Learning Invariant Patterns Based on a Convolutional Neural Network and Big Electroencephalography Data for Subject-Independent P300 Brain-Computer Interfaces

Wei Gao, Tianyou Yu, *Member, IEEE,* Jin-Gang Yu, Zhenghui Gu, *Member, IEEE,* Kendi Li, Yong Huang, Zhu Liang Yu, *Member, IEEE,* and Yuanqing Li, *Fellow, IEEE*

***Abstract*—A brain-computer interface (BCI) measures and analyzes brain activity and converts this activity into computer commands to control external devices. In contrast to traditional BCIs that require a subject-specific calibration process before being operated, a subject-independent BCI learns a subject-independent model and eliminates subject-specific calibration for new users. However, building subject-independent BCIs remains difficult because electroencephalography (EEG) is highly noisy and varies by subject. In this study, we propose an invariant pattern learning method based on a convolutional neural network (CNN) and big EEG data for subject-independent P300 BCIs. The CNN was trained using EEG data from a large number of subjects, allowing it to extract subject-independent features and make predictions for new users. We collected EEG data from 200 subjects in a P300-based spelling task using two different types of amplifiers. The offline analysis showed that almost all subjects obtained significant cross-subject and cross-amplifier effects, with an average accuracy of more than 80%. Furthermore, more than half of the subjects achieved accuracies above 85%. These results indicated that our method was effective for building a subject-independent P300 BCI, with which more than 50% of users could achieve high accuracies without subject-specific calibration.**

***Index Terms*—Brain-computer interface (BCI), electroencephalography (EEG), P300, cross-subject, invariant pattern learning, convolutional neural network (CNN), big data.**

I. Introduction

A brain-computer interface (BCI) provides direct communication between the brain and a computer. BCIs measure people's brain activity and translate this activity into control commands for external devices. Steady-state visual evoked potentials (SSVEPs), P300 waveforms, and motor imagery are three typical brain patterns commonly used in electroencephalography (EEG)-based BCIs. For instance, P300 speller systems can identify the character that the user wants to input by detecting the P300 component evoked by flashes of the corresponding button in a virtual keyboard [1], [2].

The use of a typical EEG-based BCI system usually requires a calibration phase, during which a training dataset is obtained while the user performs a specific task [3]. Based on the calibration data, a subject-specific model is trained, and it is subsequently used for online prediction. However, it is generally time consuming and inconvenient to collect training data from each user to allow them to operate the BCI system. Furthermore, it is unproductive with respect to BCI applications (e.g., text entry, wheelchair control). Therefore, the calibration period must be minimized, which is especially important for patients with limited ability to concentrate [4].

Many attempts have been made to shorten or eliminate the user's calibration phase and to develop subject-independent models that can work across sessions, subjects, and even EEG amplifiers. For example, Li et al. proposed several iterative semi-supervised methods for building classification models, including support vector machine (SVM) models, with small training datasets [5], [6]. Several online unsupervised methods were used to set up zero-training BCIs, including motor imagery BCIs [7] and P300 BCIs [4], [8]. However, in general, a period of adaptation was needed, during which the BCIs were gradually improved in terms of performance. Transfer learning, which aims to improve the learning of a predictive function in a target domain based on the knowledge in a source domain for a learning task [9], is a promising method for addressing such a problem. For instance, Lu et al. proposed an adaptive linear discriminant analysis (LDA) method for addressing the problem of considerable variations in EEGs across subjects, and the test data without labels from the new subjects were used to update the classification model [10]. Domain adaptation techniques based on weighted adaptation regularization [11], logistic regression [12], LDA methods [13], [14], and Riemannian geometry methods [15], [16], [17], [18] were widely used in BCIs and in other EEG analysis tasks

This work was supported in part by the Key R&D Program of Guangdong Province, China under Grant 2018B030339001, in part by the Key Realm R&D Program of Guangzhou, China under Grant 202007030007, in part by the National Natural Science Foundation of China under Grants 61633010, 61876064, and 62076099, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2019A1515011773, and in part by the Pearl River S&T Nova Program of Guangzhou under Grant 201906010043. (Corresponding author: Yuanqing Li.)

The authors are with the School of Automation Science and Engineering, South China University of Technology, Guangzhou, 510640, China, and also with the Brain Computer Intelligence Research Center, Pazhou Lab, Guangzhou, 510330, China (e-mail: augaow@mail.scut.edu.cn; auyuty@scut.edu.cn; jingangyu@scut.edu.cn; zhgu@scut.edu.cn; aulkd@mail.scut.edu.cn; auyonghuang@mail.scut.edu.cn; zlyu@scut.edu.cn; auyqli@scut.edu.cn).

such as emotion classification. Multitask learning framework-based methods were proposed to address subject-to-subject and session-to-session transfer problems in motor imagery-based BCIs, and the classifiers of multiple subjects were learned jointly such that the classification error of the new subjects was minimized [19], [20]. Yuan et al. presented a canonical correlation analysis (CCA) method that transferred SSVEP templates from the existing subjects to a new subject to enhance the detection of SSVEPs [21]. In [22], a subject-transfer framework based on multiple distance measurements among different subjects and model ranking and fusion was proposed to obviate inter- and intrasubject variability and to enhance the results of drowsiness detection with EEG. However, to the best of our knowledge, the development of a subject-independent BCI with satisfactory performance remains an open problem, mainly because EEG brain patterns are highly subject specific, dynamic and weak, with low signal-to-noise ratios (SNRs). Furthermore, most previous work on subject-independent BCIs did not consider large training datasets. When only limited EEG data are available for a specific BCI task, traditional machine learning methods have difficulty in learning reasonable feature representations. Additionally, cross-amplifier EEG data analysis, which is urgently needed for multicenter data collection, big EEG database establishment and BCI applications, has not been studied in the literature to date.

Deep neural networks (DNNs) are usually composed of multiple layers for extracting multilevel hierarchies of features [23] and for combining feature extraction with regression or classification. Encouraged by the excellent results deep learning has achieved in various machine learning tasks, researchers have attempted to introduce deep learning algorithms into the field of brain signal analysis and achieved promising results thus far. For instance, a convolutional neural network (CNN) and a recurrent convolutional neural network (RCNN) achieved very accurate results for P300-based speller systems [24], [25]. EEGNet, which is a specially designed CNN architecture for EEG analysis, performed well in P300, error-related negativity (ERN), movement-related cortical potential (MRCP), and sensory motor rhythm (SMR) tasks [26]. Readers can refer to references such as [27], [28] for a comprehensive review of deep learning for EEG data analysis and BCIs. While mainly focusing on DNN models for within-subject EEG classification tasks, researchers incorporated DNNs into a transfer learning framework to combine representational learning and knowledge transfer and thereby boost the EEG classification performance when small training datasets or no training data were available. Fahimi et al. proposed an end-to-end deep CNN to exploit intersubject information for mental attentional information decoding [29]. In [25], a CNN was used for establishing a cross-subject P300 BCI, where a small amount of labeled data from test subjects was still required so that the model could be fine-tuned to achieve satisfactory results. In [30], an end-to-end CNN model was used to improve the performance of a motor imagery BCI, and a transfer learning approach was proposed to adapt the global classifier to single individuals to improve the accuracy. In [31], a transfer learning method named online pre-alignment strategy (OPS) combined with two deep learning models was proposed to enhance the cross-subject and cross-dataset performances in the motor imagery paradigm without any additional calibration data, which is a good attempt to build a subject-independent motor imagery BCI. Although a few studies have shown that deep learning approaches could be used to transfer knowledge from the EEG data of a number of subjects to other subjects, such studies are still in their infancy. For three typical BCI paradigms, i.e., P300, SSVEP and motor imagery-based BCIs, few subject-independent DNN models that can work across sessions, subjects, and different EEG amplifiers have been reported, possibly because big EEG data were not available and hence the value of big EEG data was not explored.

