

# SincNet-Based Hybrid Neural Network for Motor Imagery EEG Decoding

Chang Liu<sup>1</sup>, Jing Jin<sup>1</sup>, *Senior Member, IEEE*, Ian Daly<sup>2</sup>, Shurui Li<sup>1</sup>, Hao Sun,  
Yitao Huang, Xingyu Wang, and Andrzej Cichocki, *Life Fellow, IEEE*

**Abstract**—It is difficult to identify optimal cut-off frequencies for filters used with the common spatial pattern (CSP) method in motor imagery (MI)-based brain-computer interfaces (BCIs). Most current studies choose filter cut-frequencies based on experience or intuition, resulting in sub-optimal use of MI-related spectral information in the electroencephalography (EEG). To improve information utilization, we propose a SincNet-based hybrid neural network (SHNN) for MI-based BCIs. First, raw EEG is segmented into different time windows and mapped into the CSP feature space. Then, SincNets are used as filter bank band-pass filters to automatically filter the data. Next, we used squeeze-and-excitation modules to learn a sparse representation of the filtered data. The resulting sparse data were fed into convolutional neural networks to learn deep feature representations. Finally, these deep features were fed into a gated recurrent unit module to seek sequential relations, and a fully connected layer was used for classification. We used the BCI competition IV datasets 2a and 2b to verify the effectiveness of our SHNN method. The mean classification accuracies ( $\kappa$  values) of our SHNN method are 0.7426 (0.6648) on dataset 2a and 0.8349 (0.6697) on dataset 2b,

respectively. The statistical test results demonstrate that our SHNN can significantly outperform other state-of-the-art methods on these datasets.

**Index Terms**—Brain-computer interface, motor imagery, SincNet, neural network, gated recurrent unit.

## I. INTRODUCTION

**B**RAIN-COMPUTER interfaces (BCIs) that can directly connect the outside world and the human brain without the involvement of peripheral nerves and muscles [1], have been attracting more and more attention in recent years. Practical applications of BCIs include virtual reality games [2], exoskeleton control [3], [4], rehabilitation of stroke patients [5], [6], communication with patients with consciousness disorder [7], [8], and many other applications.

Motor imagery (MI) based BCIs allow their users to enact control by imagining movement in one or more body part [9]–[11]. The electroencephalogram (EEG) is one of the most commonly used signals in MI-based BCIs and has the advantages of low cost, low risk, and high portability. Movement imagination is associated with changes in the power of the oscillatory EEG. These changes in power are known as the event-related desynchronization (ERD) and event-related synchronization (ERS) phenomena. By detecting the ERD/ERS in the EEG, the part of the body that the user is imagining moving can be identified. Compared with the other common non-invasive BCI control paradigms such as even-related potential (ERP) [12], [13] and steady-state visual evoked potential (SSVEP) [14], [15] based BCIs, the MI-based BCI does not rely on external stimuli, which can make them more convenient and intuitive for their users. However, improving the classification performance of MI-based BCIs is still a challenging problem.

Combinations of spatial, spectral, and temporal features are usually extracted from EEG for MI task classification. The common spatial pattern (CSP) method is the most commonly used spatial feature extraction method used for this process. However, the performance of CSP is easily affected by the selection of the cut-off frequencies that are used to filter the EEG [16], [17]. To address this problem, Ang *et al.* [18] proposed a filter-band CSP (FBCSP) method to filter the EEG into multiple sub-bands. However, these sub-bands also need to be determined manually, while the appropriate filter

Manuscript received December 21, 2021; revised February 23, 2022; accepted February 28, 2022. Date of publication March 2, 2022; date of current version March 11, 2022. This work was supported in part by the National Key Research and Development Program under Grant 2017YFB13003002, in part by the Grant National Natural Science Foundation of China under Grant 61573142 and Grant 61773164, in part by the Program of Introducing Talents of Discipline to Universities (the 111 Project) under Grant B17017, in part by the Shuguang Project funded by Shanghai Municipal Education Commission and Shanghai Education Development Foundation under Grant 19SG25, in part by the Ministry of Education and Science of the Russian Federation under Grant 14.756.31.0001, and in part by the Polish National Science Center under Grant UMO-2016/20/W/NZ4/00354. (*Corresponding author: Jing Jin.*)

Chang Liu, Shurui Li, Hao Sun, Yitao Huang, and Xingyu Wang 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: liuchang@mail.ecust.edu.cn; shurui1008@163.com; 13916245741@163.com; hyt2242@gmail.com; xywang@ecust.edu.cn).

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

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

Andrzej Cichocki is with the Skolkovo Institute of Science and Technology (SKOLTECH), 143026 Moscow, Russia, also with the Systems Research Institute of Polish Academy of Science, 01-447 Warsaw, Poland, and also with the Department of Informatics, Nicolaus Copernicus University, 87-100 Torun, Poland (e-mail: a.cichocki@skoltech.ru).

Digital Object Identifier 10.1109/TNSRE.2022.3156076

bands varied for different participants. With the development of deep learning methods, many researchers proposed novel feature extraction and classification methods for MI-based BCIs. Lawhern *et al.* [19] proposed a compact neural network called EEGNet, which can extract temporal and spatial features together. In a recent study, Izzuddin *et al.* [20] used SincNet, which was proposed to improve the classification performance of speech signals [21], as a band-pass filter to process MI-related EEG signals and achieve promising performance. However, these methods can not make full use of the information included in MI-related EEG signals.

Inspired by the filter-bank technique and the SincNet method, we propose a SincNet-based hybrid neural network (SHNN) in this study. First, the raw EEG is segmented into different time windows and mapped into the CSP feature space using the learned CSP projection matrices. Then, for each time window, a SincNet based CNN module is used to extract the spatial and spectral features from the mapped EEG. In addition, after the SincNet method, we also use squeeze-and-excitation (SE) modules to re-calibrate the data processed by the different maps of SincNet to obtain a sparse representation of MI-related EEG. Next, features extracted from different time windows are concatenated and fed into a gated recurrent unit (GRU) module to seek sequential relations in the data. Finally, a fully connected layer was used for classification. To verify the effectiveness of the proposed neural network, our SHNN method is evaluated on two public BCI competition datasets. Ablation experiments were also conducted to explore the function of each module in our SHNN method.

The rest of our paper is organized as follows. In section II we introduce our SHNN method and datasets. In section III we present our experiment results. Finally, in section IV and section V we present our discussion and conclusion, respectively.

## II. METHODS

In this section, we first introduce the method that was used to extract features from the raw EEG. Then we show the detailed structure of our proposed SHNN method. Next, we describe the datasets used in this study. Finally, we describe our experimental process.

