ST-CapsNet: Linking Spatial and Temporal Attention with Capsule Network for P300 Detection Improvement

Zehui Wang, Chuangquan Chen, *Member, IEEE*, Junhua Li, *Senior Member, IEEE*, Feng Wan, *Senior Member, IEEE*, Yu Sun*, *Senior Member, IEEE*, and Hongtao Wang*, *Senior Member, IEEE*

***Abstract*—A brain-computer interface (BCI), which provides an advanced direct human-machine interaction, has gained substantial research interest in the last decade for its great potential in various applications including rehabilitation and communication. Among them, the P300-based BCI speller is a typical application that is capable of identifying the expected stimulated characters. However, the applicability of the P300 speller is hampered for the low recognition rate partially attributed to the complex spatio-temporal characteristics of the EEG signals. Here, we developed a deep-learning analysis framework named ST-CapsNet to overcome the challenges regarding better P300 detection using a capsule network with both spatial and temporal attention modules. Specifically, we first employed spatial and temporal attention modules to obtain refined EEG signals by capturing event-related information. Then the obtained signals were fed into the capsule network for discriminative feature extraction and P300 detection. In order to quantitatively assess the performance of the proposed ST-CapsNet, two publicly-available datasets (i.e., Dataset IIb of BCI Competition 2003 and Dataset II of BCI Competition III) were applied. A new metric of averaged symbols under repetitions (ASUR) was adopted to evaluate the cumulative effect of symbol recognition under different repetitions. In comparison with several widely-used methods (i.e., LDA, ERP-CapsNet, CNN, MCNN, SWFP, and MsCNN-TL-ESVM), the proposed ST-CapsNet framework significantly outperformed the state-of-the-art methods in terms of ASUR. More interestingly, the absolute values of the spatial filters learned by ST-CapsNet are higher in the parietal lobe and occipital region, which is consistent with the generation mechanism of P300.**

***Index Terms*—brain-computer interfaces (BCIs), capsule network, P300, attention.**

This work was supported in part by Special Projects in Key Fields Supported by the Technology Development Project of Guangdong Province under Grant 2020ZDZX3018, in part by the Special Fund for Science and Technology of Guangdong Province under Grant 2020182, in part by Wuyi University and Hong Kong & Macao joint Research Project under Grant 2019WGALH16, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2020A1515111154, in part by the Educational Commission of Guangdong Province under Grant 2021KTSCX136, in part by the National Natural Science Foundation of China (82172056), in part by National Key Research and Development Program of China (2021ZD0200400), in part by Key Research and Development Program of Zhejiang Province (2022C03064), in part by The Science and Technology Development Fund, Macau SAR under Grant 0045/2019/AFJ and the University of Macau under Grant MYRG2022-00197-FST. (Zehui Wang and Chuangquan Chen contributed equally to this paper, * Corresponding author: Hongtao Wang and Yu Sun).

Zehui Wang, Chuangquan Chen, and Hongtao Wang are with the Faculty of Intelligent Manufacturing, Wuyi University, Jiangmen 529020, China (email: nushongtaowang@qq.com).

Junhua Li is with the Faculty of Intelligent Manufacturing, Wuyi University, Jiangmen 529020, China, and also with the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K..

F. Wan is with the Department of Electrical and Computer Engineering, Faculty of Science and Engineering, University of Macau, Macau, and also with the Centre for Cognitive and Brain Sciences, Institute of Collaborative Innovation, University of Macau, Macau.

Yu Sun is with the Key Laboratory for Biomedical Engineering of Ministry of Education of China, Department of Biomedical Engineering, Zhejiang University, Hangzhou 310027, Zhejiang, China, and also with the Department of Neurology, Sir Run Run Shaw Hospital, Zhejiang University School of Medicine, Hangzhou 310020, Zhejiang, China (yusun@zju.edu.cn).

I. Introduction

BRain-computer interfaces (BCI) provide an opportunity for people to directly interact with their surroundings through brain waves [1] [2]. For example, Long et al. combined motion imagery and P300 potentials to control a 2-D cursor movement [3], and further developed a BCI-based system to control the movement of a wheelchair [4]. Wang et al. identified the user's gaze direction using frequency-encoded steady-state visual evoked potentials [5]. Lin et al. developed a BCI-based system to estimate drivers' drowsiness [6]. Zheng et al. proposed a high-performance brain switch based on code-modulated visual evoked potentials with both fast reaction and low false positive rate (FPR) during idle state [7]. Among all BCI paradigms, Electroencephalography (EEG) is a method of acquiring brain waves that has attracted many researchers to its use due to its high temporal resolution and non-invasive nature [8] [9]. An event-related potential (ERP) based EEG is a brain reaction that occurs directly from a specific event [10]. A typical ERP component, P300 that occurs around 300ms after the target stimulus onset at the parietal lobe, has been widely used in BCI [11] [12] [13]. For instance, Farwell and Donchin [14] proposed a P300 speller paradigm in 1988, allowing individuals to type with their minds. Many datasets of P300 are based on this pioneer paradigm. It is noteworthy mentioning that the international BCI competition datasets also include the P300 paradigm, which are usually the benchmark datasets to compare the performance of various models on EEG classification. Alain Rakotomamonjy and Vincent Guigue [15] won the championship using an ensemble of support vector machines (ESVMs) for P300 detection in BCI III Competition [16]. However, the method does not take into account the importance of the individual electrodes and simply feeds the raw data into the classifier for training.

In order to improve the accuracy of detecting ERP signals, Rivet et al. [17] raised xDAWN, a spatial filtering method, to enhance P300 potentials with respect to the Non-P300

potentials; further, Barachant improved the xDAWN to a generalization to any type of ERP [18]. Most recently, with graphics processing units (GPUs) becoming more powerful, deep learning has grown tremendously. Zhang et al. proposed an improved EEGNet [19] that combined xDAWN saptial filtering with EEGNet [20] for the individually-calibrated rapid serial visual presentation (RSVP) task and won second place in the BCI Controlled Robot Contest at 2022 World Robot Contest [21]. Wang et al. proposed denoising autoencoder neural networks to improve the symbol recognition accuracy by about 0.7% compared to ESVMs, which can automatically learn features from unlabeled data and solve the problem of local minima in neural networks due to random initialization [22]. Cecotti and Graser [23] used convolutional neural networks to detect P300 for the first time and achieved a high recognition rate (95.5%) in the 15th repetition. However, it has a low symbol recognition rate in the first 5 or even 10 repetitions, leading to a low information transfer rate (ITR). To further increase the symbol recognition rate in the first 5 repetitions, Wang et al. [24], who have crowned champions of the P300-based BCI competition in the 2019 World Robot Conference, proposed Multiscale-CNN to enhance the performance of P300 detection. Three temporal kernels at different scales were applied on its temporal convolution layer to obtain discriminative time features. However, some valuable information that would help in classification will be lost during the forward propagation, because it employed the max pooling operation to reduce feature maps which only retains the most active features and discards the rest. To overcome the information loss in the pooling operation, Sabour et al. [25] proposed capsule network (CapsNet). A capsule contains a set of neurons and the output is a vector which represents various entity materialisation parameters, such as position, size, rotation etc. The length of the vector represents the probability of the corresponding class. The lower level capsules are connected to the higher level capsules by a dynamic routing algorithm. Several recent studies have demonstrated that CapsNet could achieve better performance than traditional techniques. For example, we used a multi-kernel capsule network to identify schizophrenia which outperformed other methods in our previous study [26]. Chao et al. combined multiband feature matrix (MFM) and CapsNet outperforming 2D-CNN in emotion recognition [27]. Liu et al. employed 1D-CapsNet to detect P300 which reached 96% symbol recognition rate [28]. Ma et al. attempted to use ERP-CapsNet for ERP detection and obtained much better results than the traditional machine learning algorithms and CNNs [29] and also explained the mechanism of how P300 components are preserved in capsules. However, ERP-CapsNet just took the raw EEG signals as input, which introduced additional noise.