In this study, an invariant pattern learning method based on a CNN and big EEG data is proposed for use in a subject-independent P300 BCI speller system. We built an EEG dataset of 200 subjects. Each subject performed a P300-based spelling task while data were recorded using two types of amplifiers. The subjects were divided into two groups: a group with 150 subjects building the training and validation sets and a group with the other 50 subjects for an independent test. After the CNN was trained, the preprocessed EEG data acquired from the test subjects, consisting of data corresponding to each character, were input into the CNN, whose output was regarded as the probability of the presence of the P300 component. The character with the maximum probability was identified as the character that the user intended to input. The experimental results demonstrated that almost all subjects obtained significant cross-subject and cross-amplifier effects using our method. The average accuracy was more than 80% after 10 rounds of button flashes in the offline analysis, and the average accuracy was 75.34% after an average of 6.04 rounds of button flashes in the simulated online test. Furthermore, more than 50% of subjects achieved accuracies of more than 85% in the offline analysis; the average accuracy was higher than 94%, and the average accuracy was 85.93% after an average of 5.37 rounds of button flashes in the simulated online test. These results showed that, with the help of a big training dataset, the cross-subject and cross-amplifier CNN classification models could be established for subject-independent P300 BCIs that do not require subject-specific calibration for a large percentage of users, which will substantially improve P300 BCI applications.

The remainder of this paper is organized as follows. Section II presents the methods, including those for data acquisition, data preprocessing, the CNN for P300 detection, decision making, and the simulated online test. The experimental results are presented in Section III, and a discussion is provided in Section IV. Finally, the conclusions in Section V review the approach outlined in this paper.

II. Methods

A. Equipment

During the experiment, EEG data were collected at a sampling rate of 250 Hz, with a 30-channel EEG cap (LT 37) that followed the extended 10-20 system (see Fig. 1), and signals were referenced to the right mastoid. A 64-channel amplifier

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2021.3083548, IEEE Transactions on Neural Systems and Rehabilitation Engineering

SynAmps2 (Compumedics, Neuroscan, Inc., Australia) and a 40-channel amplifier NuAmps (Compumedics, Neuroscan, Inc., Australia) were used to collect EEG signals for every subject. All electrode impedances were maintained below 5 kΩ during data collection, as determined by dynamically monitoring the impedance values displayed on the screen of the connected computer.

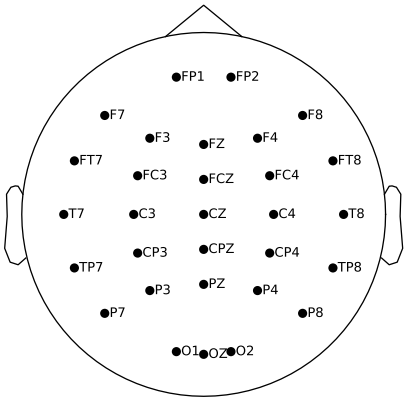

Fig. 1. The positions of the 30 channels.

B. Graphical User Interface

This study employs a P300 speller for data collection with the graphical user interface (GUI) shown in Fig. 2 [32]. Each subject was presented with a 4×10 button matrix of characters. In each trial, the subject selectively focused on one of the 40 buttons, which were intensified according to the paradigm shown in Fig. 3, to input a character. Specifically, to prepare the subject, during the 3 seconds before stimulus onset, the buttons were not intensified. Upon onset, all 40 buttons start to flash successively in a random order. Here, we made the buttons flash in a random order rather than in a fixed order to enhance the oddball effect so that the P300 component can be detected when the target button flashes. Each flash lasted for 100 ms, and the interval between the onsets of two successive flashes was 30 ms, which means that an overlap of 70 ms existed between any pair of successive flashes. A round consisted of 40 button flashes (one button corresponds to one flash per round), and 10 rounds comprised a trial. Furthermore, no pause was used between adjacent rounds. Thus, it took a total of $(400-1)\times30+100=12,070$ ms to complete the 400 flashes.

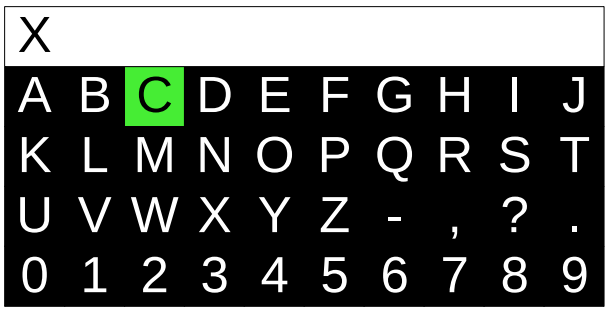

Fig. 2. Graphical user interface of the P300 speller.

C. Data Acquisition and Dataset Partitions

Two hundred healthy subjects (168 males and 32 females, between 18 and 32 years of age) participated in our experiment. The study was approved by the Ethics Committee of Sichuan Provincial Rehabilitation Hospital, China (the cooperative institution). Written informed consent was obtained from all subjects. During the experiment, the subjects were instructed to focus their attention on the flashes of specified characters (i.e., targets) in each trial and to keep a running mental count of the number of the flashes while their EEG signals were recorded. Each subject performed 120 trials, with each trial corresponding to a character input using the two amplifiers, as mentioned in Section II-A (60 trials per amplifier). To avoid subject fatigue caused by long-term experiments, which would affect data quality, for each subject, the experiment containing 120 trials was divided into 4 sessions, each of which contained 30 trials. Between any pair of successive sessions, the subject had rest time. Only when the subject was in a good condition would the next session be started. Note that the sequence of 60 characters for each subject and each amplifier was randomly produced before the experiment began.

The collected EEG data were used in the following cross-subject data analysis, including both the offline analysis and the simulated online test. The 200 subjects were divided into two groups: 150 subjects for training and validation and the other 50 subjects for an independent test. In the remainder of this paper, we denote the subsets of the dataset using the symbols shown in Table I. Each symbol, e.g., TrV-S-150, includes three parts. The first part indicates the dataset: "TrV" refers to the training and validation set, while "Te" refers to the independent test set. The second part indicates the amplifier: "S" and "N" refer to the SynAmps2 amplifier and the NuAmps amplifier, respectively, and "SN" refers to both amplifiers. The third part, "n", refers to the number of subjects in the subset. For example, "TrV-SN-149" refers to the data of 149 of the 150 subjects in the training and validation set collected by both amplifiers.

D. Data Preprocessing

The recorded 30-channel EEG signals were first bandpass filtered at 0.5–10 Hz using a fourth-order Butterworth filter, which is an infinite impulse response (IIR) filter. Filtering was conducted using MNE, an open-source Python software for analyzing human neurophysiological data [33], [34]. Notably, despite the randomness, the time sequence of flash onsets for each of the 40 buttons can still be recorded during data collection, and we can use these recorded time sequences to extract EEG epochs for different buttons for subsequent data analysis. Specifically, epochs corresponding to each flash from 0 to 600 ms after stimulus onset were extracted. Therefore, there were 150 (600 ms $\times$ 250 Hz) sampling points for each channel per epoch, and there were 400 epochs per trial. All epochs were baseline corrected using the 200 ms before the onset of the stimulus as a baseline and then downsampled at a rate of 6. Last, the signals of each epoch were normalized as follows [24]:

$$\tilde{x}_{i,j}=\frac{x_{i,j}-\bar{x}_i}{\sigma_i}, \tag{1}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2021.3083548, IEEE Transactions on Neural Systems and Rehabilitation Engineering

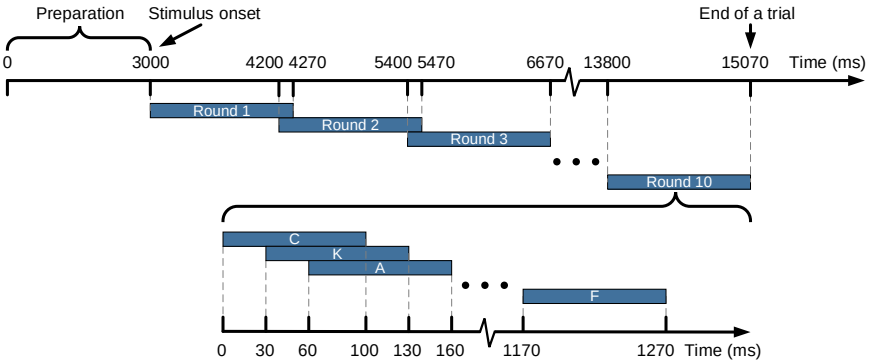

Fig. 3. Illustration of the experimental paradigm. For the input of a single character in a trial, to prepare the subject, the buttons are not initially intensified. Ten rounds were included for each trial, and each round consisted of 40 button flashes. Upon onset, all 40 buttons start to flash successively in a random order. Each flash lasts for 100 ms, and the interval between the onsets of two successive flashes is 30 ms, which means that there is an overlap of 70 ms between any pair of successive flashes. Thus, it takes a total of 12.07 s to complete the 400 flashes in a trial. Note that the flash sequences of different rounds and different trials are different, and the sequence "C, K, A, ..., F" shown in this figure is just a possible sequence.