### A. Common Spatial Pattern Features Extraction With Retaining Temporal Information

The Common spatial pattern (CSP) method has been widely used to classify ERD/ERS activity in the EEG. The principle of CSP is to seek a spatial projection matrix to maximize the covariance of one class in the EEG whilst simultaneously minimizing the covariance of the other class in the EEG (see equation (2)) [22]. For multi-class MI tasks, the one-versus-rest CSP (OVR-CSP) considers one task as one class and the remaining tasks as the other class [23]. Assume the  $X_{i,j} \in R^{C \times T}$  denotes the  $j$ -th EEG trial belonging to  $i$ -th class, where  $C$  and  $T$  are the number of channels and timepoints, respectively. The typical steps to obtain the spatial filter and CSP features are:

- (1) Use a band-pass filter with a specific cut-off range to process the raw EEG data. Then subtract the mean from the filtered data.
- (2) Calculate the covariance matrix of  $i$ -th class:

$$\Sigma_i = \frac{1}{N_i} \sum_{j=1}^{N_i} X_{i,j} X_{i,j}^T \quad (1)$$

where  $N_i$  denotes the number of EEG trials belonging to  $i$ -th class.

- (3) Use equation (3) to solve equation (2) and obtain the spatial projection matrix  $W$ :

$$\arg \min_w \frac{w^T \Sigma_1 w}{w^T \Sigma_2 w}, \quad s.t. \quad w^T \Sigma_1 w + w^T \Sigma_2 w = 1 \quad (2)$$

$$\Sigma_1 W = \lambda \Sigma_2 W \quad (3)$$

where  $w$  is a column vector of the spatial projection matrix  $W$ .

- (4) Construct the spatial filter matrix  $\tilde{W} \in R^{C \times 2M}$  with the hyperparameter  $M$  that is the number of selected eigenvectors corresponding to the largest and smallest eigenvalues.
- (5) Transfer the filtered EEG data into CSP feature space by the projection of  $\tilde{W}$ :

$$Z_{i,j} = \tilde{W}^T X_{i,j} \quad (4)$$

The traditional CSP method further uses a log method to extract CSP features from EEG data mapped into the CSP feature space. However, the extracted CSP features lack temporal information. In this study, we used the EEG data  $Z_{i,j}$  transferred into CSP feature space (hereafter referred to as CSPWT) as the input to our proposed neural network.

### B. SincNet-Based Hybrid Neural Network

In this study, we propose a SincNet-based hybrid neural network (SHNN) to classify MI tasks. Fig. 1 shows the time flow of the classification process of our SHNN method. The raw EEG data were first divided into multiple time windows, and the CSPWT method was applied to the EEG in each time window to extract the features. The extracted spatial and temporal features from each window were then fed into a sub-CNN module for learning deep features, then the deep features were fed into the GRU module for classification.

As shown in Fig. 2, the SincNet, CNN, and GRU methods are the three main parts of the proposed neural network. We first applied the SincNet method of  $N_f$  maps in our network to band-pass filter the EEG data that had been transformed into the CSP feature space (Fig. 2(c)). The filter bank parameters, including the low cut-off frequency and the filter's bandwidth, can be auto-adaptive, which can result in a minimum classification error.

The SincNet method was first proposed to discover more meaningful filters for the better recognition of speech signals [21]. Its structure like 1D convolution and the traditional 1D convolution is defined as follows:

$$y[n] = x[n] * h[n] = \sum_{l=0}^{L-1} x[l] \cdot h[n-l] \quad (5)$$

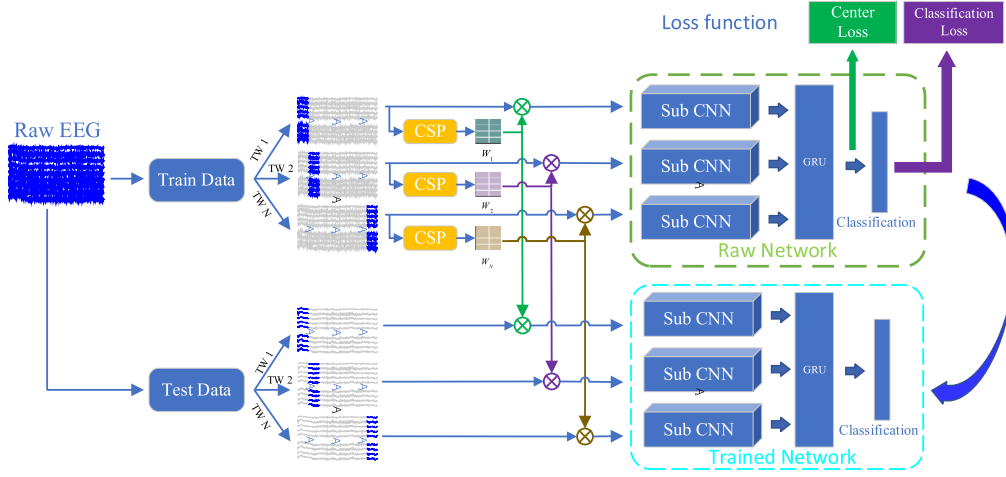


Fig. 1. Time flow of the classification progress of the SincNet-based hybrid neural network.

where  $x[n]$  is the input signal,  $y[n]$  is the filtered signal and  $h[n]$  is the filter kernel of length  $L$ . Motivated by the 1D convolution, the SincNet performs the convolution with a predefined function  $g$ :

$$y[n] = x[n] * g[n, \theta] \quad (6)$$

where  $\theta$  is the parameters that SincNet needs to learn.

In the frequency domain, a band-pass filter can be seen as a difference between two low-pass filters. Thus, the  $g[n]$  in equation (6) can be viewed in the frequency domain as:

$$G[f, f_1, f_2] = \text{rect}\left(\frac{f}{2f_2}\right) - \text{rect}\left(\frac{f}{2f_1}\right) \quad (7)$$

where  $\text{rect}(\cdot)$  denotes rectangular function,  $f_1$  and  $f_2$  are the low and high cut-off frequencies, respectively. With the use of inverse Fourier transform, the time domain equation can be written as:

$$\begin{aligned} g[n, f_1, f_2] &= 2f_2 \frac{\sin(2\pi f_2 n)}{2\pi f_2 n} - 2f_1 \frac{\sin(2\pi f_1 n)}{2\pi f_1 n} \\ &= 2f_2 \text{sinc}(2\pi f_2 n) - 2f_1 \text{sinc}(2\pi f_1 n) \end{aligned} \quad (8)$$

Therefore, SincNet can act as a band-pass filter by learning parameters  $f_1$  and  $f_2$ . However, to satisfy the limitation of  $f_1 \geq 0$  and  $f_2 \geq f_1$  in the training process, the parameters can be modified as:

$$f_1^{abs} = |f_1| \quad (9)$$

$$f_2^{abs} = f_1^{abs} + f_{band} \quad (10)$$

where  $f_{band} = |f_1 - f_2|$  is the cut-off range of the band-pass filter. Hence,  $f_1^{abs}$  and  $f_{band}$  are the parameters that SincNet needs to update during the training process. Moreover, to mitigate the problem of filter truncation, we windowed the convolutional filter with the Hamming windows, and the windowed filter kernel is:

$$g_w[n, f_1, f_2] = g[n, f_1, f_2] \cdot w[n] \quad (11)$$

where

$$w[n] = 0.54 - 0.46 \cos\left(\frac{2\pi n}{L}\right) \quad (12)$$

The outputs of SincNet were fed into the squeeze-and-excitation (SE) modules [24], [25] for recalibration. The structure of the SE modules is shown in Fig. 2(a). Suppose the input features map  $X \in \mathbb{R}^{N_f \times N_c \times N_t}$  where  $N_f$  is the number of maps of SincNet,  $N_c$  and  $N_t$  are the numbers of channels and time points. We used average pooling to generate band-wise statistics:

$$s_i = \frac{1}{c \times t} \sum_{j=1}^{j=N_c} \sum_{k=1}^{k=N_t} X_{i,j,k}, \quad i = 1, 2, \dots, N_f \quad (13)$$

