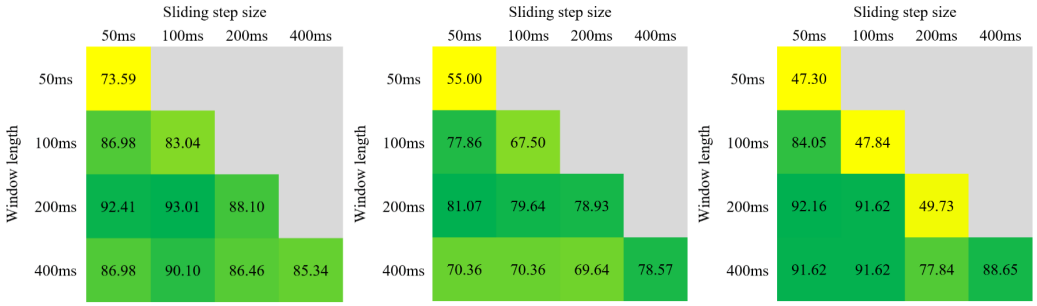
Dataset	Average time cost (s)					Max time (s)
	EM	UMM	Control 1	Control 2	Proposed	
Dataset 1	0.0242 (±0.0119)	0.0240 (±0.0104)	0.0320 (±0.0101)	0.0395 (±0.0136)	0.1020 (±0.0144)	1.8000
Dataset 2	0.0439 (±0.0220)	0.0095 (±0.0014)	0.0282 (±0.0023)	0.0225 (±0.0015)	0.1331 (±0.0019)	3.0000

Results are reported as "mean (±standard deviation)".



(a) Dataset 1 (b) Dataset 2

Fig 6. Learning curves of different methods on Dataset 1 and 2. (a) Accuracy of 32 online trials for all 5 iterations used on Dataset 1. (b) Accuracy of 35 online trials for all 10 iterations used on Dataset 2.



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

Fig 7. Grid search results for structural parameters (window length and sliding step size). (a) Average accuracy (%) on Dataset 1. (b) Average accuracy (%) on Dataset 2. (c) Average accuracy (%) on Dataset 3.

TABLE VII
AVERAGE ACCURACY (%) OF THE PROPOSED METHOD USING DIFFERENT ORDERS OF L_γ-NORM ON DATASETS 1, 2, AND 3

γ	1	2	3	4	5
Dataset 1	75.22	86.38	**92.41**	88.91	91.22
Dataset 2	70.36	80.71	**81.07**	78.93	78.57
Dataset 3	83.78	91.08	**92.16**	90.54	90.54

we tested. The inclusion of a sliding window enhances the classifier's temporal resolution, improving information extraction capabilities for individual trials, and particularly enhancing classification performance in the initial trials (Datasets 1 and 2, Control 2, and our proposed method compared to UMM and Control 1). While the improvement in the inter-class divergence metric alone does not yield significant enhancements, its concurrent application with the sliding window strengthens the classifier's ability to discern erroneous results in later trials (Dataset 1, Control 2 compared to our proposed method).

B. *Selection of Structural Parameters*

Two crucial structural parameters, namely window length and sliding step size, greatly influence the algorithm's ability to extract local features in the time domain. Therefore, we conducted a grid search on these two parameters to investigate their relationship with algorithm accuracy across the datasets. The window length was selected from [50 ms, 100 ms, 200 ms, and 400 ms], and the sliding step size was selected from [50 ms, 100 ms, 200 ms, and 400 ms], while ensuring that the sliding step size did not exceed the window length. We selected the structural parameter combination for the sliding window based on two considerations: the 20Hz down-sampling rate and the 0-800ms time range constraint. Additionally, we aimed to ensure each time point is included in an equal number of windows, which requires that the window length be an integer multiple of the sliding step size.

As depicted in Fig. 7, our proposed algorithm achieved its highest accuracy on Datasets 2 (81.07%) and 3 (92.16%) when the window length was 200 ms and the sliding step size was 50 ms. Similarly, on Dataset 1, it attained the second-highest accuracy (92.41%), with only a slight difference of 0.6% compared to the highest accuracy (93.01%, obtained with a window length of 200 ms and a sliding step size of 100 ms). Additionally, we observed that, when the window length was consistent, a sliding step size of 50 ms generally outperformed a sliding step size of 100 ms. Considering these observations, we selected a window length of 200 ms and a sliding step size of 50 ms for our algorithm.