TABLE I
PARTITIONS OF THE DATASET

Amplifier	Training and validation set (150 subjects in total)			Independent test set (50 subjects)		
	Symbol	Number of subjects ($n \leqslant 150$)	Number of trials	Symbol	Number of subjects	Number of trials
SynAmps2	TrV-S-n	n	$60n$	Te-S-50	50	60×50
NuAmps	TrV-N-n	n	$60n$	Te-N-50	50	60×50
Both amplifiers	TrV-SN-n	n	$120n$	Te-SN-50	50	120×50

where $x_{i,j}$ and $\tilde{x}_{i,j}$ are the unnormalized and normalized signals of channel i at sampling point j, respectively, and $\bar{x}_i$ and σ_i are the average and standard deviation of the signal of channel i in the epoch, respectively. This normalization is meant to avoid undesirable biases to facilitate signal classification.

After preprocessing, the data of each epoch formed a 30×25 matrix, which was denoted $\mathbf{X}_{n_\mathrm{s},n_\mathrm{t},n_\mathrm{r},n_\mathrm{c}}$, where n_s represents the subject (ranging from 1 to N_s), n_t represents the current trial (ranging from 1 to N_t), n_r represents the current flash round (ranging from 1 to N_r), and n_c represents the corresponding character (ranging from 1 to N_c). Herein, $N_\mathrm{r} = 10$, and $N_\mathrm{c} = 40$. The preprocessed signals corresponding to the first n_r0 $(n_\mathrm{r0} = 1, 2, \ldots, N_\mathrm{r})$ repeats of each character in a trial were averaged as follows:

$$\bar{\mathbf{X}}_{n_\mathrm{s},n_\mathrm{t},n_\mathrm{r0},n_\mathrm{c}} = \frac{1}{n_\mathrm{r0}} \sum_{n_\mathrm{r}=1}^{n_\mathrm{r0}} \mathbf{X}_{n_\mathrm{s},n_\mathrm{t},n_\mathrm{r},n_\mathrm{c}}. \tag{2}$$

In the training phase, only $\bar{\mathbf{X}}_{n_\mathrm{s},n_\mathrm{t},N_\mathrm{r},n_\mathrm{c}}$ $(n_\mathrm{c} = 1, 2, \ldots, N_\mathrm{c})$ were fed into the deep neural network, while in the prediction/test phase, n_r0 was set to different values to show the change in performance with respect to the number of flash rounds.

A sample was labeled as a positive sample if and only if the corresponding character was the target of the current trial. Before being fed into the network, the preprocessed training data underwent another normalization process: the conventional zero-mean unit-variance normalization widely used in CNN training to improve convergence [35]. In the prediction/test phase, the test samples were also normalized using the mean and standard deviation of the training set.

E. CNN Architecture

As shown in Fig. 4, the model used for cross-subject P300 detection was a CNN, which is similar to the network used in [24]. It contains 3 convolutional layers (denoted as "C") and 2 fully connected layers (denoted as "FC"). Each input sample is a 30×25 matrix. The first convolutional layer filters the preprocessed signal with 10 kernels sized 30×1 and a stride of 1, resulting in 10 feature maps sized 1×25. Each map of this layer plays a role in channel combination. The second and third convolutional layers both take the outputs of the previous layers as inputs and filter them with 10 kernels sized $1 \times k_2$ and $1 \times k_3$, respectively, and a stride of 1. Therefore, the output of the third convolutional layer contains 10 feature maps sized $1 \times (27 - k_2 - k_3)$. These two layers play a role in temporal filtering. The last 2 layers are both fully connected layers with k_4 neurons and 1 neuron, respectively. Here, we set the lengths of the kernels k_2 and k_3 and the number of neurons k_4 as variables that will be determined later. All layers except FC5 use the rectified linear unit (ReLU) as the activation function. The output of the last fully connected layer, FC5, is passed to the sigmoid function $S(x) = (1 + \exp(-x))^{-1}$. For the P300 detection problem, samples with the presence of P300 are regarded as positive samples, while samples with the absence of P300 are regarded as negative samples. If the labels assigned to the negative and positive samples are 0 and 1, respectively, then the output of the sigmoid function can be regarded as the modeled probability of the presence of P300 $P(y = 1 \mid \mathbf{X})$,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2021.3083548, IEEE Transactions on Neural Systems and Rehabilitation Engineering

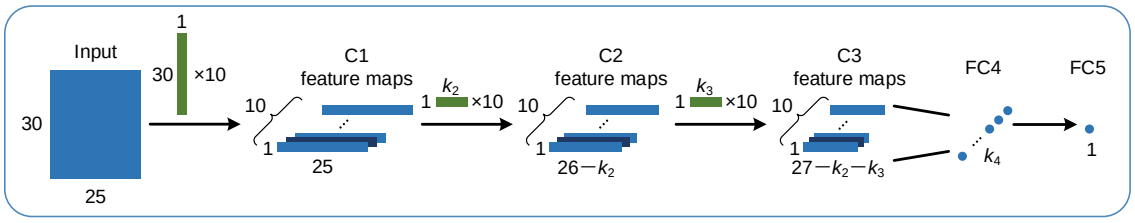

Fig. 4. The architecture of the CNN used for cross-subject P300 detection.

where $\mathbf{X}$ is the input of the network. The binary label $y = 1$ represents the presence of P300, while $y = 0$ represents the absence of P300.

The total number of the trainable parameters of the convolutional kernels is $(30 + 1) \times 10 + (10k_2 + 1) \times 10 + (10k_3 + 1) \times 10 = 100k_2 + 100k_3 + 330$, and that of the fully connected layers is $[10(27 - k_2 - k_3) + 1]\, k_4 + (k_4 + 1) = 272k_4 - 10k_2k_4 - 10k_3k_4 + 1$. Here, 1's are added to represent the number of biases. The kernels of the convolutional layers and the weights of the fully connected layers are regularized with ℓ_2 regularization constraints.

The choices of the hyperparameters associated with the network structure can be explained as follows. We first set the kernel size for the convolutional layer C1 (associated with the channel dimension) to 30×1 because this layer played a role in combining all of the 30 channels. Next, the kernel lengths k_2 and k_3 for convolutional layers C2 and C3 (associated with the time dimension) and the number of neurons k_4 for the fully connected layer FC4 were determined using a grid search strategy. Specifically, we considered a number of candidates for these hyperparameters. Note that the kernel lengths k_2 and k_3 physically determined the temporal window sizes used for feature extraction, where the input data had 25 sample points along the time dimension. We set several kernel lengths $2, 4, \ldots, 10$ as candidates for k_2 and k_3. We also set several numbers of neurons of the fully connected layer FC4 (i.e., $5, 10, \ldots, 25$) as candidates. We then evaluated these hyperparameters experimentally. The data of 10 randomly selected subjects from a training and validation set were used for validation, and the data of the remaining 140 subjects were used for training. The hyperparameters with the highest average accuracy over the 10 validation subjects were chosen as the optimal hyperparameters. For example, the hyperparameters k_2, k_3, and k_4 were chosen as 8, 10, and 15, respectively, for the training and validation set TrV-SN-150.

F. Model Training and Testing

The convolutional kernels and weights of the network were initialized with the Xavier initialization method [36]. The model was trained using Adam [37], a method for stochastic optimization, to optimize the mean-square error (MSE). Since the dataset is imbalanced, with a ratio of positive to negative samples of 1:39, the loss function was weighted by multiplying positive samples by 39 [26]. The model was trained on an NVIDIA GeForce GTX 1080 Ti GPU, with CUDA 9.0 and cuDNN v7, using TensorFlow, an open-source machine learning framework [38].