Next, two fully connected layers were used to learn the non-linear relations between different bands, then obtain the scale vector:

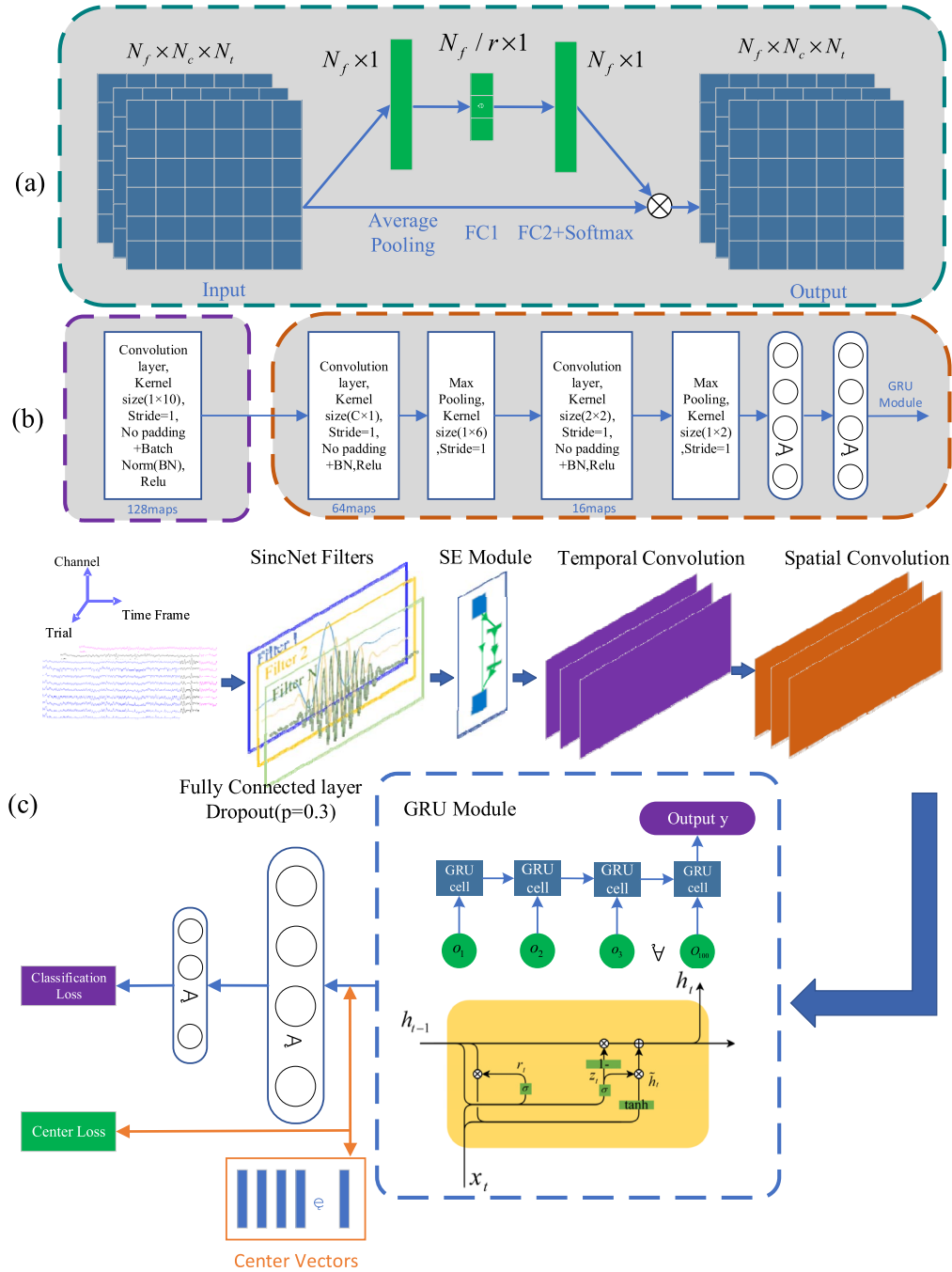
$$W = \sigma(W_{F2}, \delta(W_{F1}, S)) \quad (14)$$

where  $W$  is the scale vector,  $S = \{s_1, s_2, \dots, s_{N_f}\}$  denotes the band-wise statistics vector,  $W_{F1} \in \mathbb{R}^{\frac{N_f}{r} \times N_f}$  and  $W_{F2} \in \mathbb{R}^{N_f \times \frac{N_f}{r}}$  are weight matrixes of two fully connected layers,  $r$  is the reduction ratio, which is one of the hyperparameters,  $\delta(\cdot)$  and  $\sigma(\cdot)$  denote the rectified linear unit (ReLU) activation function and softmax function respectively. Note, the recalibrated output features of the SE modules have the same shape as the input features.

Afterward, temporal and spatial convolutional blocks were used to further extract deep features. Specifically, the temporal convolutional block consists of a convolutional layer with a kernel size of  $1 \times 10$  followed by a batch normalization layer and a ReLU layer. The mathematical formula of the ReLU function can be given as follows:

$$\text{ReLU}(x) = \max(0, x) \quad (15)$$

In addition, the spatial convolutional block has a convolutional layer with a kernel size of  $C \times 1$  (where  $C$  is the number of the EEG channels), a max-pooling layer with a kernel size of  $1 \times 6$ , a convolutional layer with a kernel size of  $2 \times 2$ , a max-pooling layer with a kernel size of  $1 \times 2$ , and a fully connected layer with 100 nodes. As shown in Fig. 2(b), the



**Fig. 2.** Schematic illustration of the proposed SincNet-based hybrid neural network. (a) The structure of the SE module, where the term  $r$  denotes the reduction ratio. (b) Diagram of the temporal and spatial convolutional blocks. (c) Diagram of the sub CNN (upper), GRU, and classification module (bottom) in Fig. 1. The “FC1” and “FC2” denote the first and second fully connected layer used in the SE module and  $r$  denotes the reduction ratio.

number of maps of three convolutional layers are 128, 64, and 16, respectively. Thus, the output dimension of the spatial convolutional block is  $16 \times 100$ .

According to Fig. 1, we concatenated the output feature vectors from the CNNs. The inputs to the CNNs are  $N_T$  different time window segmentations of the EEG mapped into the CSP feature space. These feature vectors will be fed into the gated recurrent unit (GRU) module to learn sequential information. The structure of GRU is shown in Fig. 2(c). The input of the GRU module is a vector of length 100. A reset

gate  $r_t$  and an update gate  $z_t$  were used in the GRU cells. The reset gate can adjust to allow the incorporation of new inputs with the hidden state of the previous GRU cell. Its mathematic formula can be written as follows:

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \quad (16)$$

where  $W_r$  and  $U_r$  are weight matrixes of the reset gate,  $\sigma(\bullet)$  denotes sigmoid activation function, and  $x_t$  and  $h_{t-1}$  denote the input at time  $t$  and the hidden state at time  $t - 1$ , respectively.