We next address the selection of γ. In Table VII, we discuss the effects of different values of γ on the datasets to select the L_γ-norm most widely applicable to our proposed method. It can be observed that when γ is set to 3, our proposed method achieves the highest accuracy over all the datasets.

It should be noted that the aforementioned parameters were still selected and validated using Datasets 1, 2, and 3. Although relatively consistent results were obtained, the risk of overfitting cannot be completely ruled out.

Finally, we evaluated the parameter η on our proposed unsupervised spatial dimensionality reduction strategy. This parameter represents the number of trials required to collect initial pseudo-label data before applying xDAWN spatial filtering. As shown in Fig. 8, the learning curves for our proposed method on Dataset 1 under different η values are depicted, with the black curve serving as a reference for the case where xDAWN is not used (w/o xDAWN). It can be observed that when $\eta < 3$, the spatial filters trained with pseudo-label data are unreliable. For $\eta \geqslant 3$, the reliability of the spatial filters improves as the pseudo-label accuracy and data volume increase, resulting in

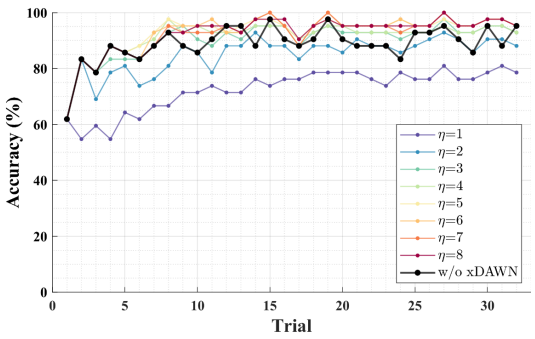


Fig 8. Learning curves achieved with our proposed method on Dataset 1 with different η values. The black curve serves as a reference for the case without xDAWN (w/o xDAWN), where 8 pre-selected channels were used.

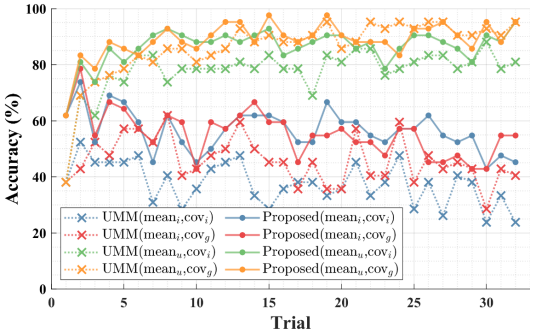


Fig 9. Average spelling accuracy curves achieved with different versions of our proposed method and UMM on Dataset 1. Instantaneous/updated mean (mean_i/mean_u) and instantaneous/global covariance (cov_i/cov_g) are used in pairs. All methods in this comparison were implemented without the spatial dimensionality reduction framework.

the overall performance of the method using xDAWN gradually surpassing that of the w/o xDAWN reference case. Finally, when $\eta \geq 5$, the trials employing xDAWN filters consistently outperform w/o xDAWN in most cases.

C. Research on Instantaneous Classification

Section II.A. describes the instantaneous version of the proposed method. Due to the nonstationary nature of ERP EEG features, this version may exhibit certain robustness in handling such variability. Fig. 9 illustrates the average spelling accuracy curves of our proposed method and UMM on Dataset 1 under different conditions: instantaneous/updated mean (mean_i/mean_u) and instantaneous/global covariance (cov_i/cov_g).

As analyzed above, regardless of the variant used, our proposed method consistently shows noticeably better performance than UMM in the first trial of the learning curve. The use of mean_i does not demonstrate an ability to overcome feature nonstationarity. Instead, the accuracy curves show a gradual decline, likely reflecting the degradation of features caused by participant fatigue and lapses in attention. In contrast, employing mean_u enables the classifier to progressively obtain robust and generalized estimates of ERP features from the participants. This benefit appears to outweigh the impact of nonstationary ERP features, at least within a single online session. Furthermore, the use of mean_u allows the classifiers to continue learning current features to some extent, aiding their adaptation to feature changes over time.

D. Limitations and Future Works

Although the overall classification performance of our proposed method surpassed other unsupervised methods, there are still some challenging limitations.