For the cross-subject and within-amplifier/cross-amplifier independent tests, the models were respectively trained on three training sets (i.e., TrV-S-150, TrV-N-150, and TrV-SN-150) containing data from 150 subjects, and they were tested on two independent test sets (i.e., Te-S-50 and Te-N-50) containing data from the other 50 subjects. In this paper, we adopted accuracy, defined as the ratio of the number of correctly input characters to the total number of input characters, as the performance metric.

G. Decision Making

Because the SNR of EEG is quite low and the interstimulus interval (ISI) was short in our setting, in the prediction phase, decisions were usually made based on several rounds of flashes. The preprocessed signals were averaged as in Eq. (2) and then sent to the deep neural network as the input. As described in Section II-E, the output of the sigmoid function is regarded as the probability of the presence of P300. Hence, for each trial, after n_r rounds of button flashes, the decision was made as follows:

$$d\left(n_\mathrm{s}, n_\mathrm{t}, n_\mathrm{r}\right) = \underset{n_\mathrm{c} \in \{1,2,\ldots,N_\mathrm{c}\}}{\operatorname{argmax}} P\left(y = 1 \mid \bar{\mathbf{X}}_{n_\mathrm{s},n_\mathrm{t},n_\mathrm{r},n_\mathrm{c}}\right). \quad (3)$$

H. Simulated Online Test

In the simulated online test, the number of rounds of button flashes for each trial is adaptive according to the confidence of making a decision. In this way, the system achieves a better balance between accuracy and spelling speed. For an adaptive online system, it is essential to find a criterion for determining when to make decisions. As described in Section II-E, the output value corresponding to the input $\bar{\mathbf{X}}_{n_\mathrm{s},n_\mathrm{t},n_\mathrm{r},n_\mathrm{c}}$ is regarded as the probability of the presence of P300 $P\left(y = 1 \mid \bar{\mathbf{X}}_{n_\mathrm{s},n_\mathrm{t},n_\mathrm{r},n_\mathrm{c}}\right)$, which indicates the confidence of making a decision. This characteristic raises the following questions: Is this probability relevant to the accuracy, and can it be further used to determine whether to make a decision?

We explored the relation between the probability of the presence of P300 and accuracy. Given subject n_s in trial n_t after n_r flash rounds, we calculated the maximum probability:

$$\begin{aligned} Q\left(n_\mathrm{s}, n_\mathrm{t}, n_\mathrm{r}\right) &:= \max_{n_\mathrm{c} \in \{1,2,\ldots,N_\mathrm{c}\}} P\left(y = 1 \mid \bar{\mathbf{X}}_{n_\mathrm{s},n_\mathrm{t},n_\mathrm{r},n_\mathrm{c}}\right) \\ &= P\left(y = 1 \mid \bar{\mathbf{X}}_{n_\mathrm{s},n_\mathrm{t},n_\mathrm{r},d(n_\mathrm{s},n_\mathrm{t},n_\mathrm{r})}\right), \end{aligned} \quad (4)$$

which indicates the confidence of making a decision for subject n_s in trial n_t after n_r flash rounds. Then, we averaged

the maximum probabilities over all trials for each subject, denoted as $C(n_{\mathrm{s}}, n_{\mathrm{r}})$, to simplify the notation:

$$C(n_{\mathrm{s}}, n_{\mathrm{r}}) \coloneqq \bar{Q}(n_{\mathrm{s}}, n_{\mathrm{r}}) = \frac{1}{N_{\mathrm{t}}} \sum_{n_{\mathrm{t}}=1}^{N_{\mathrm{t}}} Q(n_{\mathrm{s}}, n_{\mathrm{t}}, n_{\mathrm{r}}). \quad (5)$$

We denote the accuracy (%) for subject n_{s} after n_{r} flash rounds as $a(n_{\mathrm{s}}, n_{\mathrm{r}})$. As will be presented in Appendix D, we found a positive correlation between $C(n_{\mathrm{s}}, n_{\mathrm{r}})$ and $a(n_{\mathrm{s}}, n_{\mathrm{r}})$. This finding inspired us to use the output of the network to determine whether to make a decision and automatically stop the current trial in each flash round in the online test.

We first set the minimum and maximum numbers of flash rounds for making a decision, i.e., N_{rmin}, N_{rmax}, and a threshold of the probability $Q_{\mathrm{th}}(n_{\mathrm{r}}) \in [0,\ 1]$ for each flash round, i.e., $n_{\mathrm{r}} \in \{N_{\mathrm{rmin}}, \ldots, N_{\mathrm{rmax}} - 1\}$. For subject n_{s} in trial n_{t}, once $n_{\mathrm{r}} \geqslant N_{\mathrm{rmin}}$, $Q(n_{\mathrm{s}}, n_{\mathrm{t}}, n_{\mathrm{r}})$ is calculated according to Eq. (4) and then compared with $Q_{\mathrm{th}}(n_{\mathrm{r}})$. If $Q(n_{\mathrm{s}}, n_{\mathrm{t}}, n_{\mathrm{r}}) \geqslant Q_{\mathrm{th}}(n_{\mathrm{r}})$, then the system outputs $d(n_{\mathrm{s}}, n_{\mathrm{t}}, n_{\mathrm{r}})$ as the prediction. Otherwise, the next round of button flashes progresses until $Q(n_{\mathrm{s}}, n_{\mathrm{t}}, n_{\mathrm{r}}) \geqslant Q_{\mathrm{th}}(n_{\mathrm{r}})$ or $n_{\mathrm{r}} \geqslant N_{\mathrm{rmax}}$. If n_{r} reaches N_{rmax} and $Q(n_{\mathrm{s}}, n_{\mathrm{t}}, n_{\mathrm{r}})$ still has not reached the threshold $Q_{\mathrm{th}}(n_{\mathrm{r}})$, then the system outputs $d(n_{\mathrm{s}}, n_{\mathrm{t}}, N_{\mathrm{rmax}})$ as the prediction.

III. Results

A. Results of the Cross-Subject and Within-Amplifier/Cross-Amplifier Offline Analysis

As mentioned in Section II-D, the training sets only contained data averaged over 10 flash rounds, while the number of flash rounds for the test data was set to different values to show the change in performance. The average accuracies with standard deviations with respect to the number of flash rounds for the independent test set are shown in Table II. Incidentally, the first four columns of the results are based on offline analyses in which the training set in each column contains data collected by only one amplifier, while the fifth and sixth columns are based on offline analyses in which the training sets contain data collected by both amplifiers. The last column presents the averages and standard deviations of the average accuracies over the 6 training and test set situations. Among the first four columns, the first and fourth columns are based on the offline analysis in which the training set and the independent test set were collected by the same amplifier, while the second and third columns show the results of the cross-amplifier offline analyses. If the number of flash rounds was small, then the accuracies were low, typically only approximately 20% after one flash round, due to the low SNR of EEG and the short ISI of the stimulation system. Adding rounds of button flashes improved the average accuracies in all 6 training and test sets. After 10 rounds of button flashes, the average accuracies reached above 80%. Specifically, compared with situations in which training sets contain data collected by only one amplifier, training on data collected by both amplifiers achieved better results, with accuracies of 89.19% and 81.12% on the independent test sets collected by the NuAmps and SynAmps2 amplifiers, respectively. This result might due to the larger size of the training set containing data from both amplifiers. Surprisingly, when the training set contained data collected by both amplifiers and the independent test set contained data collected by only the NuAmps amplifier, the average accuracy reached 89.19% after 10 flash rounds. The 2nd and 3rd columns indicated that our method was suitable for learning on cross-amplifier data.

In addition to the independent test, we also investigated the cross-subject and within-amplifier/cross-amplifier performance of our method on the training and validation sets TrV-S-150, TrV-N-150 and TrV-SN-150 using leave-one-subject-out cross-validation. The results are shown in Table V in Appendix A.