The update gate  $z_t$  controls the degree of new information that will be updated. The mathematical formula can be given as follows:

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \quad (17)$$

Here,  $\sigma(\bullet)$ ,  $x_t$  and  $h_{t-1}$  are the same as previously introduced in equation (16),  $W_z$  and  $U_z$  are weight matrices of the update gate. Consequently, the candidate state  $\tilde{h}_t$  and output  $h_t$  can be obtained by calculating equations (18) and (19):

$$\tilde{h}_t = \tanh(W x_t + U(r_t \otimes h_{t-1})) \quad (18)$$

$$h_t = z_t \otimes \tilde{h}_t + (1 - z_t) \otimes h_{t-1} \quad (19)$$

where  $\tanh(\bullet)$  denotes hyperbolic tangent operator,  $W$  and  $U$  are weight matrices, and  $\otimes$  denotes the Hadamard product operator. Finally, the latest outputs of the GRU module were fed into a fully connected layer and a softmax layer for the classification.

Since the lengths of different time windows are the same, the size of inputs of sub-CNN (Fig. 1) is identical. Therefore, the sub-CNN structure is consistent between different time windows.

### C. Loss Function

In this study, we used the cross-entropy loss to minimize the classification error between the predicted labels and the ground-truth labels. Moreover, the sparse loss [24] and center loss [26] were used to simplify the neural network and improve the discriminability of different class features, respectively. The objective functions of cross-entropy loss  $L_{CE}$ , sparse loss  $L_{Sparse}$ , and center loss  $L_{Center}$  are given as follows:

$$L_{CE} = -\frac{1}{N_{batch}} \sum_{i=1}^{N_{batch}} y_i^g \log y_i^p \quad (20)$$

$$L_{Sparse} = \frac{1}{N_T} \sum_{j=1}^{N_T} \|W_j\|_1 \quad (21)$$

$$L_{Center} = \frac{1}{2} \sum_{j=1}^{N_{batch}} \|f_k - v_{y_k}^e\|_2 \quad (22)$$

$$L_{total} = L_{CE} + L_{Sparse} + \lambda * L_{Center} \quad (23)$$

where  $y_i^g$  denotes the ground-truth label of the  $i$ -th training sample,  $y_i^p$  denotes the normalized probability values of the  $i$ -th training sample,  $N_{batch}$  is the number of samples in a training batch,  $W_j$  is the scale vector of the filter banks in the  $j$ -th time window,  $\|\bullet\|$  denotes  $l_1$ -norm operator,  $f_k$  denotes the feature vector extracted from  $k$ -th sample by the GRU module,  $y_k$  denotes the ground-truth label of the  $k$ -th sample, and  $v_{y_k}^e$  denotes the center vector of class  $y_k$  in the  $e$ -th training epoch.

As introduced in [27], the center vectors can be updated in each training epoch through the following formulas:

$$\begin{cases} \Delta v_j^e = \frac{\sum_{i=1}^{N_{batch}} \theta(y_i = j) * \|v_j^e - f_i\|_2}{1 + \sum_{i=1}^{N_{batch}} \theta(y_i = j)} \\ v_j^{e+1} = v_j^e - \rho * \Delta v_j^e \end{cases} \quad (24)$$

$$\theta(y_i = j) = \begin{cases} 0, & y_i \neq j \\ 1, & y_i = j \end{cases} \quad (25)$$

where  $\Delta v_j^e$  is the average distance between the  $j$ -th class samples and center vectors of the  $j$ -th class in the  $e$ -th training epoch, and  $v_j^{e+1}$  and  $v_j^e$  are the center vectors in the  $(e+1)$ -th and  $e$ -th training epochs, respectively. The term  $\rho$  denotes the learning rate for center loss, and the value of  $\rho$  is in the range of 0 to 1.

The center vectors can be updated with the training process, and the center vectors are firstly initialized with random Gaussian noise of dimension  $N_c \times 100$ , where  $N_c$  denotes the classes number of the MI tasks, 100 is the length of the GRU output vector. The target of center loss is to minimize the gap between the center vectors of one class and its GRU outputs, which can make the features extracted from the same class of data more aggregated (as shown in Fig. 6). Therefore, the ability of the neural network to identify different classes of data could be improved

### D. Dataset Description

In this study, we used two public BCI competition datasets to validate the effectiveness of the proposed neural network. The detailed information of the two datasets are shown as follows:

*DS1*: The first dataset is the BCI competition IV dataset 2b [28], which was recorded from nine participants (B1-B9) at a sample rate of 250 Hz. For each participant, 720 trials from two MI tasks (left hand vs right hand motor imagery imagination) were performed, and the EEG was recorded from 3 electrodes placed at positions C3, Cz, and C4 in the international 10/20 system for EEG electrode placement. Detailed information about this dataset can be found via [www.bbc.de/competition/iv/#dataset2b](http://www.bbc.de/competition/iv/#dataset2b).

*DS2*: The second dataset is the BCI competition IV dataset 2a [29], which was recorded from nine participants (A1-A9). For each participant, 576 trials of EEG were recorded from 22 Ag/AgCl electrodes at a sample rate of 250 Hz. In this dataset, four kinds of MI tasks were performed. Specifically, participants performed left hand (class 1), right hand (class 2), tongue (class 3), and feet (class 4) motor imagery tasks. The number of trials recorded for each class is 144. Detailed information about this dataset can be found via [www.bbc.de/competition/iv/#dataset2a](http://www.bbc.de/competition/iv/#dataset2a).

### E. Experiment Setup

We used the adaptive moment (Adam) optimizer [30] to train our proposed neural network and the optimizer parameters  $\beta_1$  and  $\beta_2$  were set to 0.5 and 0.999, respectively. The learning rate  $\rho$  and weight of center loss were set to 0.00005 and 0.1, respectively. The batch size and learning rate of the neural network were set to 16 and 0.0001. In addition, the reduction ratio  $r$  in the SE module was chosen as 3. The number of maps of each SincNet is 15 denoting that there are 15 auto-adaptive band-pass filters. Due to the 3 seconds of motor imagination time in DS1 and DS2, we segmented the raw EEG into four-time windows of length 2 seconds, which started from -0.5 seconds to 1 second with an interval of 0.5 seconds (i.e., -0.5s~1.5s, 0s~2s, 0.5s~2.5s, and 1s~3s). In this study, due to the SincNet acting as band-pass filters,