In order to reduce signal noise and further improve the P300 detection accuracy, we employed spatial and temporal attention mechanism to refine the input EEG signals, and then fed the refined EEG signals into ERP-CapsNet for classification. Several attention mechanisms have been widely used, such as the Squeeze-Excitation (SE) block [30] proposed by Hu et al. which adaptively generates channel attention maps and recalibrates the feature responses of channels by explicitly modelling the interdependencies between channels. It first generates average-pooled features from the original convolutional feature maps via the average pooling functions, then feeds the generated features into a multilayer perceptron (MLP) with Sigmoid activation, which yields a channel attention map. Then the element-wise multiplication of original convolutional feature maps with the channel attention map gives the calibrated channel feature response, i.e., the channel refined feature map. The work of Hu et al. inspired Woo et al. to develop a more powerful attention mechanism, the Convolutional Block Attention Module (CBAM) [31]. It consists of a channel attention module and a spatial attention module. The channel attention module is a variant of the SE block. It generates average-pooled and max-pooled features from the original convolutional feature maps via the average pooling and max pooling functions which are then fed into a shared MLP where the outputs are summed and activated by a Sigmoid function to produce a channel attention map. The channel attention map is also element-wise multiplied with original convolutional feature maps to obtain channel refined feature maps. The spatial attention module first compresses the channel refined feature maps into two features via the max and average pooling functions respectively, and then generates the spatial attention map via a 7×7 convolution. Finally, the channel refined feature maps are element-wise multiplied with the spatial attention maps to obtain channel and spatial refined feature maps. Their experimental results on various image datasets showed that inserting CBAMs into the baseline model can significantly improve the classification performance. Inspired by this, we try to combine ERP-CapsNet [29] with CBAMs, which we call ST-CapsNet, expecting to improve the performance of P300 detection.

The main contributions of this work are summarized as follows: 1) To our knowledge, this is the first attempt to combine spatial and temporal attention with a capsule network to improve the accuracy of P300 detection. 2) We proposed a more comprehensive method (ASUR) to measure symbol recognition performance by comparing the average correctly recognized symbols under the first 5, 10 and 15 repetitions of a stimulus round.

II. DATASETS

A. Description

The data sets used in this paper are the dataset IIb of BCI competition 2003 and dataset II of BCI competition III [16]. We separated dataset II into two data sets: dataset II-A and II-B because it contains two subjects (subjects A and B). These datasets are complete records of P300 evoked potentials recorded with BCI2000 [32] using a paradigm described by Farwell and Donchin [14]. The subjects were presented with a 6x6 matrix of symbols. All rows and columns in the matrix were randomly intensified at a frequency of 5.7 Hz. By staring at the desired symbol in the matrix, a P300 evoked potential would occur in the subjects' brains when the desired symbol flashed. When other symbols flashed, stimulated potentials do not have a P300 component and are called Non-P300 evoked potentials. The P300 potentials are different from the Non-P300 potentials, because the rare target stimuli cause subjects'

brains to generate P300 potentials [33]. Six columns and six rows were randomly intensified in the matrix; only one column and one row contain the desired symbol, which means there are two P300 evoked potentials and ten Non-P300 evoked potentials in one stimulation round. Due to the extremely low SNR of ERPs, the stimulation round should be repeated several times to improve the P300 recognition accuracy.

The EEG data was recorded from 64 electrodes at a sampling rate of 240 Hz in several sessions. Each session consisted of a number of runs. In each run, subjects focused on a series of symbols. At first the screen was displayed for 2.5 seconds, during which time each symbol had the same intensity (i.e., the matrix was blank). Subsequently, one of the rows or columns in the matrix was randomly enhanced for 100 ms, and then the matrix was blanked for 75 ms. The enhancement of the rows/columns was carried out randomly 12 times in a block. The block was repeated 15 times for each symbol to spell. There were a total of 31 symbols in dataset IIb, and 100 symbols in datasets II-A and II-B. Table I shows the number of P300 and Non-P300 samples for training and testing in each dataset. For more information pertaining to the dataset, please refer to https://www.bbci.de/competition.

TABLE I: The sample composition of the training and testing sets for each dataset.

Dataset	Training		Testing	
	P300	Non-P300	P300	Non-P300
IIb	1260	6300	930	4650
II-A	2550	12750	3000	15000
II-B	2550	12750	3000	15000

B. Data Preprocessing

To reduce the effect of the imbalance of the data sets, we averaged two randomly selected samples from P300 samples many times so that the number of P300 is the same as the number of Non-P300. The preprocessing step consists of the following stages. We first extracted all data samples between 0 to 650 ms, i.e., 156 time samples after the start of an intensification. Afterwards, an FIR band-pass filter (Hamming window) with a frequency range of 0.1 to 20Hz was adopted that was followed by downsampled (to half of the staple points for each channel) and normalized steps (via Z-score in eq (1) and sigmod approach) to normalize the filtered EEG data. The sigmoid function was used because the value range of the reconstructed EEG signal in the decoder layer is from 0 to 1. The obtained band-pass filtered and normalized EEG data was set as input for the ST-CapsNet.

$$X_{ij} \leftarrow \frac{X_{ij} - \bar{X}_i}{\sigma_i} \tag{1}$$

$X \in \mathbb{R}^{C\times 78}$ is the half downsampled filtered EEG signal and X_{ij} is the signal value of the i-th electrode at the j-th time point. $\bar{X}_i$ and σ_i are the average and standard deviation of the *i*-th electrode signal. C represents the number of electrodes, and 78 stands for the time samples of the signals. We set C to 64 because datasets IIb, II-A, and II-B all have 64 electrodes.

III. Methods

ERP-CapsNet has shown good performance in P300 detection [29]. However, it just took the raw EEG signals as input which might introduce some additional noise. Hence, to reduce the noise of EEG signals and improve the accuracy of P300 detection, we linked spatial and temporal attention modules with ERP-CapsNet as illustrated in Fig.1.

A. Spatial Attention

We define the variable $V \in \mathbb{R}^{c\times h\times w}$, where c, h and w represent the channel, height and width dimensions of V, respectively. The spatial attention module is used to enhance the spatial information of the raw input EEG signal X, as summarised below.

$$M_S = \sigma({W_1}^T \mathrm{ReLU}({W_0}^T F^s_{avg}) \oplus {W_1}^T \mathrm{ReLU}({W_0}^T F^s_{max})) \tag{2}$$

where F^s_{avg} and $F^s_{max} \in \mathbb{R}^{C\times 1\times 1}$ are the features generated from the reshaped signal $X_R \in \mathbb{R}^{C\times 1\times 78}$ through max pooling and average pooling function along the width dimension (the pooling kernel size and pooling stride were set to 78 and 1, respectively). In the shared MLP, $W_0 \in \mathbb{R}^{C\times \frac{C}{r}}$ is the weight between the input layer and the hidden layer, while $W_1 \in \mathbb{R}^{\frac{C}{r}\times C}$ is the weight between the hidden layer and the output layer, and *r* is the reduction ratio. We set *r* to 16 as suggested in [31]. The function $\oplus$ denotes element-wise addition, and σ is the sigmoid operation. $M_S \in \mathbb{R}^{C\times 1\times 1}$ is the spatial attention map that we get at last in the spatial attention module. By simply multiplying the reshaped signal X_R with the spatial attention map M_S through the function $\otimes$ which denotes the element-wise multiplication, we get the spatial refined signal $X_S \in \mathbb{R}^{C\times 1\times 78}$. Note that M_s is auto broadcasted along the width dimension when doing the element-wise multiplication due to the special mechanism of Pytorch [34].