B. Results of the Simulated Online Test

We conducted a simulated online test with the collected data. The parameter settings in Section II-H were as follows. The minimum and maximum numbers of flash rounds for making a decision were set as $N_{\mathrm{rmin}} = 4$ and $N_{\mathrm{rmax}} = 10$. We did not consider $n_{\mathrm{r}} < 4$ because we empirically found that the performance was unacceptably low in this case, and $n_{\mathrm{r}} > 10$ was not considered because of the low speed. To set the threshold $Q_{\mathrm{th}}(n_{\mathrm{r}})$, we mainly considered the reliability of the decision, i.e., we used a higher threshold to guarantee confident decisions for cases with few flash rounds, and we used a lower threshold for cases with more flash rounds since having more rounds of flashes could improve the reliability of the decision. The value of $Q_{\mathrm{th}}(n_{\mathrm{r}})$ was selected by referring to the results of leave-one-subject-out cross-validation on the training and validation sets, i.e., the minimum values of the top 5% $C(n_{\mathrm{s}}, n_{\mathrm{r}})\big|_{n_{\mathrm{r}}=4}$ and 20% $C(n_{\mathrm{s}}, n_{\mathrm{r}})\big|_{n_{\mathrm{r}}=7}$ were set as the thresholds for $n_{\mathrm{r}} = 4, 5, 6$ and $n_{\mathrm{r}} = 7, 8, 9$, respectively. Here, $C(n_{\mathrm{s}}, n_{\mathrm{r}})$ is the average maximum probability of subject n_{s} after n_{r} flash rounds, averaged over all trials, as mentioned in Section II-H. For example, for training set TrV-SN-150, when $n_{\mathrm{r}} = 4$, 5% of the subjects in the training and validation set had C values higher than 0.992, and when $n_{\mathrm{r}} = 7$, 20% of the subjects had C values higher than 0.979. Thus, in this situation, we set $Q_{\mathrm{th}}(n_{\mathrm{r}}) = 0.992$ for $n_{\mathrm{r}} = 4, 5, 6$, and $Q_{\mathrm{th}}(n_{\mathrm{r}}) = 0.979$ for $n_{\mathrm{r}} = 7, 8, 9$. Note that no threshold was needed for $n_{\mathrm{r}} = 10$ since a decision must be made regardless of the probability value $Q(n_{\mathrm{s}}, n_{\mathrm{t}}, n_{\mathrm{r}})$ (see Section II-H).

We calculated the average accuracies and number of flash rounds for the independent test set, and the results are presented in Table III. In all training and independent test set situations, the average accuracies were above 70% or even above 80% after approximately 6 flash rounds. Specifically, when the training set and the independent test set were both collected by the NuAmps amplifier, the 50 subjects achieved an average accuracy of 82.33% in an average of 6.07 flash rounds.

C. Comparison with Other Methods

To provide a comparison, we applied two linear methods, i.e., SVM (based on LIBLINEAR, a library for large linear classification [39]) and LDA to our big data and performed data analysis similar to that performed for our CNN. The corresponding results are shown in the first three columns of Table IV, where each average accuracy with the standard

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2021.3083548, IEEE Transactions on Neural Systems and Rehabilitation Engineering

TABLE II
AVERAGE ACCURACIES WITH STANDARD DEVIATIONS (%) WITH RESPECT TO THE NUMBER OF FLASH ROUNDS FOR THE INDEPENDENT TEST SET

	Training set/ Independent test set	TrV-S-150/ Te-S-50	TrV-S-150/ Te-N-50	TrV-N-150/ Te-S-50	TrV-N-150/ Te-N-50	TrV-SN-150/ Te-S-50	TrV-SN-150/ Te-N-50	Average
Number of flash rounds	1	19.44 ± 10.62	24.50 ± 12.30	19.29 ± 12.29	26.36 ± 11.19	18.10 ± 11.11	26.40 ± 12.62	22.35 ± 3.49
	2	34.94 ± 16.44	41.95 ± 18.64	35.99 ± 17.11	45.44 ± 17.79	35.56 ± 15.24	45.41 ± 18.02	39.88 ± 4.55
	3	46.17 ± 17.59	53.17 ± 21.02	47.28 ± 18.48	56.58 ± 19.00	47.74 ± 16.05	57.86 ± 19.71	51.47 ± 4.64
	4	55.34 ± 19.22	62.28 ± 20.83	55.64 ± 19.76	65.21 ± 18.79	56.29 ± 18.01	66.11 ± 19.08	60.14 ± 4.55
	5	61.89 ± 19.35	68.97 ± 19.69	62.52 ± 18.81	72.87 ± 17.33	62.29 ± 18.47	72.46 ± 17.45	66.83 ± 4.77
	6	67.50 ± 19.68	73.22 ± 18.73	67.96 ± 18.78	76.98 ± 16.90	68.56 ± 17.77	77.87 ± 17.08	72.02 ± 4.27
	7	71.44 ± 19.78	77.12 ± 17.96	71.28 ± 18.40	80.67 ± 15.81	72.42 ± 18.14	81.43 ± 15.78	75.73 ± 4.24
	8	73.95 ± 20.37	79.64 ± 17.79	73.60 ± 17.86	84.47 ± 13.62	75.73 ± 16.93	84.40 ± 14.60	78.63 ± 4.55
	9	76.88 ± 18.40	82.81 ± 16.22	76.33 ± 17.30	86.78 ± 12.68	78.82 ± 15.45	87.64 ± 13.27	81.54 ± 4.52
	10	79.02 ± 17.45	85.43 ± 15.06	79.22 ± 16.07	88.44 ± 12.00	81.12 ± 14.93	89.19 ± 12.10	83.74 ± 4.17

TABLE III
RESULTS OF THE SIMULATED ONLINE TEST FOR THE INDEPENDENT TEST SET

Training set/ Independent test set	TrV-S-150/ Te-S-50	TrV-S-150/ Te-N-50	TrV-N-150/ Te-S-50	TrV-N-150/ Te-N-50	TrV-SN-150/ Te-S-50	TrV-SN-150/ Te-N-50	Average
Average accuracy (%)	70.61 ± 17.20	77.40 ± 15.90	70.91 ± 16.20	82.33 ± 13.01	70.01 ± 15.62	80.77 ± 14.28	75.34 ± 5.05
Average number of flash rounds	6.19 ± 1.30	5.94 ± 1.33	6.40 ± 1.33	6.07 ± 1.35	5.94 ± 1.22	5.73 ± 1.26	6.04 ± 0.21

TABLE IV
AVERAGE ACCURACIES WITH STANDARD DEVIATIONS (%) WITH RESPECT TO THE NUMBER OF FLASH ROUNDS OBTAINED WITH DIFFERENT METHODS, AVERAGED OVER THE 6 TRAINING AND INDEPENDENT TEST SETTINGS, AS IN TABLE II

	Model	SVM	LDA	CNN	LimitedCal			Pretrain+FT
					SVM	LDA	CNN	CNN
Number of flash rounds	1	22.37 ± 5.72	22.34 ± 6.27	22.35 ± 3.49	16.50 ± 1.61	8.54 ± 0.63	12.21 ± 1.12	29.92 ± 0.90
	2	38.32 ± 7.69	37.98 ± 8.84	39.88 ± 4.55	27.20 ± 2.50	13.10 ± 1.07	19.30 ± 1.60	48.64 ± 1.68
	3	50.35 ± 8.71	49.97 ± 9.79	51.47 ± 4.64	36.76 ± 2.48	17.58 ± 1.30	25.30 ± 1.66	61.07 ± 1.87
	4	58.34 ± 8.95	57.73 ± 10.61	60.14 ± 4.55	44.76 ± 2.78	21.83 ± 1.65	30.24 ± 1.69	69.83 ± 1.89
	5	65.08 ± 8.18	64.05 ± 10.29	66.83 ± 4.77	51.37 ± 2.87	25.89 ± 1.71	34.38 ± 1.78	76.27 ± 1.88
	6	70.22 ± 7.38	69.22 ± 9.37	72.02 ± 4.27	56.72 ± 2.65	29.73 ± 1.72	38.16 ± 2.02	80.75 ± 1.66
	7	74.14 ± 6.97	72.74 ± 8.87	75.73 ± 4.24	61.08 ± 2.62	33.16 ± 1.80	41.32 ± 1.82	84.34 ± 1.75
	8	77.81 ± 6.33	76.77 ± 8.13	78.63 ± 4.55	65.00 ± 2.45	36.34 ± 2.02	44.37 ± 1.86	87.00 ± 1.67
	9	80.41 ± 6.00	79.19 ± 7.85	81.54 ± 4.52	68.26 ± 2.35	39.46 ± 1.95	47.11 ± 1.66	89.15 ± 1.54
	10	82.42 ± 5.36	81.46 ± 7.32	83.74 ± 4.17	71.20 ± 2.47	42.20 ± 1.73	49.69 ± 1.64	90.82 ± 1.27

deviation was obtained by averaging the accuracies across the 6 training and test settings, as in Table II. We can see from the table that these methods had comparable performances for cross-subject and within-amplifier/cross-amplifier tests for our data. The performances of another CNN model, EEGNet [26], were also evaluated, and an average accuracy of 82.48% was achieved after 10 rounds of button flashes, which is also comparable to that of our proposed CNN model.