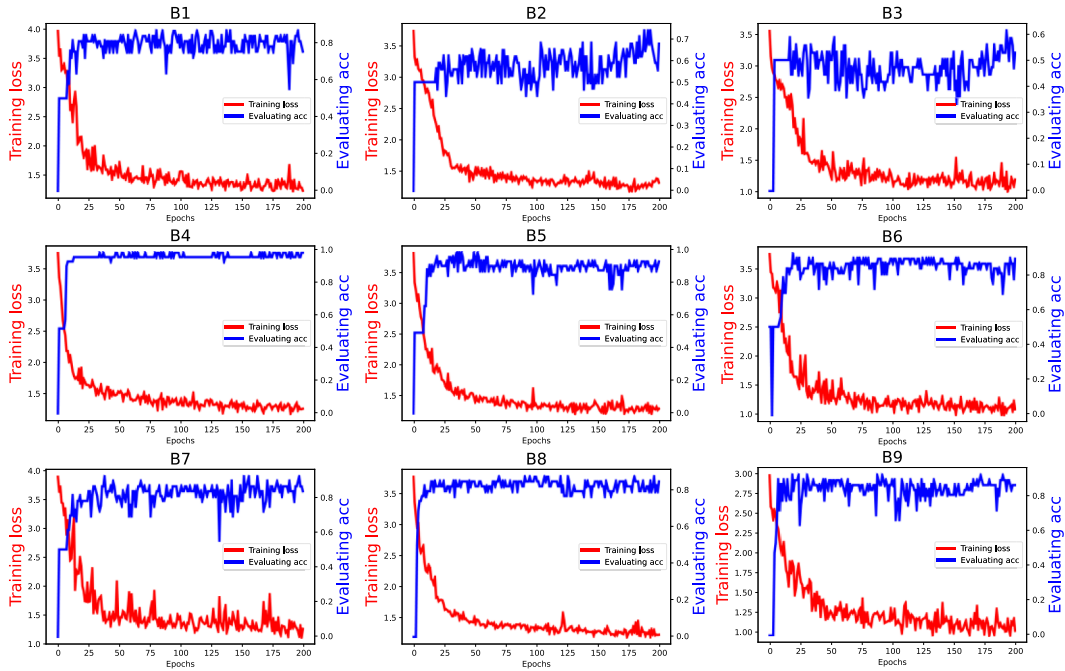


Fig. 3. The training loss and evaluating accuracy of each participant (B1-B9) on DS1. We iteratively trained the SHNN method 200 times for each participant. For each training epoch, the validating datasets were used to validate the SHNN method after training. The mean training time of 200 epochs for each participant is 399.31 seconds. The red lines denote the training loss curves, while the blue lines denote the evaluating accuracy curves.

we only use the notch-filter to remove the 50/60Hz power-line noise of the raw EEG. We trained our proposed neural network with Pytorch on an AMD R7 3700X CPU, 32 GB RAM, and the Nvidia 2080Ti platform. For each participant, eight of the ten training datasets were used to train the SHNN method, while one of the ten datasets was used as a validation dataset, and the performance of our SHNN method was evaluated on the last of the ten datasets.

In this study, Cohen’s kappa coefficient and classification accuracy are used as two metrics for performance evaluation. The mathematical formula of Cohen’s kappa coefficient is given as follows:

$$k = \frac{p_o - p_e}{1 - p_e} \quad (26)$$

where  $p_o$  denotes the classification accuracy,  $p_e = \frac{\sum_i n_{oi}n_{io}}{N^2}$  denotes the hypothetical probability of chance agreement,  $n_{oi}$  and  $n_{io}$  are the sum of each column and row of the confusion matrix, and  $N$  denotes the sum of all entries in the confusion matrix.

### III. RESULT

#### A. The Training Loss and Evaluating Accuracy of the SHNN Method

We first analyzed the training loss and the accuracy of the SHNN method on DS1. Fig. 3 shows the training loss and accuracy for each participant (B1-B9) on DS1. The number of training epochs is 200. For each training epoch, the validating datasets were used to validate the SHNN method after training. The red lines denote the training loss curves, while the blue

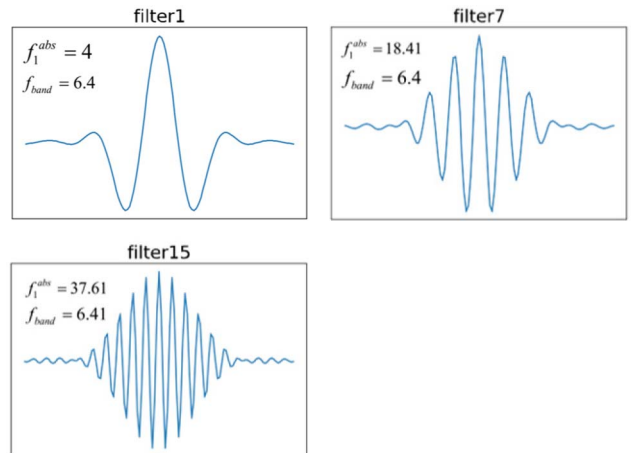


Fig. 4. Examples of the first, seventh and fifteenth sinc-filters  $g_w[n]$  learned by the SincNet method for participant B4 on DS1. The  $f_1^{abs}$  and  $f_{band}$  are two parameters, namely, the low frequency and bandwidth of the sinc-filter.

lines denote the accuracy curves. As shown in Fig. 3, the accuracy is increased while the training loss is reduced. It can be seen that the accuracy increased quickly during the first 25 training epochs and the training loss is generally stable after training about 100 epochs, which suggests that the SHNN method can be trained well for MI task classification. The average training time and single-trial test time are 399.31 s and 2.03 s, respectively.

We take one participant as an example to illustrate the operation of our method. Fig. 4 shows the first, seventh, and

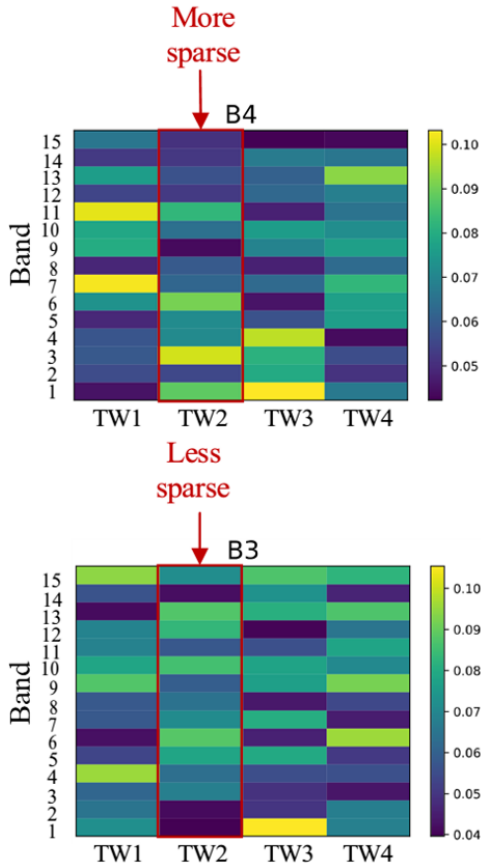


Fig. 5. The weight values of the EEG data filtered by 15 sinc-filters across four time windows (TW1-TW4) for participants B3 and B4 on DS1.

fifteenth sinc-filters  $g_w[n]$  learned by the SincNet method for participant B4 on DS1. The low frequency signal components and optimal bandwidths were automatically identified by SincNet and varied across the different sinc-filters. The first sinc-filter can extract spectral information from 4 Hz to 10.4 Hz, while the seventh and fifteenth sinc-filters extract spectral information from 18.41 Hz to 24.81 Hz and 37.61 Hz to 44.02 Hz, respectively. The mu and beta rhythms (8-30 Hz) that have been proved crucial for the MI tasks classification [9] are included in the band-pass frequency range of these sinc-filters.