$$X_S = X_R \otimes M_S \tag{3}$$

B. Temporal Attention

In the temporal attention module, the spatial refined signal X_S first compressed itself into two feature maps (i.e., F^t_{avg} and $F^t_{max} \in \mathbb{R}^{1\times 1\times 78}$) through max pooling and average pooling function along the channel dimension (the pooling kernel size and pooling stride were set to C and 1, respectively). Then the two feature maps were stacked and convolved by a convolution layer with a $1 \times D$ (D can be taken as 3, 5, and 7) filter, a stride of 1, same padding, and sigmoid activation, producing a temporal attention map $M_T \in \mathbb{R}^{1\times 1\times 78}$.

$$M_T = \sigma\left(Conv^{1\times D}\left(F^t_{avg}; F^t_{max}\right)\right) \tag{4}$$

Afterwards, we can get the refined EEG signal $X_{ST} \in \mathbb{R}^{C\times 1\times 78}$ through the function below, which is auto broadcasted along the channel dimension.

$$X_{ST} = X_S \otimes M_T \tag{5}$$

C. Capsule Network

In the capsule network, we first extracted temporal features from X_{ST} using 10 $C \times 1$ spatial filters through convolution

This article has been accepted for publication in IEEE Transactions on Neural Systems and Rehabilitation Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2023.3237319

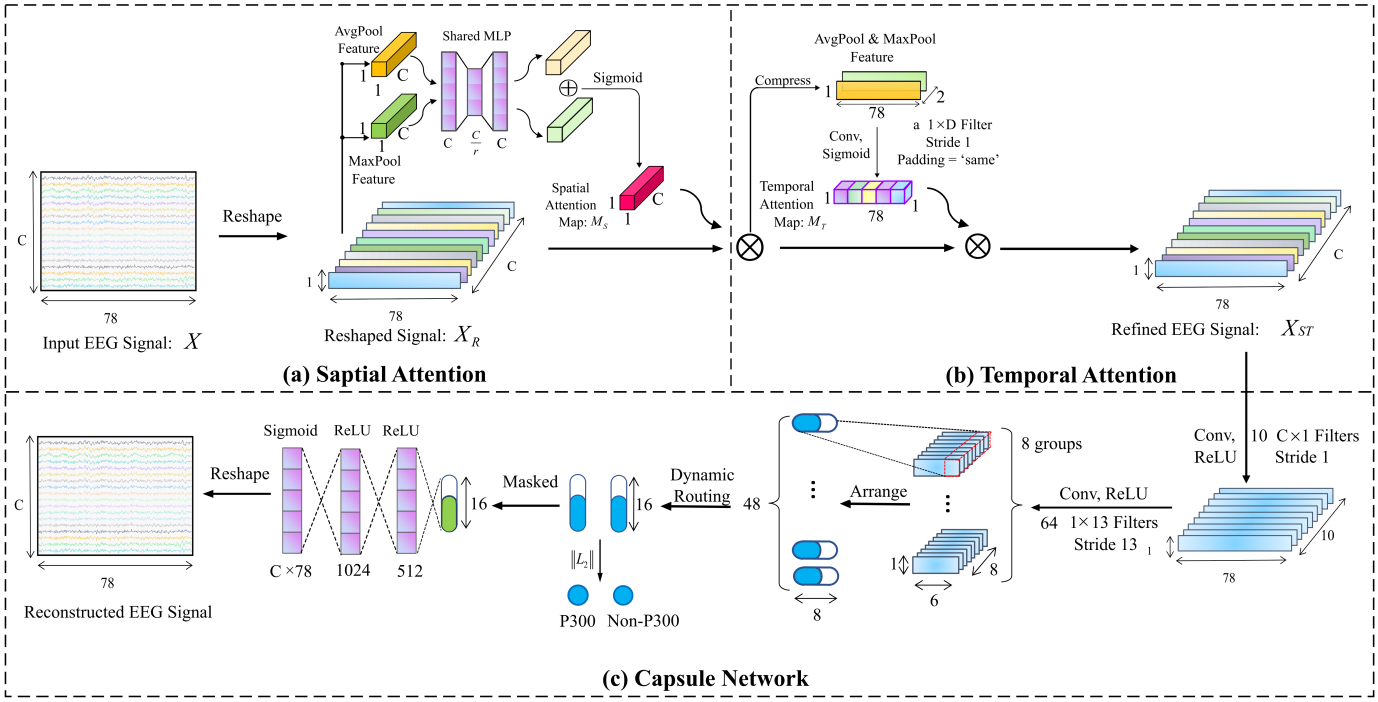

Fig. 1: Overview of ST-CapsNet architecture. It consists of three blocks, (a) the spatial attention module, (b) the temporal attention module and (c) the capsule network.

operation and ReLU function, where the stride is 1. Next, we used 64 1×13 temporal filters by convolution and ReLU operations to extract temporal features of which size is $64 \times 1 \times 8$. The temporal features are divided into 8 groups. The size of each group is $8 \times 1 \times 6$, which means six *8D* primary capsules. So we got 48 *8D* primary capsules in total as the input of the dynamic routing. The output of the dynamic routing is two *16D* output capsules. The lengths of the two output capsules, calculated through the L2 norm and then activated by Softmax, represent the probabilities of P300 and Non-P300, respectively. We can determine the label of the input sample X using eq (6), where p_{target} and p_{non} represent the probability that the model identifies sample X as a P300 and a non-P300 sample, respectively

$$Classifier(X) = \begin{cases} 1, & (p_{target} > p_{non}) \\ 0, & (otherwise) \end{cases} \tag{6}$$

The mechanism of the dynamic routing algorithm is completely different from that of the CNN and is described in Algorithm 1. Sabour et al. suggested that better convergence can be obtained by using three routing iterations than one iteration [35]. Therefore, we set the maximum number of routing iterations, i.e., N to 3. After the dynamic routing layer, we keep the output capsule representing the category of the input EEG sample X as the input of the decoder network and mask the other output capsule. The decoder network consists of three fully connected layers; the number of neurons is 512, 1024, $C \times 78$, and the activation functions are ReLU, ReLU, sigmoid.

The loss function of ST-CapsNet consists of two components, namely margin loss and reconstruction loss. The margin loss is defined as follows:

$$\begin{aligned} L_j = T_j \max(0, m^+ - ||v_j||)^2 \\ + \lambda(1 - T_j) \max(0, ||v_j|| - m^-)^2 \end{aligned} \tag{7}$$

where L_j stands for the loss of *j*-th output capsule, λ = 0.5, m^+ = 0.9, m^- = 0.1. T_j = 1 if the label of the input sample is *j*, otherwise T_j = 0. For binary classification, the margin loss function is more efficient, because it punishes the predictions depending on how closely they match with the sign of the target [36]. The reconstruction loss L_r is obtained by calculating the mean squared error between the input EEG signal and the reconstructed EEG signal. Adding the reconstruction loss can boost the routing performance [25]. The total loss of the ST-CapsNet is summed as follows:

$$L = \sum_{j=1} L_j + \alpha L_r \tag{8}$$

where α is set to 0.0005.