While our CNN-based method showed comparable performance to traditional methods like SVM and LDA as above, CNN-based methods have two merits: (1) They are more scalable than traditional methods. Specifically, if the dataset is too large (e.g., much larger than the dataset in this study) or has an extremely high number of dimensions, then SVM and LDA are not applicable due to the involvement of solving a large-scale optimization problem or inverting a very large matrix; (2) Their performances can be remarkably boosted by using just a very limited amount of calibration data to fine-tune the pretrained model, while this is infeasible for traditional methods.

To verify the second point above, we further carry out another comparative experiment. Concretely, we compare with the following two settings on the independent test sets:

LimitedCal: The models were trained from scratch with a small amount (5 trials in this study) of calibration data collected from the test subject and then evaluated with the

remaining data.

Pretrain+FT: The models trained with data from the training and validation sets TrV-S-150, TrV-N-150 and TrV-SN-150, whose performances had been evaluated in Section III-A, were adopted as pretrained models. Then, for each test subject, a small amount (5 trials in this study) of data was used as calibration data to fine-tune the pretrained models, and the remaining data were used to evaluate the retrained models.

We studied the method LimitedCal with 3 models (i.e., SVM, LDA, and the proposed CNN), while for the method Pretrain+FT, only the CNN was studied since the other two models are inapplicable to this method. For each subject and each amplifier, we equally partitioned the 60 trials into 12 folds, each of which contained data of 5 trials, and evaluated the above two methods using the cross-validation strategy. Each time, 1 fold was used for training/fine-tuning, while the other 11 folds were used to evaluate the model. Note that because we wanted to train/fine-tune the model with as little data as possible, we took only 1 fold instead of 11 folds for training/fine-tuning. This procedure was repeated 12 times, and the accuracies of these 12 times were averaged.

Table IV presents the average accuracies with standard deviations for these methods. Using the method LimitedCal, we found that when trained from scratch with data of only 5 trials, which took approximately 75 seconds for data collection, neither the LDA nor CNN model could achieve acceptable results, while the SVM model achieved an average accuracy above 70% after 10 flash rounds, showing its powerful data fitting ability. However, these results were still unsatisfactory in practice and were much worse than those of our zero-training method, which are shown in the third column. In regard to the method Pretrain+FT, although the models were retrained with data of only 5 trials, the average accuracy increased significantly from 83.74% to 90.82%.

IV. Discussion

In this study, we proposed a CNN and big EEG data-based learning method for a subject-independent P300 BCI speller system, attempting to realize a P300 speller without requiring subject-specific calibration. The experimental results demonstrate the effectiveness of our method for learning invariant patterns across subjects and across amplifiers.

The good performance of the learning method is attributed to the excellent ability of the deep neural network for data fitting and the large size of our dataset, which includes data from a relatively large number of subjects compared with existing work. These two factors provided the model with the possibility of extracting subject-independent features. A question may arise regarding whether larger training sets will always result in better performance. We explored the influence of the size of the training set on the cross-subject and cross-amplifier classification accuracy and found that when the size of the training set was relatively small, containing data from a relatively small number of subjects, the accuracy increased as the number of subjects in the training set increased. After the number of subjects reached a certain level, increasing the number of subjects had little effect on the accuracy. For more detailed results, readers can refer to Appendix C. However, enlarging the dataset to a higher order of magnitude (e.g., involving data of 2,000 subjects) may be good practice to explore the impact of big data on the results.

Our results showed that the standard deviations of the accuracies over all subjects were large, reaching above 10% and even above 20% in some cases. This result indicates that the performance of cross-subject classification varied greatly by subject. This inference is also confirmed in our research on the distributions of accuracies for different subjects, which is presented in Appendix B. In fact, a percentage of subjects had very high performances. Averaging the ratios over the six training and test set situations, we found that 66.00% of the subjects for the independent test set achieved accuracies above 80%. Among them, 56.66% of the 50 subjects achieved accuracies above 85%. However, some subjects achieved poor accuracies. Specifically, 4.33% of the 50 subjects in the independent test set achieved accuracies below 50%. The event-related potential (ERP) waveforms (without downsampling) from the Cz and Oz channels are presented in Fig. 5 and Fig. 6 for five subjects who had excellent and five subjects who had bad performances, respectively. For each stimulus type (target or nontarget), the ERP waveforms of each subject were extracted by taking the time-locked average of the EEGs across 60 trials and 10 flash rounds per trial. Note that the nontarget waveforms, which inherently have no dominating P300 patterns, appear to be flat lines due to the averaging operation. The waveforms of the subjects with good performances, which are shown in Fig. 5, contain obvious features of typical ERP components, such as N200 and P300. In contrast, as shown in Fig. 6, the subjects with bad performances did not exhibit typical features in their brain patterns. For example, for channel Oz of the two amplifiers, the waveforms of Subjects 6–10 did not contain typical P300 components. Furthermore, for some subjects, the ERP components had relatively large offsets in terms of time (e.g., Subjects 6, 7, and 10 in Fig. 6). It is worth mentioning that some subjects who had poor performances in our cross-subject analyses may achieve high accuracies after a subject-specific calibration process since their waveforms for target and nontarget stimuli may still be distinguishable. For instance, for each of the above five subjects with poor performance, we performed 5-fold cross-validation using his/her data, and a subject-specific SVM model was trained in each fold. These five subjects achieved fold-averaged accuracies of 95.00%, 100.00%, 82.50%, 95.00% and 100.00%.

From Table IV, we find that when the calibration data of the test subject are extremely limited, instead of using these limited data to train the model from scratch, it is better to use a model trained with a dataset collected from a large number of other subjects, even if the calibration phase is completely eliminated. Adding a short-time calibration phase and fine-tuning the pretrained model will further improve the accuracy.

The results of fine-tuning in Table IV demonstrate that fine-tuning can help enhance the cross-subject and cross-amplifier performances significantly. After fine-tuning with data of only 5 trials, in all 6 training and independent test set situations, none of the subjects in the independent test set had an accura-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2021.3083548, IEEE Transactions on Neural Systems and Rehabilitation Engineering

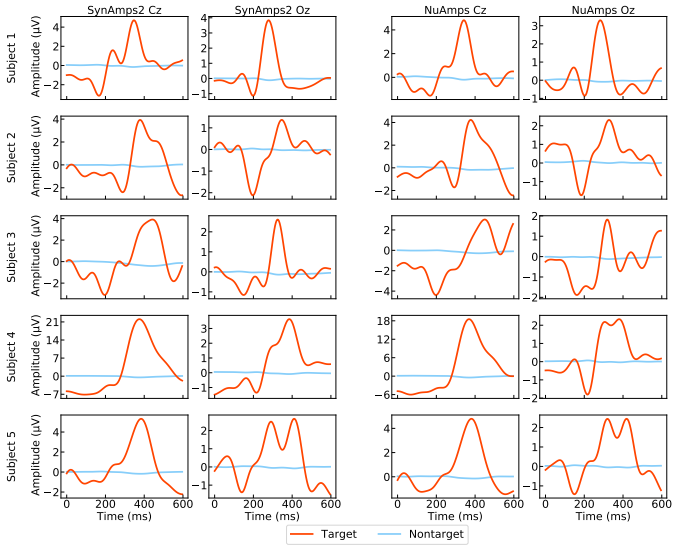

Fig. 5. ERP waveforms for five subjects (denoted as Subjects 1–5 here) with good performances in cross-subject learning. Each row corresponds to the waveforms of one subject, and the left and right sides correspond to the amplifier SynAmps2 and the amplifier NuAmps, respectively. On the left/right side, the first and second columns are signals collected from the Cz and Oz channels, respectively.