When training with pseudo-labels (labels predicted by the model), the unreliability or inaccuracy of these labels can lead the model to learn incorrect patterns, ultimately causing the classifier to fail to learn useful features or information, thereby losing its classification ability, which is the problem of pseudo-label contamination. As no real labels are used, any early erroneous classification results can severely affect the reliability of the classifier, even leading to a vicious cycle of decreasing reliability of pseudo-labels and declining classifier performance, resulting in classifier degradation. This phenomenon can be observed in Table IV: the classification accuracy of participant S04 in Dataset 2 remains consistently zero, significantly lower than the method's average classification accuracy, and even lower than the expected accuracy when randomly selecting labels. This issue has also been addressed by Sosulski et al. [34]. In response, they attempted to use confidence as a predictor of this degradation. They found that when using confidence-based mean estimation in UMM, there was a noticeable difference in cumulative confidence between degraded and non-degraded states. This initial insight inspired us to explore a simple correction mechanism based on confidence, aiming to identify and rectify classifiers affected by pseudo-label contamination by setting a threshold. However, the question of how to address pseudo-label contamination and overcome the barriers that restrict the further improvement of fully unsupervised methods in ERP-BCI systems remains to be further investigated in future research.

We discovered that the UMM framework itself does not perform any feature extraction or dimensionality reduction on the channel dimension. Therefore, in the early signal processing steps, we employed 8 fixed channels selected based on empirical knowledge, which was applied to all participants from all of our evaluation datasets. However, empirical channel selection is an artificially chosen method based on neurophysiological knowledge and empirical summaries, which cannot represent universally optimal channel selection across participants nor can it be used to design spatial filters specific to each participant to reflect participant-specific spatial information. We anticipate that blind source separation and other methods can be further employed in the future to improve the algorithm's accuracy.

Furthermore, future research is needed to explore how to apply early stopping [38], [39], [40] to our proposed method, how to measure its information transfer rate (ITR), and further considerations need to be made of the practical online usage of our proposed method.

V. Conclusion

In this work, we have introduced a classification method, sDDM, for performing fully unsupervised online feature extraction/selection. The method enhances the extraction of time-varying information from the EEG by incorporating a sliding window during traversal across different model assumptions. It

employs a novel metric to measure the inter-class distribution difference between target and non-target classes in Mahalanobis space and integrates event similarity within the target class to further eliminate erroneous classification hypotheses. Additionally, the method ensures spatial feature extraction capability by applying a spatial dimensionality reduction strategy combining empirical channel selection and xDAWN. To assess its effectiveness, we conducted comparative experiments with the latest unsupervised classification methods, EM and UMM, as well as two ablation scenarios with our proposed method, using data from a combination of a total of 52 healthy individuals and patients. Our proposed method achieves the highest classification accuracies over all datasets, obtaining spelling accuracies of over 92% for healthy individuals and 81% for patients, demonstrating its broad applicability.

REFERENCES

[1] S. Gao *et al.*, "Visual and Auditory Brain–Computer Interfaces," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 5, pp. 1436–1447, May 2014, doi: 10.1109/TBME.2014.2300164.

[2] H. Zhang *et al.*, "Asynchronous P300-Based Brain–Computer Interfaces: A Computational Approach With Statistical Models," *IEEE Trans. Biomed. Eng.*, vol. 55, no. 6, pp. 1754–1763, Jun. 2008, doi: 10.1109/TBME.2008.919128.

[3] X. Xiao *et al.*, "Discriminative Canonical Pattern Matching for Single-Trial Classification of ERP Components," *IEEE Trans. Biomed. Eng.*, vol. 67, no. 8, pp. 2266–2275, Aug. 2020, doi: 10.1109/TBME.2019.2958641.

[4] A. Rezeika *et al.*, "Brain–Computer Interface Spellers: A Review," *Brain Sci.*, vol. 8, no. 4, Art. no. 4, Apr. 2018, doi: 10.3390/brainsci8040057.

[5] A. Kübler, "The history of BCI: From a vision for the future to real support for personhood in people with locked-in syndrome," *Neuroethics*, vol. 13, no. 2, pp. 163–180, Jul. 2020, doi: 10.1007/s12152-019-09409-4.

[6] J. Jin *et al.*, "Developing a Novel Tactile P300 Brain-Computer Interface With a Cheeks-Stim Paradigm," *IEEE Trans. Biomed. Eng.*, vol. 67, no. 9, pp. 2585–2593, Sep. 2020, doi: 10.1109/TBME.2020.2965178.

