


# Adaptive Spatiotemporal Graph Convolutional Networks for Motor Imagery Classification

Biao Sun , *Member, IEEE*, Han Zhang , Zexu Wu, Yunyan Zhang, *Member, IEEE*, and Ting Li, *Member, IEEE*

**Abstract**—Classification of electroencephalogram-based motor imagery (MI-EEG) tasks is crucial in brain computer interfaces (BCI). In view of the characteristics of non-stationarity, time-variability and individual diversity of EEG signals, a novel framework based on graph neural network is proposed for MI-EEG classification. First, an adaptive graph convolutional layer (AGCL) is constructed, by which the electrode channel information are integrated dynamically. We further propose an adaptive spatiotemporal graph convolutional network (ASTGCN), which fully exploits the characteristics of EEG signals in time domain and the channel correlations in spatial domain simultaneously. We execute the experiments using EEG signals recorded at motor imagery scenarios, where twenty-five healthy subjects performed MI movements of the right hand and feet to generate motor commands. Experimental results reveal that the proposed method outperforms state-of-the-art methods in terms of both classification quality and robustness. The advantages of ASTGCN include high accuracy, high efficiency, and robustness to cross-trial and cross-subject variations, making it an ideal candidate for long-term MI-EEG applications.

**Index Terms**—Brain computer interfaces (BCI), electroencephalogram (EEG), motor imagery (MI), graph neural network, spatiotemporal structure.

## I. INTRODUCTION

CLASSIFICATION of electroencephalogram-based motor imagery (MI-EEG) tasks raises a big challenge in the design and development of brain computer interfaces (BCIs). Due to the non-stationarity, time-variability, and individual diversity of EEG signals, traditional machine learning approaches like support vector machine (SVM) [1] and Bayesian classifiers [2] have limitations for obtaining high classification performance.

Manuscript received November 19, 2020; accepted December 26, 2020. Date of publication January 8, 2021; date of current version February 3, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61971303 and Grant 81971660, in part by the Tianjin Outstanding Youth Fund Project under Grant 20JCJC00230, in part by the Program of Chinese Institute for Brain Research in Beijing under Grant 2020-NKX-XM-14, in part by the Beijing Major Science and Technology Project under Grant Z191100010618004, in part by the Tianjin Key Project under Grant 18JCZDJC32700, and in part by Sichuan International Cooperation under Grant 2021YFH0004. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xun Chen. (*Corresponding author: Ting Li.*)

Biao Sun, Han Zhang, and Zexu Wu are with the School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: sunbiao@tju.edu.cn; zhanghanup137@163.com; wuzexuxuexi@163.com).

Yunyan Zhang is with the Department of Physics, Paderborn University, 33098 Paderborn, Germany (e-mail: bsuntju@gmail.com).

Ting Li is with the Institute of Biomedical Engineering, Chinese Academy of Medical Science & Pecking Union Medical College, Tianjin 300020, China (e-mail: t.li619@foxmail.com).

Digital Object Identifier 10.1109/LSP.2021.3049683

Many papers have been published that use deep neural networks for EEG-based MI tasks [3]–[5]. An *et al.* proposed a deep belief network (DBN) for two-class MI classification and outperformed the SVM method [6]. Yang *et al.* proposed a frequency complementary feature map selection method with a five-layer convolutional neural network (CNN) for EEG-based MI classification, and demonstrated that CNN is capable of learning discriminant, deep structure information for MI classification without relying on the handcrafted features [7]. Bashivan *et al.* proposed a multi-dimensional feature extraction method and employed a recurrent neural network (RNN) architecture to model cognitive events from EEG data [8]. Jirayucharoensak *et al.* proposed an autoencoder (AE) architecture for automatic emotion recognition using EEG signals [9]. Tabar *et al.* proposed a hybrid architecture by combining CNN and AE for EEG MI classification and obtained convincing performance [10]. The recently proposed EEGNet adopted a different scheme by combining feature extraction and classification into one pipeline, and obtained state-of-the-art results in several BCI paradigms [11]. Dai *et al.* proposed an HS-CNN method, which achieved average classification accuracies of 91.57% and 87.6% on two public datasets [12]. Recently, the emerging field of graph neural networks (GNN) has shown potential in modeling multichannel signals by graphs in non-Euclidean space. Immediate applications are as diverse as image classification [13]–[16], node classification [17], action recognition [18], [19], and traffic forecasting [20], [21]. The ChebNet proposed by Defferrard *et al.* introduced the Chebyshev polynomial to polynomial parameterize the graph convolution operation, which dramatically accelerates the running speed of graph convolution while ensuring performance [14]. The graph convolutional network (GCN) proposed by Kipf *et al.* is a further improvement based on ChebNet. As the first-order approximation of ChebNet, GCN not only improves the performance of large-scale neural networks but also improves scalability [17]. The performance of spatiotemporal graph convolutional networks (ST-GCN) proposed by Yan *et al.* has been dramatically improved compared to other mainstream methods in skeleton-based action recognition tasks [18]. The dynamic spatiotemporal graph convolutional neural networks (DSTGCNN) proposed by Diao *et al.* have outstanding performance in traffic forecasting tasks [21]. However, despite the recent enthusiasm for this area, the study of developing GNN models for MI-EEG classification tasks remains almost untouched. Moreover, most existing GNN models cannot capture the full information of the EEG signals, and, as we demonstrate in this work, can produce poor results