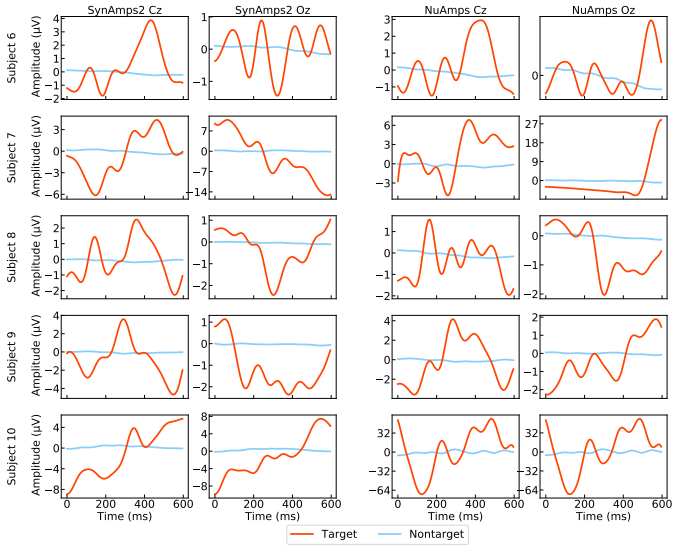

Fig. 6. ERP waveforms for five subjects (denoted as Subjects 6–10 here) with bad performances in cross-subject learning. Each row corresponds to the waveforms of one subject, and the left and right sides correspond to the amplifier SynAmps2 and the amplifier NuAmps, respectively. On the left/right side, the first and second columns are signals collected from the Cz and Oz channels, respectively.

cy below 50%. In between fully-calibrated and calibration-free, the short-time fine-tuning strategy is practically very meaningful, which can obtain large performance gain without sacrificing much of the user convenience. Such fine-tuning is widely used in transfer learning related methods, which is also a merit of CNN-based methods like ours relative to traditional methods.

From the results of the simulated online test presented in Table III, all 50 subjects in the group for the independent test set had an average accuracy of 75.34% after 6.04 flash rounds. Furthermore, we investigated the simulated online performances of the subjects who had cross-subject accuracies above 85% in the offline analysis, and the results are shown in Table VII in Appendix B. These subjects accounted for 56.66% of the total number of subjects and achieved an average accuracy of 85.93% after an average of 5.37 rounds of button flashes in the simulated online test. Such performance is nearly equivalent to that of subject-specific calibration systems, according to our experience. Thus, it is appropriate to build an online subject-independent system for this portion of subjects. In the offline analyses, the average time needed for forward propagation through the CNN was 1.683 ms per trial (10 flash rounds). Therefore, it is feasible to modify this offline system as a real-time online system. That is, a CNN model can be trained in advance with data from a large number of subjects and then applied to new users.

Whether a new user is suitable for cross-subject learning can be determined by using one of the following two methods. First, the new user should attempt to use the online system without subject-specific calibration. If possible, feedback of the user's accuracy can be directly obtained to evaluate the cross-subject learning performance of the subject. Otherwise, since we found a positive correlation between the maximum probability of the presence of P300 and the accuracy, which is mentioned in Section II-H and detailed in Appendix D, whether the user is likely to achieve a good performance can be estimated based on the maximum probability of the presence of P300. If the user can achieve high accuracy in the cross-subject test, then he/she can continue using the system without subject-specific calibration. Otherwise, the user should perform the subject-specific calibration task, and the system will use the data collected in the calibration phase to fine-tune a subject-specific model.

Compared with existing deep learning-based transfer learning methods for BCI, our method has the following advantages. First, the dataset used in our study is large, including data of 200 subjects, which is approximately 5–10 times more data than that in most existing studies [25], [40]. The large number of subjects provides the possibility of extracting subject-independent ERP features and makes the results more statistically universal. Second, a small amount of data from the test subjects is generally used to fine-tune the model in existing studies on transfer learning. However, we conducted a test in which no data of test subjects was used to fine-tune the CNN model and achieve good performances. Thus, our method is effective for subject-independent P300 BCIs. Furthermore, as presented in Appendix B, more than 60% of the subjects achieved accuracies higher than 80%, and more than 50% of the subjects achieved accuracies higher than 85%. For these subjects, the subject-specific calibration stage can be entirely eliminated. In addition, less than 40% of subjects has accuracies lower than 80%. We can use a small amount of data of test subjects to fine-tune the CNN model to improve the performance of cross-subject classification for these subjects with unsatisfactory performances.

We used a paradigm that is different from the traditional

P300 speller paradigm. The traditional P300 speller paradigm takes one row/column instead of one button as a flash unit. In this way, the number of flashes is much less than the number of buttons so that the flashing time of a round can be greatly reduced. However, this approach introduces a serious shortcoming to the traditional paradigm. If the row and column of the target character flash just one next to another in a certain round, then the subject will have difficulty distinguishing the two flashes, which makes the latter flash hardly evoke the P300 component. Moreover, in the free spelling phase, since the target is unknown to the speller system, this situation cannot be surely avoided in such a paradigm. To address this issue, we took one character rather than one row/column as a flash unit. We then designed a paradigm in which the button flashes overlap to reduce the flashing time. The approach of overlapping flashing would have an impact on EEG signals; i.e., the EEG segments corresponding to the buttons whose flashes were adjacent to the target button flashes in the time domain would contain components that have the same waveform as P300 but a slight time shift. To alleviate this impact, in each trial, we averaged the signals corresponding to multiple rounds for each button as in Eq. (2) in Section II-D. After the averaging operation, the P300 component of the signals corresponding to the flashes of the target button can be enhanced. In contrast, for nontarget buttons, due to the randomness of the flash sequence for each round, this component would not appear at the same time of its multiple flash rounds and thus will become weak after averaging. Therefore, the EEG signals corresponding to the target and nontarget can be distinguishable.

It is also worth noting that although our CNN architecture was carefully designed and very important to the success of our study, we do not claim that the CNN architecture itself is a key contribution of this paper. The main contribution of this paper is that we found that cross-subjects and cross-amplifier classification models could be established for subject-independent P300 BCIs using a large training dataset, which would substantially improve the applications of P300 BCIs. In fact, as we experimentally showed in Section III-C, similar network structures, such as EEGNet [26], and linear classifiers, such as SVM [41] and LDA [42], may also be effective for analyzing our data to achieve results comparable to ours. However, if the dataset is too large (e.g., much larger than the dataset in this study) or has an extremely high number of dimensions, then an SVM and LDA may not be appropriate for this task due to the requirement of solving a large-scale optimization problem or inverting a very large matrix. Moreover, a CNN allows training with a pretraining-retraining method. In the situation where a small amount of calibration data can be collected from the test subject, a CNN can be pretrained in advance with data collected from a large number of other subjects and then fine-tuned with the calibration data. In this way, at a small cost, a CNN will significantly outperform the other algorithms that are generally trained from scratch.

V. Conclusion

This study presented an invariant pattern learning method based on a CNN and big EEG data for a subject-independent P300 BCI. A CNN with 3 convolutional layers and 2 fully connected layers was built for P300 detection. Two hundred subjects participated in EEG data collection, during which each subject performed a P300-based spelling task, and data were recorded using two types of amplifiers. These subjects were divided into two groups: a group of 150 subjects for training and validation and a group of 50 subjects for an independent test. The offline analysis showed that, without subject-specific calibration, after 10 rounds of button flashes, an average accuracy of more than 80% was achieved. Almost all subjects obtained significant cross-subject and cross-amplifier effects. Furthermore, the ERP waveforms exhibited typical P300 features for subjects with high accuracies but not for subjects with low accuracies. These results indicate that our method based on the CNN and big EEG data was effective in learning invariant patterns across subjects and across amplifiers and is promising for building a subject-independent P300 BCI, especially for subjects whose typical P300 potentials can be evoked by oddball visual stimuli, and the performance could be further improved by fine-tuning the model with a small amount of calibration data from the test subject. However, there was still a small percentage of subjects with low accuracies. In future studies, we will improve the method by, for example, optimizing the network structure, and we will further enlarge the training set to increase the accuracies for this group of subjects and thus increase the percentage of subjects with high accuracies. We will also implement the method in an online system and extend it to patients, such as those with strokes or spinal cord injuries.

References

[1] 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, 1988.