To better show the role of the SE module in our SHNN method, we also illustrate some examples of the weight values learned by the SE module. Fig. 5 shows examples of the weight values of the EEG data filtered by the 15 sinc-filter of the SincNet method across the four-time windows for participants B3 and B4 on DS1.

The classification accuracy and kappa values for participants B3 and B4 are (B3: accuracy = 0.5833, kappa = 0.1667, and B4: accuracy = 0.9730, kappa = 0.9458), which are the worst and best performances achieved by the SHNN method on DS1. As shown in Fig. 5, the weight values from 0 to 1 were automatically assigned to the data after filtering by the different sinc-filters. When inspecting time window 2 (TW2), comparing the weight values distribution of participants B3 and B4, reveals that, for participant B3, the weight values of

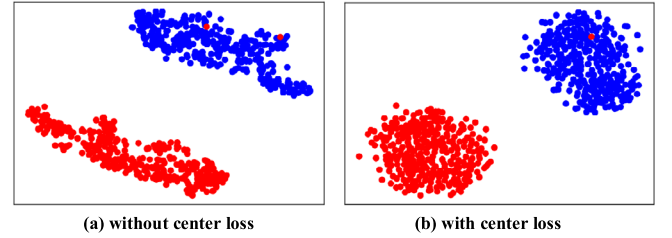


Fig. 6. The distribution of feature vectors extracted by SHNN (a) without and (b) with center loss for participants B4 on DS1. All feature vectors are mapped in the 2D embedding space by t-SNE method.

frequency bands 3 to band 13 are similar, while for participant B4, the weight values of the different frequency bands are more sparse. Thus, a more sparse representation of features may contribute to a better classification performance due to the suppression of redundant information.

We used the t-distributed stochastic neighbor embedding (t-SNE) method [31] to map the feature vectors to a 2D embedding space before and after applying the center loss method. The distribution of feature vectors for participant B4 is shown in Fig. 6. The red and blue dots indicate the left-hand and right-hand classes of samples, respectively. It can be seen that after applying the center loss method, the feature vectors of the same class are gathering together and the variance is smaller. Hence, the center loss method can improve the feature vectors' power for the discriminating different class samples, which also can be proved by the ablation experiments results in Table I (see below).

## B. Result of the Ablation Experiments

Three ablation experiments were conducted on DS1 to verify the effectiveness of the SincNet method and specific loss function (e.g., sparse  $l_1$ -norm loss and center loss) within our proposed SHNN method. We used three models named model1, model2, and model3, which represent three cases:

1) *Model1*: The SHNN method is tested without any additional structure (SincNet and SE modules) and only trained with cross-entropy loss.

2) *Model2*: The SHNN method is tested without the SincNet module and trained with cross-entropy loss, sparse  $l_1$ -norm loss and center loss.

3) *Model3*: The complete SHNN method is only trained with cross-entropy loss.

Table I shows the classification accuracies, kappa values, and Wilcoxon signed-rank test results of three ablation experiments, respectively. The mean classification accuracies of model1, model2, and model3 are 0.7428, 0.7827, and 0.8044, while the mean classification accuracy of our proposed model is 0.8349. Moreover, the mean kappa values of model1, model2, and model3 are 0.4858, 0.5643, and 0.6086, while the mean kappa value of our proposed model is 0.6697. In addition, the Wilcoxon signed-rank test results demonstrate that our complete SHNN method, i.e., included the SincNet and SE modules and trained with cross-entropy loss, sparse  $l_1$ -norm loss and center loss, outperformed the model1

TABLE I

THE CLASSIFICATION ACCURACIES, KAPPA VALUES AND WILCOXON SIGNED-RANK TEST RESULTS OF ABLATION EXPERIMENTS ON DS1

Participant	Model1 Acc(Kappa)	Model2 Acc(Kappa)	Model3 Acc(Kappa)	Proposed Model Acc(Kappa)
B1	0.7583(0.5167)	0.7778(0.5556)	0.8055(0.6111)	0.8333(0.6667)
B2	0.5588(0.1176)	0.5882(0.1765)	0.5882(0.1765)	0.6176(0.2353)
B3	0.5278(0.0556)	0.5556(0.1111)	0.5833(0.1667)	0.5833(0.1667)
B4	0.8919(0.7842)	0.9459(0.8918)	0.9189(0.8370)	0.9730(0.9458)
B5	0.8378(0.6763)	0.8649(0.7283)	0.8919(0.7830)	0.9189(0.8375)
B6	0.7500(0.5000)	0.8056(0.6111)	0.8333(0.6667)	0.8889(0.7778)
B7	0.7778(0.5556)	0.8333(0.6667)	0.8611(0.7222)	0.8611(0.7222)
B8	0.8053(0.6105)	0.8421(0.6842)	0.8684(0.7368)	0.9211(0.8421)
B9	0.7778(0.5556)	0.8308(0.6531)	0.8889(0.7778)	0.9167(0.8333)
Mean	0.7428(0.4858)	0.7827(0.5643)	0.8044(0.6086)	0.8349(0.6697)
p-value (vs. proposed model)	0.008	0.008	0.018	-

TABLE II

THE CLASSIFICATION ACCURACIES, KAPPA VALUES, AND WILCOXON SIGNED-RANK TEST RESULTS OF THE DIFFERENT METHODS APPLIED TO ALL PARTICIPANTS IN DS1

Participants	Raza et al [37]	Oh et al [35]	Xu et al [36]	Bashashati et al [34]	Proposed Model
	Acc	Acc	Acc	Acc	Acc(Kappa)
B1	0.7500	0.8060	0.7260	0.7140	0.8333(0.6667)
B2	0.5156	0.6160	0.6030	0.6057	0.6176(0.2353)
B3	0.5219	0.6060	0.6690	0.5809	0.5833(0.1667)
B4	0.9656	0.9810	0.9120	0.9713	0.9730(0.9458)
B5	0.7719	0.8180	0.8060	0.9132	0.9189(0.8375)
B6	0.7406	0.8430	0.7060	0.8594	0.8889(0.7778)
B7	0.7000	0.7250	0.7320	0.7698	0.8611(0.7222)
B8	0.8844	0.8620	0.7770	0.9196	0.9211(0.8421)
B9	0.7500	0.8600	0.7120	0.8453	0.9167(0.8333)
Mean	0.7333	0.7910	0.7381	0.7977	0.8349(0.6697)
p-value (vs. proposed model)	0.008	0.038	0.021	0.008	-

( $Z=-2.670$ ,  $p=0.008$ ), model2 ( $Z=-2.666$ ,  $p=0.008$ ) and model3 ( $Z=-2.371$ ,  $p=0.018$ ).

### C. Comparison of the Classification Accuracies and Kappa Values

In order to validate the efficacy of our SHNN method, we compared its performance with other state-of-the-art methods, including EEGNet, FBCSP, and so on [18], [19], [32]–[37]. Table II shows the classification accuracies, kappa values, and Wilcoxon signed-rank test results from the nine participants in DS1. Table III shows the classification accuracies, kappa values, and Wilcoxon signed-rank test results of the nine participants in DS2.