D. Training

We used parameters of a pre-trained model to initialize ST-CapsNet in attention layers and two convolution layers to obtain better convergence and avoid local optimum as suggested in [35]. The pre-trained model is shown in TableII. All models were implemented in PyTorch and trained on GeForce RTX 2080 Ti. The batch size was set to 64. The learning rate was initially set to 0.001 with an exponential decay rate of 0.96. For the pre-trained CNN, we employed cross-entropy loss. The Adam optimizer with default parameters was used to optimize all models. To avoid overfitting, the early stop and data augmentation in braindecode [37] were used.

Algorithm 1: Dynamic routing

Input:

$u_i \in \mathbb{R}^8$: primary capsule, $i \in \{1, 2, ...48\}$

Output:

$v_j \in \mathbb{R}^{16}$: output capsule, $j \in \{1, 2\}$

Begin:

b = **0** ▷ initialize parameter $b \in \mathbb{R}^{48\times 2}$

for k in 1:N **do** ▷ routing iteration

for i in 1:48 and j in 1:2 **do**

$c_{ij} = \frac{exp(b_{ij})}{\sum_{j=1}^{2} exp(b_{ij})}$ ▷couping coeffcients

for j in 1:2 **do**

$s_j = \sum_{i=1}^{48} c_{ij} W_{ij} u_i$ ▷ weight $W_{ij} \in \mathbb{R}^{16\times 8}$

$v_j = \frac{\|s_j\|^2}{1+\|s_j\|^2} \frac{s_j}{\|s_j\|}$ ▷ squash

for i in 1:48 and j in 1:2 **do**

$b_{ij} = b_{ij} + W_{ij} u_i \cdot v_j$ ▷ update

return v_1, v_2

TABLE II: Pre-trained network architecture. C represents the number of electrodes. C is set to 64 for datasets IIb, II-A, and II-B.

Layer	#Filters	Size	Output	Activation	Stride
Reshape			(C, 1, 78)		
Attention(spatial, temporal)			(C, 1, 78)		
Reshape			(C, 78)		
Conv2D	10	(C, 1)	(10, 1, 78)	ReLU	1
Covn2D	64	(1, 13)	(64, 1, 6)	ReLU	13
Fully-connected	384		2	Softmax	

E. Target Symbol Recognization

The StimulusCodes [16] shown in Fig.2 have a value range of 0 to 12 (0 when no row/column is being intensified, 1 to 6 for intensified columns, 7 to 12 for intensified rows). Because

Fig. 2: Different row/column intensifications are assigned to the StimulusCodes [16]. The numbers in blue are the StimulusCodes.

of the low SNR of ERP, subjects need to take 15 repetitions to recognize one symbol in the P300 speller paradigm. Each repetition has 12 stimuli that correspond to 12 stimulus codes. Let ${p_k}^{(i)}$ denote the length of the *16D* output capsule which stands for the probability of P300 when the stimulus code is *k* in the *i*-th repetition. P_k is the sum of those P300 possibilities from the first to the *n*-th repetition.

$$P_k = \sum_{i=1}^{n} {p_k}^{(i)} \tag{9}$$

Then we can identify the column *c* and row *r* of the target symbol in the *n*-th repetition by:

$$c = \arg\max_{k\in[1,6]} P_k \tag{10}$$

$$r = \arg\max_{k\in[7,12]} P_k \tag{11}$$

IV. RESULTS

A. Algorithms for Comparison

To evaluate the accuracy of P300 detection and symbol recognition, we compared our ST-CapsNet with six models (i.e., a capsule network, two traditional methods, two deep learning models, and a method combining deep learning and traditional algorithms). The details of the models are described as follows:

1) ERP-CapsNet, which was state of the art, is the first capsule network applied to ERP detection and achieved good results [29]. The network structure is the same as the Capsule Network in Fig.1.
2) CNN-1 is the first proposed CNN model for P300 detection [23]. It consists of four layers; the first two are convolutional layers (with a 64×1 spatial kernel and 50 1×13 temproal kernels separately) used to extract spatial and temporal features respectively, and the last two are fully connected layers (with 100 and 2 neurons respectively) used to classify ERP signals.
3) MCNN-1 is an ensemble of five CNN-1 models, each trained on a different partition of the data [23]. There are five data partitions in total because the number of Non-P300 samples is five times larger than the number of P300 samples in the original data. Each data partition is derived from the original data and has the same number of P300 and Non-P300 samples. CNN-1 and MCNN-1 are often used to compare P300 performance as benchmarks.
4) Linear discriminant analysis (LDA) with covariance shrinkage has shown better performance than a conventional LDA classifier in detecting single trial ERP signals [38]. We abbreviated this approach as S-LDA and copied the reproduction results done by Ma et al. [29] for a clear comparison.
5) Spatially Weighted FLD-PCA (SWFP) is designed for single trial ERP detection, which outperformed than Hierarchical Discriminant Component Analysis (HDCA) [39] and Hierarchical Discriminant Principal Component Analysis Algorithm (HDCPA) [40]. First, a spatial filter is estimated at each time point using Fisher Linear Discriminant (FLD), and then all the estimated spatial filters (78 in total) are applied to an EEG sample to obtain a spatially filtered EEG sample. Each channel of this EEG sample is then applied with principal component analysis (PCA) for dimensionality reduction. Six principal components are retained to explain $> 70\%$ variance as reported in [40].

6) MsCNN-TL-ESVM was proposed by Sourav Kundu and Samit Ari [41]. It consists of two blocks, the feature extraction block and the classification block. The authors first used a convolution network with spatial filters with fixed size (64×1) and multiple temporal filters of different sizes (1×20 and 1×10) based on transfer learning to extract discriminant spatial and temporal features, after which they applied Fisher ratio to select important features and then sent those selected features to the ensemble of SVMs for symbol recognition.

B. Evaluation Metrics

We adopted accuracy (Acc.) and F1-score as metrics to evaluate the performance of P300 detection in single trial. To evaluate the performance of symbol recognition, it is not sufficient to compare the number of symbols correctly recognized under separate repetitions, because the P300-based speller paradigm has the characteristic of cumulative effect, i.e., the recognition accuracy of the previous repetition affects the recognition accuracy of the next repetition. Here we give an assumption that a good model should perform well with fewer repetitions (reach a higher information transfer rate) without sacrificing overall performance (correctly identifying as many symbols as possible under all repetitions). Hence, to quantify the performance of models in recognizing symbols under different repetitions, we proposed a comprehensive evaluation measure as following:

$$\mathrm{ASUR}_k = \frac{1}{k}\sum_{i=1}^{k} C_i \tag{12}$$

where C_i means the correctly recognized symbols in the *i*-th repetition. ASUR_k stands for the average correctly recognized symbols per repetition when we take k repetitions into account. We take three values of k (5, 10, 15). ASUR_5, ASUR_{10} and ASUR_{15} represent the average correctly recognized symbols in the first five, ten and fifteenth repetitions separately. It is worth mentioning that ASUR_{15} means the overall performance of symbol recognition because there are 15 repetitions in total. Besides, higher ASUR_5 and ASUR_{10} mean higher accuracy of symbol recognition with fewer repetitions. In addition, to compare the symobol recognition speed of models under different repetitions, we referred to the formula for calculating ITR under the i-th repetition in the paper [42], defined as follows:

$$\mathrm{ITR} = \frac{60\left((1-A_i)\log_2\frac{1-A_i}{G-1} + A_i\log_2 A_i + \log_2 G\right)}{2.5+2.1i} \tag{13}$$

where A_i is the accuracy of symbol recognition rate (in percent) under the i-th repetition, and G (G is 36 here) is the number of symbols presented in the p300-speller paradigm as shown in Fig. 2.