when testing on real BCI applications. Hence, it is becoming increasingly evident that novel models need to be constructed with better classification accuracy and robustness.

In this letter, we propose a GNN framework for MI-EEG classification, based on the spatiotemporal structure of multi-channel EEG signals. The GNN framework is termed as adaptive spatiotemporal graph convolutional network (ASTGCN), which provides ideas for the subsequent exploration of the channel relationship of the MI-EEG and the improvement of classification performance. In traditional GCN, the graph convolution operation treats each neighbor node with equal importance. In contrast, the proposed method adaptively determines the importance of neighbor nodes and extracts the features in the time domain at the same time.

## II. METHODS

Graph convolution applies the convolution operation from grid data in Euclidean space to graph data in non-Euclidean space. The graph is expressed as  $G = (V, E)$ ,  $V$  is the set of nodes in the graph,  $V_i \in V$  represents a node, the number is  $N$ ,  $(V_i, V_j) \in E$  represents the edges of the graph. Adjacency matrix  $\mathbf{A} \in \mathbb{R}^{N \times N}$  is used to represent the connection relationship between nodes. For an undirected graph, if there is a connection relationship between  $V_i$  and  $V_j$ , then  $A_{ij}$  is equal to 1, otherwise,  $A_{ij}$  is equal to 0. However, for some data, the influence of neighbor nodes on the central nodes is different, and the weights of edges are unknown, so it is not appropriate to describe them by the undirected graph. Therefore, we proposed an adaptive graph convolutional layer (AGCL) to learn the weights on edges adaptively.

The Laplacian matrix of graph  $G$  is defined as:

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \in \mathbb{R}^{N \times N}, \quad (1)$$

where the diagonal matrix  $\mathbf{D}$  is the degree matrix of graph,  $D_{ij} = \sum_j A_{ij}$ . Since  $\mathbf{L}$  is a real symmetric matrix, it can be orthogonalized and diagonalized to get the following formula:

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T, \quad (2)$$

where the orthonormal matrix  $\mathbf{U}$  is the eigenvector matrix of  $\mathbf{L}$ , and  $\mathbf{U}\mathbf{U}^T = \mathbf{I}_N$  is the  $N \times N$  identity matrix,  $\mathbf{\Lambda} = \text{diag}([\lambda_0, \dots, \lambda_{N-1}])$  is the eigenvalue matrix of  $\mathbf{L}$ .

Assuming the signal  $\mathbf{x} \in \mathbb{R}^N$ , the graph Fourier transform is defined as:

$$\hat{\mathbf{x}} = \mathbf{U}^T \mathbf{x}, \quad (3)$$

where  $\hat{\mathbf{x}}$  is the signal obtained after the signal  $\mathbf{x}$  undergoes the graph Fourier transform. The inverse graph Fourier transform is as follows:

$$\mathbf{x} = \mathbf{U}\hat{\mathbf{x}} = \mathbf{U}\mathbf{U}^T \mathbf{x}. \quad (4)$$

Graph convolution operation of two signals is defined as:

$$\begin{aligned} \mathbf{x}_1 *_{\mathcal{G}} \mathbf{x}_2 &= \mathbf{U}((\mathbf{U}^T \mathbf{x}_1) \odot (\mathbf{U}^T \mathbf{x}_2)) \\ &= \mathbf{U}(\hat{\mathbf{x}}_1 \odot (\mathbf{U}^T \mathbf{x}_2)) \\ &= \mathbf{U}(\text{diag}(\hat{\mathbf{x}}_1)(\mathbf{U}^T \mathbf{x}_2)) \\ &= \mathbf{U}\text{diag}(\hat{\mathbf{x}}_1)\mathbf{U}^T \mathbf{x}_2, \end{aligned} \quad (5)$$

where  $\odot$  represents the Hadamard product.