As shown in Tables II and III, our SHNN method achieved the highest mean classification accuracies of 0.8349 and 0.7462, and the highest mean kappa values of 0.6697 and 0.6648 on datasets DS1 and DS2, respectively. Compared with other state-of-the-art methods, our SHNN method achieved improvements of 10.16%, 4.39%, 9.68%, and 3.72% in average classification accuracies on DS1, respectively, while for

DS2, the improvements are 9.58% (kappa), 7.48% (kappa), 14.62% (acc), and 9.19% (acc). It can be seen that, for fourteen of the eighteen participants in DS1 and DS2, our SHNN method achieved the highest classification performance. Since the classification results were not normally distributed (as confirmed by Kolmogorov–Smirnov test results), we used Wilcoxon signed-rank test to compare the performance of our SHNN method with the other state-of-the-art methods. The statistical test results demonstrate the superiority of our SHNN method in the two-class and multi-class MI classification tasks. The p-value between SHNN and eight compared method on two datasets are 0.008 ( $Z=-2.666$ ), 0.038 ( $Z=-2.073$ ), 0.021 ( $Z=-2.310$ ), 0.008 ( $Z=-2.666$ ), 0.008 ( $Z=-2.666$ ), 0.028 ( $Z=-2.192$ ), 0.008 ( $Z=-2.666$ ) and 0.038 ( $Z=-2.073$ ), respectively.

## IV. DISCUSSION

CSP is a very widely used and popular feature extraction method for MI classification tasks. However, previous studies that have used CSP to extract spatial features from EEG ignore



**TABLE III**  
THE CLASSIFICATION ACCURACIES, KAPPA VALUES, AND WILCOXON SIGNED-RANK TEST RESULTS OF THE DIFFERENT METHODS FOR ALL PARTICIPANTS IN DS2

Participants	FBCSP	Asensio-Cubero et al	Kam et al	EEGNet	Proposed Model
	[18]	[32]	[33]	[19]	
	Kappa	Kappa	Acc	Acc	Acc(Kappa)
A1	0.6760	0.7500	0.7400	0.7188	0.8276(0.7706)
A2	0.4170	0.5000	0.3500	0.5104	0.6897(0.5864)
A3	0.7450	0.7400	0.7600	0.7917	0.7931(0.7538)
A4	0.4810	0.4000	0.5300	0.5799	0.6552(0.5426)
A5	0.3980	0.1900	0.3800	0.6458	0.5862(0.4467)
A6	0.2730	0.4100	0.3100	0.5104	0.4828(0.3539)
A7	0.7730	0.7800	0.8400	0.6632	0.8621(0.8162)
A8	0.7550	0.7200	0.7400	0.7431	0.8966(0.8619)
A9	0.6060	0.7800	0.7400	0.7257	0.8987(0.8528)
Mean	0.5690	0.5900	0.6000	0.6543	0.7426(0.6648)
p-value (vs. proposed model)	0.008	0.028	0.008	0.038	

the temporal information contained in the EEG. Moreover, the performance of CSP is easily affected by the selection of the cut-off frequencies of the band-pass filters used with the method. In this study, we proposed a new method, SHNN, to automatically learn spatial-spectral-temporal information from the EEG.

The four main advantages of our SHNN method are: (a) SincNets were used to automatically identify band-pass filters from the data, meaning the parameters of each filter can be automatically learned from the data, which results in a better spectral features extraction of the MI-related information from the EEG. (b) The use of the GRU module and CSPWT feature extraction method can make full use of the temporal information in the EEG and boost the classification performance. (c) Center loss was used to improve the distinguishability of different classes of samples. (d) The sparse representation of data filtered into different frequency bands can be learned by the SE module.

The function of the SincNets module in our SHNN method is to band-pass filter the EEG into different frequency ranges. As shown in Fig.4, according to the characteristics of the EEG data, each SincNet can determine a specific frequency range in order to optimally band-pass filter the data into spectral features. Moreover, the optimal cut-off ranges of the band-pass filters vary across different participants and the SincNet module can make full use of the differences in spectral information between participants. Hence, our SHNN method can extract more discriminative spectral information from the EEG for classification.

The traditional CSP method has the limitation of losing temporal information of EEG. In this study, we used the CSP projection matrix to map EEG data into the CSP feature space while retaining the temporal information from the EEG. In addition, the GRU module was used to learn the sequential information from different time window segmentations of the EEG. Furthermore, the center loss was used to train our SHNN method, which allows it to reduce the variance of a single class of samples and gather the samples together. Fig. 6 shows

the clustering performance of the center loss method. The feature vectors from a single class are gathered together and the variance is reduced. Table I demonstrates that the use of center loss can improve the classification performance of our SHNN method (with center loss vs. without center loss, DS1 accuracies: 0.8349 vs. 0.8044, DS2 accuracies: 0.6697 vs. 0.6086).

The SE module in our method can re-calibrate the data by explicitly modeling interdependencies between EEG channels [23]. As shown in Fig. 5, different weight values were automatically assigned to data processed by different SincNets and segmented into the different time windows. The frequency bands include discriminated information that can be assigned to a higher weight value, which can enhance the features extracted from those bands. On the other hand, the redundant features could be suppressed by assigning a lower or zero weight value. Therefore, the SE module can improve the representational power and ease the learning of our SHNN method.

Of course, our SHNN method still has some limitations. Although Tables II and III show the superiority of our SHNN method, the structure of SHNN is complex. Recently, pruning [38] and knowledge distillation [39] techniques have been used to simplify the structure of neural networks while maintaining their performance at the same time. We will try to combine these methods with our SHNN method to achieve a more compact neural network. Secondly, our SHNN method is an offline neural network that has not been validated in online BCI environments. In future work we will explore its use in online BCI experiments.

## V. CONCLUSION

In this study, we proposed a SincNet based hybrid neural network consisting of SincNets, SE modules, and GRU modules, which can automatically filter data and extract spatial, spectral, and temporal features from EEG. The results of our experiments, conducted on two public BCI competition datasets, demonstrate that the performance of our SHNN

method is significantly better than that of other state-of-the-art methods. Our ablation experiments result also demonstrate the effectiveness of the SincNet and SE modules within our method.