C. Performace of P300 Detection in Single Trial

The kernel size of temporal attention module in ST-CapsNet was chosen to be 1×5. The results are shown in Table III. It is obvious that ST-CapsNet outperforms other models both in accuracy and F1-score on datasets IIb and II-B, while ERP-CapsNet has a little higher F1-score than ST-CapsNet on dataset II-A. The results indicate attention modules of ST-CapsNet could boost the performance of P300 detection in single trial.

TABLE III: Results of P300 detection in single trial.

Model	Dataset					
	II-b		II-A		II-B	
	Acc.	F1-score	Acc.	F1-score	Acc.	F1-score
ST-CapsNet	**0.8997**	**0.7213**	**0.7785**	0.4535	**0.8342**	**0.5405**
ERP-CapsNet [29]	0.8927	0.7146	0.7456	**0.4546**	0.8160	0.5287
CNN-1	0.8810	0.6957	0.7037	0.4311	0.7819	0.5090
MCNN-1	0.8746	0.6899	0.6899	0.4260	0.7586	0.5034
S-LDA [29]	0.8694	0.6428	0.7435	0.4336	0.8057	0.4983
SWFP	0.8448	0.6430	0.7302	0.4462	0.7840	0.5073
MsCNN-TL-ESVM	0.8778	0.6945	0.7492	0.4543	0.8153	0.5382

D. Performance of Symbol Recognition

The correctly recognized symbols in every repetition for each model on datasets IIb, II-A, II-B are shown in Tables IV, V, VI. The character '-' means the authors did not report the value in their papers. Table IV illustrates that ST-CapsNet, ERP-CapsNet, CNN1 and MsCNN-TL-ESVM can correctly identify all symbols in the 4th repetition, while S-LDA requires 5 repetitions and even SWFP need take 8 repetitions to correctly recognize all symbols on dataset IIb. In addition, ST-CapsNet and MsCNN-TL-ESVM have almost the same performance and are better than the other methods. On dataset II-A, both ST-CapsNet and ERP-CapsNet correctly identified 98 symbols in the 15th repetition, and ST-CapsNet is more accurate in the 5th to 10th repetitions while ERP-CapsNet is more accurate in the 11th to 13th repetitions. On dataset II-B, ST-CapsNet has the highest accuracy from repetition 4 to 7 , while MsCNN-TL-ESVM are the most accurate from repetition 9 to 13.

TABLE IV: Number of correctly classified symbols for dataset IIb.

Model	Repetition														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ST-CapsNet	**27**	**29**	**30**	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**
ERP-CapsNet [29]	24	27	29	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**
CNN1	21	25	29	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**
MCNN-1	23	26	29	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**
S-LDA [29]	20	23	25	27	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**
SWFP	21	26	27	29	29	30	30	**31**	**31**	**31**	**31**	**31**	**31**	**31**	**31**
MsCNN-TL-ESVM [41]	**27**	28	**30**	**31**	**31**	**31**	**31**	**31**	-	-	-	-	-	-	-

TABLE V: Number of correctly classified symbols for dataset II-A.

Model	Repetition														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ST-CapsNet	18	31	**53**	56	**68**	**79**	**82**	**85**	84	**88**	89	92	92	**95**	**98**
ERP-CaspNet [29]	16	36	52	57	**68**	75	76	83	82	87	**92**	**94**	**94**	**95**	**98**
CNN-1 [23]	16	33	47	52	61	65	77	78	**85**	86	90	91	91	93	97
MCNN-1 [23]	18	31	50	54	61	68	76	76	79	82	89	92	91	93	97
S-LDA [29]	14	24	46	55	58	66	77	75	79	85	86	89	90	91	95
SWFP	16	28	48	**58**	67	71	78	81	84	87	89	92	91	**95**	97
MsCNN-TL-ESVM [41]	**24**	**38**	46	50	60	70	72	79	84	86	89	89	92	94	96

This article has been accepted for publication in IEEE Transactions on Neural Systems and Rehabilitation Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2023.3237319

TABLE VI: Number of correctly classified symbols for dataset II-B.

Model	Repetition														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ST-CapsNet	41	**61**	66	**78**	**85**	**86**	**92**	90	91	95	**96**	96	95	**97**	**96**
ERP-CaspNet [29]	**45**	60	66	73	81	85	87	90	90	94	94	94	94	95	**96**
CNN-1 [23]	35	52	59	68	79	81	82	89	92	91	91	90	91	92	92
MCNN-1 [23]	39	55	62	64	77	79	86	**92**	91	92	95	95	95	94	94
S-LDA [29]	39	54	64	67	75	78	85	88	87	92	92	96	93	94	96
SWFP	40	60	**69**	71	80	82	86	89	89	92	92	93	92	94	94
MsCNN-TL-ESVM [41]	40	59	67	74	79	84	90	**92**	**94**	**97**	**96**	**98**	**97**	**97**	**96**

TABLE VII: $ASUR_k$ (k = 5, 10, 15) on datasets II-b, II-A and II-B

Model	dataset II-b			dataset II-A			dataset II-B		
	5	10	15	5	10	15	5	10	15
ST-CapsNet	**29.6**	**30.3**	**30.5**	45.2	**64.4**	**74.0**	**66.2**	**78.5**	**84.3**
ERP-CaspNet	28.4	29.7	30.1	**45.8**	63.2	73.7	65.0	77.1	82.9
CNN-1	27.4	29.2	29.8	41.8	60.0	70.8	58.6	72.8	78.9
MCNN-1	28.0	29.5	30.0	42.8	59.5	70.5	59.4	73.7	80.7
S-LDA	25.2	28.1	29.1	39.4	57.9	68.7	59.8	72.9	80.0
SWFP	26.4	28.5	29.3	43.4	61.8	72.1	64.0	75.8	81.5
MsCNN-TL-ESVM	29.4	30.2	30.4	43.6	60.9	71.3	63.8	77.6	84.0

As summarized in Table V and Table VI, some models have higher accuracy when there are more repetitions but lower recognition accuracy when there are fewer repetitions, which means that different models have different accuracy tendencies under repetitions. Our ST-CapsNet tends to be more accurate with fewer repetitions, while ERP-CapsNet and MsCNN need more repetitions to be accurate. Table VII illustrated that ST-CapsNet has the highest accuracy of symbol recognition on the overall performance (highest $ASUR_{15}$) on the three datasets (II-b, II-A and II-B). ERP-CapsNet is a little more accurate in the first 5 repetitions. In summary, our ST-CapsNet outperforms ERP-CapsNet by about 1 percent and is better than the other models in symbol recognition.

E. Performance of ITR

To show the speed of symbol spelling, we compared the ITR under each repetition as shown in Fig.3. The kernel size of the temporal module was chosen to be 1×5. On dataset IIb, ST-CapsNet and MsCNN-TL-SVM achieved almost the same ITR performance (both with highest ITR of 51.56 bits/min) and outperformed the other models significantly. Furthermore, ST-CapsNet achieved the highest ITR of 13.32 bits/min in the 6th repetition on dataset II-A and 19.74 bits/min in the 2nd repetition on dataset II-B, respectively. Interestingly, we found that with the same symbol recognition rate, the performance of ITR decreases significantly with the number of repetitions. Thus, improving the symbol recognition rate for the first few repetitions is a key point to obtain a higher ITR.

F. Effect of Temporal Attention to Model Performance under Various Kernel Sizes

We also explored the performance of ST-CapsNet with different temporal attention kernel sizes (1×3, 1×5, 1×7). Table VIII illustrates that, in single trial P300 detection, 1×3 kernel outperformed the other two on dataset IIb, and 1×5 is the best on datasets II-A and II-B. Although there is a difference in performance between these three kernels in detecting the P300, it is not significant. The number of correctly recognized symobols and $ASUR_k$ values are given in Tables IX,X separately. ST-CapsNet with 1×7 kernel has better performance of symbol recognition in the first five and ten repetitions, while with 1×5 kernel has the best overall performance. Those findings showed that ST-CapsNet is not sensitive to the choice of kernel size of the temporal attention module.