[2] E. Donchin, K. M. Spencer, and R. Wijesinghe, "The mental prosthesis: assessing the speed of a P300-based brain-computer interface," *IEEE Trans. Rehabil. Eng.*, vol. 8, no. 2, pp. 174–179, 2000.

[3] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, and F. Yger, "A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update," *J. Neural Eng.*, vol. 15, no. 3, p. 031005, 2018.

[4] P.-J. Kindermans, M. Schreuder, B. Schrauwen, K.-R. Müller, and M. Tangermann, "True zero-training brain-computer interfacing–an online study," *PLoS ONE*, vol. 9, no. 7, 2014.

[5] Y. Li and C. Guan, "Joint feature re-extraction and classification using an iterative semi-supervised support vector machine algorithm," *Mach. Learn.*, vol. 71, no. 1, pp. 33–53, 2008.

[6] Y. Li, C. Guan, H. Li, and Z. Chin, "A self-training semi-supervised SVM algorithm and its application in an EEG-based brain computer interface speller system," *Pattern Recogn. Lett.*, vol. 29, no. 9, pp. 1285–1294, 2008.

[7] M. Krauledat, M. Tangermann, B. Blankertz, and K.-R. Müller, "Towards zero training for brain-computer interfacing," *PLoS ONE*, vol. 3, no. 8, 2008.

[8] T. Verhoeven, D. Hübner, M. Tangermann, K.-R. Müller, J. Dambre, and P.-J. Kindermans, "Improving zero-training brain-computer interfaces by mixing model estimators," *J. Neural Eng.*, vol. 14, no. 3, p. 036021, 2017.

[9] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2009.

[10] S. Lu, C. Guan, and H. Zhang, "Unsupervised brain computer interface based on intersubject information and online adaptation," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 17, no. 2, pp. 135–145, 2009.

[11] D. Wu, "Online and offline domain adaptation for reducing BCI calibration effort," *IEEE Trans. Hum-Mach. Syst.*, vol. 47, no. 4, pp. 550–563, 2016.

[12] Z. Lan, O. Sourina, L. Wang, R. Scherer, and G. R. Müller-Putz, "Domain adaptation techniques for EEG-based emotion recognition: A comparative study on two public datasets," *IEEE Trans. Cogn. Dev. Syst.*, 2018.

[13] W. Tu and S. Sun, "A subject transfer framework for EEG classification," *Neurocomputing*, vol. 82, pp. 109–116, 2012.

[14] W. Samek, F. C. Meinecke, and K.-R. Müller, "Transferring subspaces between subjects in brain-computer interfacing," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 8, pp. 2289–2298, 2013.

[15] N. R. Waytowich, V. J. Lawhern, A. W. Bohannon, K. R. Ball, and B. J. Lance, "Spectral transfer learning using information geometry for a user-independent brain-computer interface," *Front. Neurosci.*, vol. 10, p. 430, 2016.

[16] P. Zanini, M. Congedo, C. Jutten, S. Said, and Y. Berthoumieu, "Transfer learning: A Riemannian geometry framework with applications to brain–computer interfaces," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 5, pp. 1107–1116, 2017.

[17] P. L. C. Rodrigues, C. Jutten, and M. Congedo, "Riemannian Procrustes Analysis: Transfer learning for brain-computer interfaces," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 8, pp. 2390–2401, 2018.

[18] H. Qi, Y. Xue, L. Xu, Y. Cao, and X. Jiao, "A speedy calibration method using Riemannian geometry measurement and other-subject samples on a P300 speller," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 3, pp. 602–608, 2018.

[19] V. Jayaram, M. Alamgir, Y. Altun, B. Scholkopf, and M. Grosse-Wentrup, "Transfer learning in brain-computer interfaces," *IEEE Comput. Intell. M.*, vol. 11, no. 1, pp. 20–31, 2016.

[20] A. M. Azab, L. Mihaylova, K. K. Ang, and M. Arvaneh, "Weighted transfer learning for improving motor imagery-based brain–computer interface," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 7, pp. 1352–1359, 2019.

[21] P. Yuan, X. Chen, Y. Wang, X. Gao, and S. Gao, "Enhancing performances of SSVEP-based brain–computer interfaces via exploiting inter-subject information," *J. Neural Eng.*, vol. 12, no. 4, p. 046006, 2015.

[22] C.-S. Wei, Y.-P. Lin, Y.-T. Wang, C.-T. Lin, and T.-P. Jung, "A subject-transfer framework for obviating inter- and intra-subject variability in EEG-based drowsiness detection," *NeuroImage*, vol. 174, pp. 407–419, 2018.

[23] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[24] H. Cecotti and A. Gräser, "Convolutional neural networks for P300 detection with application to brain-computer interfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 433–445, 2011.

[25] R. Maddula, J. Stivers, M. Mousavi, S. Ravindran, and V. de Sa, "Deep recurrent convolutional neural networks for classifying P300 BCI signals," in *Proceedings of the Graz BCI Conference*, 2017.

[26] 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, p. 056013, 2018.

[27] A. Craik, Y. He, and J. L. Contreras-Vidal, "Deep learning for electroencephalogram (EEG) classification tasks: a review," *J. Neural Eng.*, vol. 16, no. 3, p. 031001, 2019.

[28] X. Zhang, L. Yao, X. Wang, J. Monaghan, D. Mcalpine, and Y. Zhang, "A survey on deep learning based brain computer interface: Recent advances and new frontiers," *arXiv preprint arXiv:1905.04149*, 2019.

[29] F. Fahimi, Z. Zhang, W. B. Goh, T.-S. Lee, K. K. Ang, and C. Guan, "Inter-subject transfer learning with an end-to-end deep convolutional neural network for EEG-based BCI," *J. Neural Eng.*, vol. 16, no. 2, p. 026007, 2019.

[30] H. Dose, J. S. Møller, H. K. Iversen, and S. Puthusserypady, "An end-to-end deep learning approach to MI-EEG signal classification for BCIs," *Expert Syst. Appl.*, vol. 114, pp. 532–542, 2018.

[31] L. Xu, M. Xu, Y. Ke, X. An, S. Liu, and D. Ming, "Cross-dataset variability problem in EEG decoding with deep learning," *Front. Hum. Neurosci.*, vol. 14, 2020.

[32] T. Yu, Z. Yu, Z. Gu, and Y. Li, "Grouped automatic relevance determination and its application in channel selection for P300 BCIs," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 23, no. 6, pp. 1068–1077, 2015.

[33] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, L. Parkkonen, and M. S. Hämäläinen, "MNE software for processing MEG and EEG data," *NeuroImage*, vol. 86, pp. 446–460, 2014.

[34] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen *et al.*, "MEG and EEG data analysis with MNE-python," *Front. Neurosci.*, vol. 7, p. 267, 2013.

[35] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 9–48.

[36] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[38] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "TensorFlow: A system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.

[39] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, no. Aug, pp. 1871–1874, 2008.

[40] M. Hajinoroozi, Z. Mao, T.-P. Jung, C.-T. Lin, and Y. Huang, "EEG-based prediction of driver's cognitive performance by deep convolutional neural network," *Signal Process.-Image*, vol. 47, pp. 549–555, 2016.

[41] M. Kaper, P. Meinicke, U. Grossekathoefer, T. Lingner, and H. Ritter, "BCI competition 2003-data set IIb: support vector machines for the P300 speller paradigm," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 1073–1076, 2004.

[42] A. Lenhardt, M. Kaper, and H. J. Ritter, "An adaptive P300-based online brain–computer interface," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 16, no. 2, pp. 121–130, 2008.

[43] R. K. Chaurasiya, N. D. Londhe, and S. Ghosh, "Binary DE-based channel selection and weighted ensemble of SVM classification for novel brain–computer interface using Devanagari script-based P300 speller paradigm," *Int. J. Hum.-Comput. Int.*, vol. 32, no. 11, pp. 861–877, 2016.

[44] W. Speier, A. Deshpande, L. Cui, N. Chandravadia, D. Roberts, and N. Pouratian, "A comparison of stimulus types in online classification of the P300 speller using language models," *PLoS ONE*, vol. 12, no. 4, 2017.

[45] L. Bianchi, C. Liti, and V. Piccialli, "A new early stopping method for P300 spellers," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 8, pp. 1635–1643, 2019.

[46] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis*. John Wiley & Sons, 2012, vol. 821.