## REFERENCES

- [1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain-computer interfaces for communication and control," *Clin. Neurophysiol.*, vol. 113, no. 6, pp. 767–791, 2002.
- [2] G. Cattan, C. Mendoza, A. Andreev, and M. Congedo, "Recommendations for integrating a P300-based brain computer interface in virtual reality environments for gaming," *Computers*, vol. 7, no. 2, p. 34, May 2018.
- [3] S. Chen *et al.*, "Longitudinal electroencephalography analysis in Subacute stroke patients during intervention of brain-computer interface with exoskeleton feedback," *Frontiers Neurosci.*, vol. 14, p. 809, Jan. 2020.
- [4] J. Choi, K. T. Kim, J. H. Jeong, L. Kim, S. J. Lee, and H. Kim, "Developing a motor imagery-based real-time asynchronous hybrid BCI controller for a lower-limb exoskeleton," *Sensors*, vol. 20, no. 24, p. 7309, Dec. 2020.
- [5] Y. Miao *et al.*, "BCI-based rehabilitation on the stroke in sequela stage," *Neural Plasticity*, vol. 2020, pp. 1–10, Dec. 2020.
- [6] R. Xu *et al.*, "A closed-loop brain-computer interface triggering an active ankle-foot orthosis for inducing cortical neural plasticity," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 7, pp. 2092–2101, Jul. 2014.
- [7] J. Huang *et al.*, "Hybrid asynchronous brain-computer interface for yes/no communication in patients with disorders of consciousness," *J. Neural Eng.*, vol. 18, no. 5, 2021, Art. no. 056001.
- [8] F. Wang *et al.*, "Enhancing clinical communication assessments using an audiovisual BCI for patients with disorders of consciousness," *J. Neural Eng.*, vol. 14, no. 4, 2017, Art. no. 046024.
- [9] J. Jin, C. Liu, I. M. Y. Daly, S. Li, X. Wang, and A. Cichocki, "Bispectrum-based channel selection for motor imagery based brain-computer interfacing," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 10, pp. 2153–2163, Oct. 2020.
- [10] C. Zuo *et al.*, "Cluster decomposing and multi-objective optimization based-ensemble learning framework for motor imagery-based brain-computer interfaces," *J. Neural Eng.*, vol. 18, no. 2, 2021, Art. no. 026018.
- [11] P. H. Gubert, M. H. Costa, C. D. Silva, and A. Trofino-Neto, "The performance impact of data augmentation in CSP-based motor-imagery systems for BCI applications," *Biomed. Signal Process. Control*, vol. 62, Sep. 2020, Art. no. 102152.
- [12] 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. 2019.
- [13] M. Eidel and A. Kübler, "Wheelchair control in a virtual environment by healthy participants using a P300-BCI based on tactile stimulation: Training effects and usability," *Frontiers Hum. Neurosci.*, vol. 14, p. 265, Jul. 2020.
- [14] J. Jin, Z. Wang, R. Xu, C. Liu, X. Wang, and A. Cichocki, "Robust similarity measurement based on a novel time filter for SSVEPs detection," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Oct. 14, 2021, doi: 10.1109/TNNLS.2021.3118468.
- [15] B. Liu, X. Chen, X. Li, Y. Wang, X. Gao, and S. Gao, "Align and pool for EEG headset domain adaptation (ALPHA) to facilitate dry electrode based SSVEP-BCI," *IEEE Trans. Biomed. Eng.*, vol. 69, no. 2, pp. 795–806, Feb. 2022.
- [16] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K. R. Müller, "Optimizing spatial filters for robust EEG single-trial analysis," *IEEE Signal Process. Mag.*, vol. 25, no. 1, pp. 41–56, Jan. 2008.
- [17] J. Chen, Z. Yu, Z. Gu, and Y. Li, "Deep temporal-spatial feature learning for motor imagery-based brain-computer interfaces," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 11, pp. 2356–2366, Nov. 2020.
- [18] K. K. Ang, Z. Y. Chin, C. Wang, C. Guan, and H. Zhang, "Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b," *Frontiers Neurosci.*, vol. 6, no. 1, p. 39, 2012.
- [19] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "EEGNet: A compact convolutional neural network for EEG-based brain-computer interfaces," *J. Neural Eng.*, vol. 15, no. 5, 2018, Art. no. 056013.
- [20] T. A. Izzuddin, N. M. Safri, and M. A. Othman, "Compact convolutional neural network (CNN) based on SincNet for end-to-end motor imagery decoding and analysis," *Biocybern. Biomed. Eng.*, vol. 41, no. 4, pp. 1629–1645, Oct. 2021, doi: 10.1016/j.bbe.2021.10.001.
- [21] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with SincNet," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2018, pp. 1021–1028.
- [22] J. Jin, R. Xiao, I. Daly, Y. Miao, X. Wang, and A. Cichocki, "Internal feature selection method of CSP based on L1-norm and Dempster-Shafer theory," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 11, pp. 4814–4825, Nov. 2021.
- [23] C. Liu *et al.*, "Distinguishable spatial-spectral feature learning neural network framework for motor imagery-based brain-computer interface," *J. Neural Eng.*, vol. 18, no. 4, 2021, Art. no. 0460e4.
- [24] H. Zhang, X. Zhao, Z. Wu, B. Sun, and T. Li, "Motor imagery recognition with automatic EEG channel selection and deep learning," *J. Neural Eng.*, vol. 18, no. 1, p. 16004, Nov. 2020.
- [25] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
- [26] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 499–515.
- [27] L. Yang, Y. Song, X. Jia, K. Ma, and L. Xie, "Two-branch 3D convolutional neural network for motor imagery EEG decoding," *J. Neural Eng.*, vol. 18, no. 4, Aug. 2021, Art. no. 0460c7.
- [28] M. Tangermann *et al.*, "Review of the BCI competition IV," *Frontiers Neurosci.*, vol. 6, no. 1, p. 55, 2012.
- [29] C. Brunner, R. Leeb, G. Müller-Putz, A. Schlögl, and G. Pfurtscheller, "BCI Competition 2008–Graz data set A," *Inst. Knowl. Discovery (Lab. Brain-Comput. Interfaces), Graz Univ. Technol.*, vol. 16, pp. 1–6, Jan. 2008.
- [30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [31] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.
- [32] J. Asensio-Cubero, J. Q. Gan, and R. Palaniappan, "Extracting optimal tempo-spatial features using local discriminant bases and common spatial patterns for brain computer interfacing," *Biomed. Signal Process. Control*, vol. 8, no. 6, pp. 772–778, Nov. 2013.
- [33] T.-E. Kam, H.-I. Suk, and S.-W. Lee, "Non-homogeneous spatial filter optimization for ElectroEncephaloGram (EEG)-based motor imagery classification," *Neurocomputing*, vol. 108, pp. 58–68, May 2013.
- [34] H. Bashashati, R. K. Ward, and A. Bashashati, "User-customized brain computer interfaces using Bayesian optimization," *J. Neural Eng.*, vol. 13, no. 2, 2016, Art. no. 026001.
- [35] S.-H. Oh, Y.-R. Lee, and H.-N. Kim, "A novel EEG feature extraction method using Hjorth parameter," *Int. J. Electron. Elect. Eng.*, vol. 2, no. 2, pp. 106–110, 2014.
- [36] G. Xu *et al.*, "A deep transfer convolutional neural network framework for EEG signal classification," *IEEE Access*, vol. 7, pp. 112767–112776, 2019.
- [37] H. Raza, H. Cecotti, Y. Li, and G. Prasad, "Adaptive learning with covariate shift-detection for motor imagery-based brain-computer interface," *Soft Comput.*, vol. 20, no. 8, pp. 3085–3096, 2016.
- [38] J.-H. Luo, J. Wu, and W. Lin, "ThiNet: A filter level pruning method for deep neural network compression," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 5058–5066.
- [39] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.