TABLE VIII: Results of P300 detection under different kernel sizes of temporal attention module.

Kernel	Dataset					
	II-b		II-A		II-B	
	Acc.	F1-score	Acc.	F1-score	Acc.	F1-score
1x3	**0.9046**	**0.7256**	0.7514	0.4520	0.8364	0.5355
1x5	0.8997	0.7213	**0.7785**	**0.4535**	0.8342	**0.5405**
1x7	0.9016	0.7228	0.7609	0.4526	**0.8414**	0.5402

TABLE IX: Number of correctly classified symbols under different kernel sizes of temporal attention module.

Dataset	Kernel	Repetition														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
IIb	1x3	25	27	30	31	31	31	31	31	31	31	31	31	31	31	31
	1x5	27	29	30	31	31	31	31	31	31	31	31	31	31	31	31
	1x7	26	27	29	30	31	31	31	31	31	31	31	31	31	31	31
II-A	1x3	19	29	54	60	70	77	79	84	84	89	89	91	93	94	95
	1x5	18	31	53	56	68	79	82	85	84	88	89	92	92	95	98
	1x7	20	36	54	60	72	76	80	82	79	89	89	93	89	93	97
II-B	1x3	43	62	68	74	82	86	88	90	90	94	95	95	94	96	96
	1x5	41	61	66	78	85	86	92	90	91	95	96	96	95	97	96
	1x7	47	60	71	73	81	88	89	90	91	93	95	94	95	96	96

TABLE X: $ASUR_k$ (k = 5, 10, 15) under under different kernel sizes of temporal attention module.

Kernel	dataset II-b			dataset II-A			dataset II-B		
	5	10	15	5	10	15	5	10	15
1x3	28.8	29.9	30.3	46.4	64.5	73.8	65.8	77.7	83.5
1x5	**29.6**	**30.3**	**30.5**	45.2	64.4	**74.0**	66.2	**78.5**	**84.3**
1x7	28.6	29.8	30.2	**48.4**	**64.8**	73.9	**66.4**	78.3	83.9

V. DISCUSSION

In this paper we used a capsule network with spatial and temporal attention modules to improve the performance of detecting P300. This method has superior performance compared to ERP-CapsNet, CNN-1, MCNN-1, S-LDA, SWFP, MsCNN-TL-ESVM for P300 detection in single trial. Among them, the traditional methods (S-LDA, SWFP) have the worst performance, probably because those handcrafted features do not contain rich discriminative information, and the number of parameters of these two models is so small that there is a risk of underfitting. The results of classical convolutional networks (CNN-1, MCNN1) are slightly better, but still less satisfactory. ERP-CapsNet is about two points higher than classical convolutional networks, probably because the capsule network used a dynamic routing layer to replace the max pooling layer, thus

This article has been accepted for publication in IEEE Transactions on Neural Systems and Rehabilitation Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2023.3237319

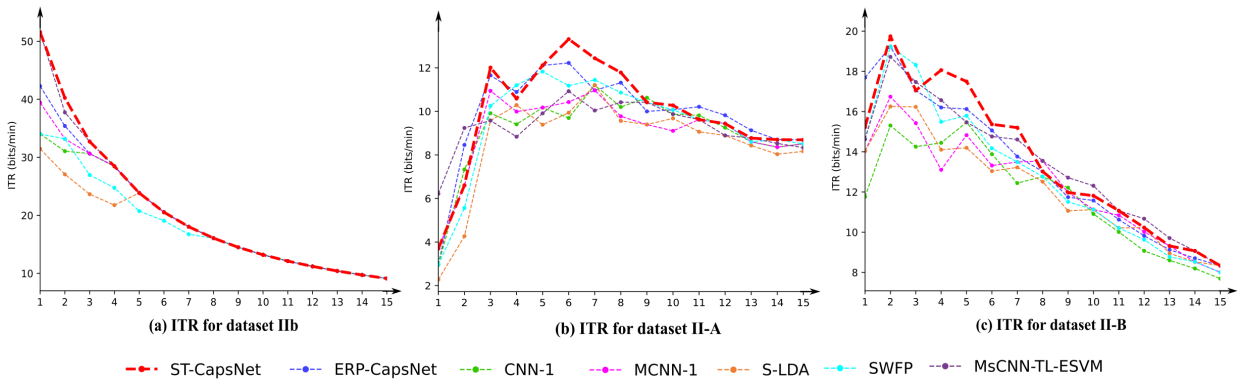

Fig. 3: ITR Comparison under 15 repetitions. (a), (b) and (c) are for datasets IIb, II-A and II-B, respectively

avoiding information loss during backpropagation. MsCNN-TL-ESVM is a combination of multi-scale convolutional network (automatically extract rich multi-scale temporal features) and ensembled SVMs (reduce the variance of the classifiers to avoid the risk of overfitting), and employed migration learning training stratage (ensure the amount of training data). The results are excellent and have nearly the same performance as ERP-CapsNet. Our proposed ST-CapsNet outperformed ERP-CapsNet by about 1 percentage probably because we employed attention mechanisms to make the capsule model automatically learn and strengthen discriminative features focusing on space and time.

To be able to accurately detect the symbols to be spelled, a typical solution is to increase the number of repetitions which could improve SNR. However, as the number of repetitions increases, the time taken to detect individual symbols becomes longer. A good model should be able to recognize as many symbols as possible with as few repetitions as possible. A traditional metric of evaluating the accuracy of symbol recognition is to directly compare the correctly recognized symbols at repetitions 5, 10 and 15, respectively as used in [43] [44]. However, this approach does not take into account the cumulative effect of the P300-based speller paradigm, where the spelling accuracy of the previous repetiton affects the accuracy of the next repetition. Thus, we introduced a new metric ASUR to evaluate the accuracy of symbol recognition. Higher ASUR_5 and ASUR_{10} indicate higher average symbol recognition rate for the first 5 and the first 10 repetitions, respectively. Higher ASUR_{15} indicates better overall performance of the symbol recognition. Our experimental results show that the spatial and temporal attention modules can improve the accuracy of ERP-CapsNet for symbol recognition at low repetitions without losing the overall performance. In addition, in the temporal attention module, we tested different sizes of kernels (1×3, 1×5 and 1×7). These three different kernels all could achieve better results than ERP-CapsNet on both P300 detection in single trial and symbol recognition with similar performance, indicating that ST-CapsNet is less sensitive to the choice of kernel size.

To investigate the region of interest learned by spatial attention module, we ranked the averaged values of spatial attention maps in descending order, and marked top eight electrodes in red as shown in Fig.4. We found that all three spatial attention maps share two common channels (Cz and CPz), and the enhanced electrodes were located roughly in the central and parietal lobes of the brain, indicating that the attention module was able to capture the spatial features of P300. Furthermore, the learned spatial attention maps generally accords with those of previous studies [24] [15] [23].