[7] L. A. Farwell and E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalogr. Clin. Neurophysiol.*, vol. 70, no. 6, pp. 510–523, Dec. 1988, doi: 10.1016/0013-4694(88)90149-6.

[8] S. Aggarwal and N. Chugh, "Review of Machine Learning Techniques for EEG Based Brain Computer Interface," *Arch. Comput. Method Eng.*, vol. 29, no. 5, pp. 3001–3020, Aug. 2022, doi: 10.1007/s11831-021-09684-6.

[9] D. J. Krusienski *et al.*, "Toward enhanced P300 speller performance," *J. Neurosci. Methods*, vol. 167, no. 1, pp. 15–21, Jan. 2008, doi: 10.1016/j.jneumeth.2007.07.017.

[10] U. Hoffmann *et al.*, "An efficient P300-based brain–computer interface for disabled subjects," *J. Neurosci. Methods*, vol. 167, no. 1, pp. 115–125, Jan. 2008, doi: 10.1016/j.jneumeth.2007.03.005.

[11] B. Blankertz *et al.*, "Single-trial analysis and classification of ERP components — A tutorial," *Neuroimage*, vol. 56, no. 2, pp. 814–825, May 2011, doi: 10.1016/j.neuroimage.2010.06.048.

[12] J. Sosulski and M. Tangermann, "Introducing block-Toeplitz covariance matrices to remaster linear discriminant analysis for event-related potential brain–computer interfaces," *J. Neural Eng.*, vol. 19, no. 6, p. 066001, Dec. 2022, doi: 10.1088/1741-2552/ac9c98.

[13] S. Kundu and S. Ari, "MsCNN: A Deep Learning Framework for P300-Based Brain–Computer Interface Speller," *IEEE Trans. Med. Robotics Bionics*, vol. 2, no. 1, pp. 86–93, Feb. 2020, doi: 10.1109/TMRB.2019.2959559.

[14] S. Kundu and S. Ari, "P300 based character recognition using sparse autoencoder with ensemble of SVMs," *Biocybern. Biomed. Eng.*, vol. 39, no. 4, pp. 956–966, Oct. 2019, doi: 10.1016/j.bbe.2019.08.001.

[15] S. Kundu and S. Ari, "P300 based character recognition using convolutional neural network and support vector machine," *Biomed. Signal Process. Control*, vol. 55, p. 101645, Jan. 2020, doi: 10.1016/j.bspc.2019.101645.

[16] V. J. Lawhern *et al.*, "EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces," *J. Neural Eng.*, vol. 15, no. 5, p. 056013, Jul. 2018, doi: 10.1088/1741-2552/aace8c.

[17] R. T. Schirrmeister *et al.*, "Deep learning with convolutional neural networks for EEG decoding and visualization," *Hum. Brain Mapp.*, vol. 38, no. 11, pp. 5391–5420, 2017, doi: 10.1002/hbm.23730.

[18] E. Santamaría-Vázquez *et al.*, "EEG-Inception: A Novel Deep Convolutional Neural Network for Assistive ERP-Based Brain-Computer Interfaces," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 12, pp. 2773–2782, Dec. 2020, doi: 10.1109/TNSRE.2020.3048106.

[19] Z. Wang *et al.*, "ST-CapsNet: Linking Spatial and Temporal Attention With Capsule Network for P300 Detection Improvement," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 31, pp. 991–1000, 2023, doi: 10.1109/TNSRE.2023.3237319.

[20] J. Jin *et al.*, "Whether generic model works for rapid ERP-based BCI calibration," *J. Neurosci. Methods*, vol. 212, no. 1, pp. 94–99, Jan. 2013, doi: 10.1016/j.jneumeth.2012.09.020.

[21] Y. Zhao *et al.*, "A transplantation of subject-independent model in cross-platform BCI," *Int. J. Mach. Learn. & Cyber.*, vol. 9, no. 6, pp. 959–967, Jun. 2018, doi: 10.1007/s13042-016-0620-1.

[22] J. Jin *et al.*, "The Study of Generic Model Set for Reducing Calibration Time in P300-Based Brain–Computer Interface," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 1, pp. 3–12, Jan. 2020, doi: 10.1109/TNSRE.2019.2956488.

[23] W. Gao *et al.*, "Eliminating or Shortening the Calibration for a P300 Brain–Computer Interface Based on a Convolutional Neural Network and Big Electroencephalography Data: An Online Study," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 31, pp. 1754–1763, 2023, doi: 10.1109/TNSRE.2023.3259991.