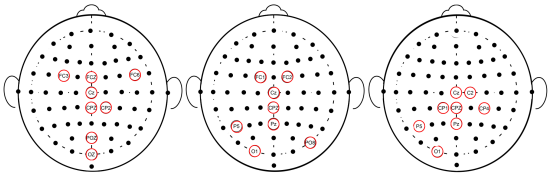

Fig. 4: Region of Interest (marked by red circles) in spatial attention module. These red-circled channels correspond the largest 8 values in the spatial attention maps. The leftmost, middle, and rightmost represent the regions of interest for datasets IIb, II-A, and II-B, respectively

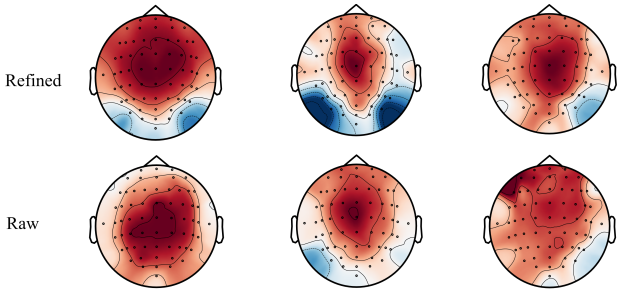

Fig. 5: Averaged topography with the target minus the non-target component over the entire time winodw. From the left-most column to the rightmost column correspond to datasets IIb, II-A, II-B

To further investigate the mechanisms of how the spatial and temporal attention modules affect the raw EEG signals, we sent all raw EEG signals to attention layers and obtained refined EEG signals. However, due to complex non-linear transformations, the characteristics of the EEG signals change considerably in time and space, which is difficult to understand humanly. From another perspective, comparing the difference between the mean P300 signal and the mean Non-P300 signal

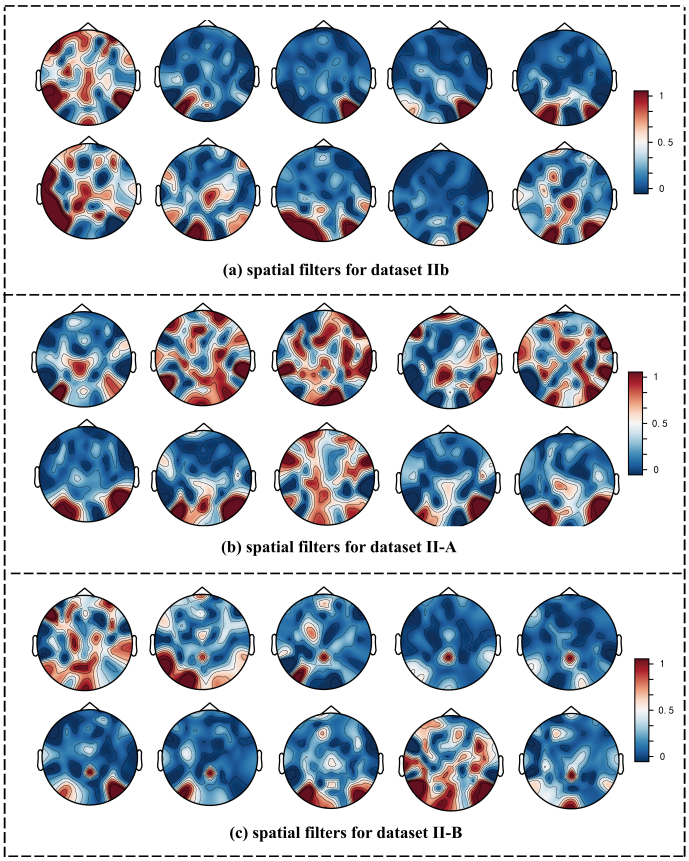

Fig. 6: The spatial filters obtained in the first convolutional layer. There are 10 spatial filter feature maps for each dataset. (a), (b) and (c) correspond to datasets IIb, II-A and II-B, respectively

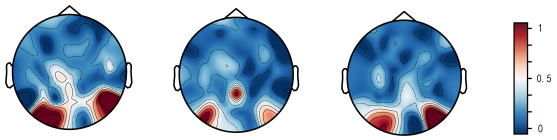

Fig. 7: The weights of the average of 10 spatial filters in the first convolution layer. The topographic maps correspond to datasets IIb, II-A, II-B from the leftmost to the rightmost.

is a better approach, as the attention layers maximize the difference between the P300 samples and the Non-P300 samples, as shown in Tables III. We therefore subtracted the mean Non-P300 signal from the mean P300 signal and averaged the EEG topographies over the entire time period, as plotted in Fig.5. We can see from this figure that on datasets IIb and II-A, the energy areas of both the refined and raw EEG topographies are concentrated in the parietal lobe; while on dataset II-B, the energy in the parietal lobe of the refined versus the raw EEG topographies is more focused. Those findings indicate that attentional mechanisms can enhance the ability to capture P300 features.

We also explored the spatial features learned by the capsule newtork in ST-CapsNet. First, we selected 10 spatial filters in the first convolutional layer of the capsule network, and took their absolute values for normalization. Next, we used MNE-Python [45] to plot the topography of datasets IIb, II-A and II-B. Fig.6 shows the weights of each of the learned spatial filters. Fig.7 shows the average of the 10 spatial filter weights for each dataset. We can find that the average spatial filter has higher values in the parietal and occipital regions, which is consistent with the results in [23] and [24]. The ranking of best 8 electrodes for the datasets IIb, II-A, II-B are shown in Table XI. The electrodes are arranged in descending order of absolute values of the averaged 10 spatial filters. The common electrodes between the three datasets are PO7, PO8, O1, CPz, Pz, which is in general agreement with the results in [23].

TABLE XI: The ranking of best 8 electrodes for datasets IIb, II-A, and II-B.

Ranking	1	2	3	4	5	6	7	8
IIb	PO7	P8	PO8	O1	Oz	CP1	CPz	Pz
II-A	Pz	PO7	PO8	POz	O1	CPz	Cz	FC1
II-B	PO8	PO7	O1	PO4	PO3	Pz	CPz	POz

Our approach illustrates that extracting good spatial and temporal features is crucial for the classification of EEG signals, as reported by others. For example, the deep subject-adapted convolutional neural network (SACNN) by Liu et al. uses parallel multiscale convolutional networks to extract temporal and spatial features from raw EEG data and achieve good classification accuracy [46]. Despite the excellent performance of ST-CapsNet in P300 detection, the method has some shortcomings. The capsule network model has a relatively large number of parameters compared to traditional methods and CNNs which means it needs longer training time and requires higher performance equipment. Although ST-CapsNet is able to achieve higher accuracy of symbol recognition at low repetitions, we are not able to precisely control the recognition accuracy at a single repetition. Because P300 detection in single trial and symbol recognition are two tasks, and our model and loss function are designed for the first task without a well-designed training method for the second task. In the future, we will look for a better approach in terms of reducing the number of parameters in the model and designing a separate training method for the symbol recognition task.

VI. CONCLUSION

In this study, we proposed a novel deep-learning analysis framework—ST-CapsNet to enhance the performance of P300 detection. Specifically, instead of sending EEG signals directly to the capsule network, the complex spatio-temporal characteristics of EEG signals were initially extracted through spatial and temporal attention modules, which were served as inputs to the capsule network for P300 detection. On this account, the spatial and temporal of P300 features could be attained. Subsequent performance evaluation was conducted on two publicly-available datasets that reveals superiority of the proposed ST-CapsNet in both single-trial P300 detection and cumulative effect under different repetitions (i.e., better ASUR). Within this context, our results demonstrate the beneficial effect of adding attention mechanisms to the capsule network in P300 speller, which may lead to new directions for developing better P300-based BCI communication system.

REFERENCES

[1] B. Rebsamen, C. Guan, H. Zhang, C. Wang, C. Teo, M. H. Ang, and E. Burdet, "A brain controlled wheelchair to navigate in familiar environments," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, no. 6, pp. 590–598, 2010.

[2] J. Li, J. Liang, Q. Zhao, J. Li, K. Hong, and L. Zhang, "Design of assistive wheelchair system directly steered by human thoughts," *International journal of neural systems*, vol. 23, no. 03, p. 1350013, 2013.

[3] J. Long, Y. Li, T. Yu, and Z. Gu, "Target selection with hybrid feature for bci-based 2-d cursor control," *IEEE Transactions on biomedical engineering*, vol. 59, no. 1, pp. 132–140, 2011.

[4] J. Long, Y. Li, H. Wang, T. Yu, J. Pan, and F. Li, "A hybrid brain computer interface to control the direction and speed of a simulated or real wheelchair," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 20, no. 5, pp. 720–729, 2012.

[5] Y. Wang, R. Wang, X. Gao, B. Hong, and S. Gao, "A practical vep-based brain-computer interface," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 14, no. 2, pp. 234–240, 2006.

[6] C.-T. Lin, R.-C. Wu, S.-F. Liang, W.-H. Chao, Y.-J. Chen, and T.-P. Jung, "Eeg-based drowsiness estimation for safety driving using independent component analysis," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 12, pp. 2726–2738, 2005.

[7] L. Zheng, W. Pei, X. Gao, L. Zhang, and Y. Wang, "A high-performance brain switch based on code-modulated visual evoked potentials," *Journal of Neural Engineering*, vol. 19, no. 1, p. 016002, 2022.

[8] T. Burns and R. Rajan, "Combining complexity measures of eeg data: multiplying measures reveal previously hidden information," *F1000Research*, vol. 4, 2015.

[9] T. Ball, M. Kern, I. Mutschler, A. Aertsen, and A. Schulze-Bonhage, "Signal quality of simultaneously recorded invasive and non-invasive eeg," *Neuroimage*, vol. 46, no. 3, pp. 708–716, 2009.

[10] S. J. Luck, *An introduction to the event-related potential technique*. MIT press, 2014.

[11] E. Başar, C. Başar-Eroglu, B. Rosen, and A. Schütt, "A new approach to endogenous event-related potentials in man: relation between eeg and p300-wave," *International Journal of Neuroscience*, vol. 24, no. 1, pp. 1–21, 1984.

[12] B. R. Dunn, D. A. Dunn, M. Languis, and D. Andrews, "The relation of erp components to complex memory processing," *Brain and cognition*, vol. 36, no. 3, pp. 355–376, 1998.

[13] M. G. Coles and M. D. Rugg, *Event-related brain potentials: An introduction.* Oxford University Press, 1995.

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

[15] A. Rakotomamonjy and V. Guigue, "Bci competition iii: dataset ii-ensemble of svms for bci p300 speller," *IEEE transactions on biomedical engineering*, vol. 55, no. 3, pp. 1147–1154, 2008.

[16] D. Krusienski and G. Schalk, "Bci competition iii challenge 2004," 2004.

[17] B. Rivet, A. Souloumiac, V. Attina, and G. Gibert, "xdawn algorithm to enhance evoked potentials: application to brain–computer interface," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 8, pp. 2035–2043, 2009.

[18] A. Barachant, "Meg decoding using riemannian geometry and unsupervised classification," *Grenoble University: Grenoble, France*, 2014.

[19] H. Zhang, Z. Wang, Y. Yu, H. Yin, C. Chen, and H. Wang, "An improved eegnet for single-trial eeg classification in rapid serial visual presentation task," *Brain Science Advances*, vol. 8, no. 2, pp. 111–126, 2022.

[20] 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," *Journal of neural engineering*, vol. 15, no. 5, p. 056013, 2018.

[21] B. Liu, X. Chen, Y. Wang, and X. Gao, "Promoting brain–computer interface in china by bci controlled robot contest in world robot contest," 2022.

[22] H. Wang, H. Huang, Y. Liu, H. Xu, and T. Li, "An event related potential electroencephalogram signal analysis method based on denoising auto-encoder neural network," *Control Theory and Applications*, vol. 36, no. 4, pp. 589–595, 2019.

[23] H. Cecotti and A. Graser, "Convolutional neural networks for p300 detection with application to brain-computer interfaces," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 3, pp. 433–445, 2010.

[24] H. Wang, Z. Pei, L. Xu, T. Xu, A. Bezerianos, Y. Sun, and J. Li, "Performance enhancement of p300 detection by multiscale-cnn," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021.

[25] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *Advances in neural information processing systems*, vol. 30, 2017.

[26] T. Wang, A. Bezerianos, A. Cichocki, and J. Li, "Multikernel capsule network for schizophrenia identification," *IEEE Transactions on Cybernetics*, pp. 1–10, 2020.

[27] H. Chao, L. Dong, Y. Liu, and B. Lu, "Emotion recognition from multiband eeg signals using capsnet," *Sensors*, vol. 19, no. 9, p. 2212, 2019.

[28] X. Liu, Q. Xie, J. Lv, H. Huang, and W. Wang, "P300 event-related potential detection using one-dimensional convolutional capsule networks," *Expert Systems with Applications*, vol. 174, p. 114701, 2021.

[29] R. Ma, T. Yu, X. Zhong, Z. L. Yu, Y. Li, and Z. Gu, "Capsule network for erp detection in brain-computer interface," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 718–730, 2021.

[30] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

[31] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

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

[33] B. Blankertz, M. Krauledat, G. Dornhege, J. Williamson, R. Murray-Smith, and K.-R. Müller, "A note on brain actuated spelling with the berlin brain-computer interface," in *International Conference on Universal Access in Human-Computer Interaction*, pp. 759–768, Springer, 2007.

[34] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[35] J. Yin, S. Li, H. Zhu, and X. Luo, "Hyperspectral image classification using capsnet with well-initialized shallow layers," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 7, pp. 1095–1099, 2019.

[36] Z. Chiyuan, "Lossfunctions.jl."

[37] R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball, "Deep learning with convolutional neural networks for eeg decoding and visualization," *Human Brain Mapping*, aug 2017.

[38] B. Blankertz, S. Lemm, M. Treder, S. Haufe, and K.-R. Müller, "Single-trial analysis and classification of erp componentsa tutorial," *NeuroImage*, vol. 56, no. 2, pp. 814–825, 2011.

[39] A. D. Gerson, L. C. Parra, and P. Sajda, "Cortically coupled computer vision for rapid image search," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 14, no. 2, pp. 174–179, 2006.

[40] G. F. Alpert, R. Manor, A. B. Spanier, L. Y. Deouell, and A. B. Geva, "Spatiotemporal representations of rapid visual target detection: A single-trial eeg classification algorithm," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 8, pp. 2290–2303, 2013.

[41] S. Kundu and S. Ari, "Mscnn: a deep learning framework for p300-based brain–computer interface speller," vol. 2, no. 1, pp. 86–93, 2019.

[42] M. Liu, W. Wu, Z. Gu, Z. Yu, F. Qi, and Y. Li, "Deep learning based on batch normalization for p300 signal detection," *Neurocomputing*, vol. 275, pp. 288–297, 2018.

[43] S. Kundu and S. Ari, "P300 detection using ensemble of svm for brain-computer interface application," in *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–5, IEEE, 2018.

[44] W. Kong, S. Guo, Y. Long, Y. Peng, H. Zeng, X. Zhang, and J. Zhang, "Weighted extreme learning machine for p300 detection with application to brain computer interface," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–11, 2018.

[45] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, and M. S. Hämäläinen, "MEG and EEG data analysis with MNE-Python," *Frontiers in Neuroscience*, vol. 7, no. 267, pp. 1–13, 2013.

[46] S. Liu, J. Zhang, A. Wang, H. Wu, Q. Zhao, and J. Long, "Subject adaptation convolutional neural network for eeg-based motor imagery classification," *Journal of Neural Engineering*, vol. 19, p. 066003, nov 2022.