[24] Y. Li *et al.*, "A self-training semi-supervised SVM algorithm and its application in an EEG-based brain computer interface speller system," *Pattern Recognit. Lett.*, vol. 29, no. 9, pp. 1285–1294, Jul. 2008, doi: 10.1016/j.patrec.2008.01.030.

[25] J. Long *et al.*, "Semi-supervised joint spatio-temporal feature selection for P300-based BCI speller," *Cogn. Neurodynamics*, vol. 5, no. 4, pp. 387–398, Nov. 2011, doi: 10.1007/s11571-011-9167-8.

[26] D. Wu, "Active semi-supervised transfer learning (ASTL) for offline BCI calibration," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2017, pp. 246–251. doi: 10.1109/SMC.2017.8122610.

[27] M. Ogino *et al.*, "Semi-Supervised Learning for Auditory Event-Related Potential-Based Brain–Computer Interface," *IEEE Access*, vol. 9, pp. 47008–47023, 2021, doi: 10.1109/ACCESS.2021.3067337.

[28] J. Li *et al.*, "A novel semi-supervised meta learning method for subject-transfer brain–computer interface," *Neural Netw.*, vol. 163, pp. 195–204, Jun. 2023, doi: 10.1016/j.neunet.2023.03.039.

[29] P.-J. Kindermans *et al.*, "A Bayesian Model for Exploiting Application Constraints to Enable Unsupervised Training of a P300-based BCI," *PLOS ONE*, vol. 7, no. 4, p. e33758, Apr. 2012, doi: 10.1371/journal.pone.0033758.

[30] D. Hubner *et al.*, "Unsupervised Learning for Brain-Computer Interfaces Based on Event-Related Potentials: Review and Online Comparison [Research Frontier]," *IEEE Comput. Intell. Mag.*, vol. 13, no. 2, pp. 66–77, May 2018, doi: 10.1109/MCI.2018.2807039.

[31] D. Hübner *et al.*, "Learning from label proportions in brain-computer interfaces: Online unsupervised learning with guarantees," *PLOS ONE*, vol. 12, no. 4, p. e0175856, Apr. 2017, doi: 10.1371/journal.pone.0175856.

[32] T. Verhoeven *et al.*, "Improving zero-training brain-computer interfaces by mixing model estimators," *J. Neural Eng.*, vol. 14, no. 3, p. 036021, Jun. 2017, doi: 10.1088/1741-2552/aa6639.

[33] J. Jin *et al.*, "An optimized ERP brain–computer interface based on facial expression changes," *J. Neural Eng.*, vol. 11, no. 3, p. 036004, Apr. 2014, doi: 10.1088/1741-2560/11/3/036004.

[34] J. Sosulski and M. Tangermann, "UMM: Unsupervised Mean-difference Maximization," *arXiv*, Jun. 2023, doi: 10.48550/arXiv.2306.11830.

[35] B. Rivet* *et al.*, "xDAWN Algorithm to Enhance Evoked Potentials: Application to Brain–Computer Interface," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 8, pp. 2035–2043, Aug. 2009, doi: 10.1109/TBME.2009.2012869.

[36] A. Riccio *et al.*, "Attention and P300-based BCI performance in people with amyotrophic lateral sclerosis," *Front. Hum. Neurosci.*, vol. 7, p. 732, 2013.

[37] B. Blankertz *et al.*, "The BCI competition III: validating alternative approaches to actual BCI problems," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 14, no. 2, pp. 153–159, Jun. 2006, doi: 10.1109/TNSRE.2006.875642.

[38] J. Jin *et al.*, "Optimized stimulus presentation patterns for an event-related potential EEG-based brain–computer interface," *Med. Biol. Eng. Comput.*, vol. 49, no. 2, pp. 181–191, Feb. 2011, doi: 10.1007/s11517-010-0689-8.

[39] M. Schreuder *et al.*, "Performance optimization of ERP-based BCIs using dynamic stopping," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, 2011, pp. 4580–4583.

[40] M. Schreuder *et al.*, "Optimizing event-related potential based brain–computer interfaces: a systematic evaluation of dynamic stopping methods," *J. Neural Eng.*, vol. 10, no. 3, p. 036025, May 2013, doi: 10.1088/1741-2560/10/3/036025.