Set filter function  $g_{\theta} = \text{diag}(\theta)$ , the convolution operation is:

$$g_{\theta} *_{\mathcal{G}} \mathbf{x} = \mathbf{U}\text{diag}(\theta)\mathbf{U}^T \mathbf{x}. \quad (6)$$

Let  $g_{\theta}$  be the function  $g_{\theta}(\mathbf{\Lambda})$  of the eigenvalue matrix of Laplace matrix  $\mathbf{L}$ . By using  $K$  order Chebyshev polynomial approximation, it can be obtained as follows:

$$g_{\theta}(\mathbf{\Lambda}) = \sum_{k=0}^K \theta_k T_k(\tilde{\mathbf{\Lambda}}), \quad (7)$$

$$\tilde{\mathbf{\Lambda}} = \frac{2}{\lambda_{\max}} \mathbf{\Lambda} - \mathbf{I}_N. \quad (8)$$

In the above formula,  $\lambda_{\max}$  is the largest eigenvalue of  $\mathbf{L}$ , and  $\theta$  is the coefficient of Chebyshev polynomial. And the Chebyshev polynomial can be recursively defined as:

$$\begin{cases} T_0(\tilde{\mathbf{\Lambda}}) = 1, \\ T_1(\tilde{\mathbf{\Lambda}}) = \tilde{\mathbf{\Lambda}}, \\ T_k(\tilde{\mathbf{\Lambda}}) = 2\tilde{\mathbf{\Lambda}}T_{k-1} - T_{k-2}(\tilde{\mathbf{\Lambda}}), \quad k > 1. \end{cases} \quad (9)$$

According to (7) and (9), (6) can be rewritten as:

$$g_{\theta} *_{\mathcal{G}} \mathbf{x} = \sum_{k=0}^K \theta_k T_k(\tilde{\mathbf{\Lambda}}). \quad (10)$$

Let the Chebyshev polynomial order  $K = 1$  and approximate  $\lambda_{\max}$  to 2. The convolution operation is as follows:

$$\begin{aligned} g_{\theta} *_{\mathcal{G}} \mathbf{x} &= \theta_0 \mathbf{x} + \theta_1 (\mathbf{L} - \mathbf{I}_N) \mathbf{x} \\ &= \theta_0 \mathbf{x} - \theta_1 \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \mathbf{x}. \end{aligned} \quad (11)$$

Formula (11) has two trainable parameters, by using  $\theta = \theta_0 = -\theta_1$  to further simplify (11), we can obtain:

$$g_{\theta} *_{\mathcal{G}} \mathbf{x} = \theta (\mathbf{I}_N + \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}) \mathbf{x}. \quad (12)$$

To avoid the gradient disappearing/exploding problem, we normalize  $\mathbf{I}_N + \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$  with  $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$  and  $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ , and the graph convolution operation is expressed as:

$$g_{\theta} *_{\mathcal{G}} \mathbf{x} = \theta (\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}) \mathbf{x}. \quad (13)$$

When extending the input signal from the spatial domain to the spatiotemporal domain, the graph convolution operation becomes:

$$\mathbf{H}_t = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{X}_t \theta_t, \quad (14)$$

where  $\mathbf{X}_t \in \mathbb{R}^N$  is the signal at the time point  $t$ ,  $\mathbf{H}_t$  is the output of graph convolution, and  $\theta$  is a scalar. To adaptively learn the weights on edges, we add a trainable adaptive matrix  $\mathbf{W}$  to adjust the normalized adjacency matrix dynamically and obtain the AGCL formula:

$$\mathbf{H}_t = ((\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}) \odot \mathbf{W}) \mathbf{X}_t \theta_t + b_t. \quad (15)$$

where  $b_t$  is the bias of the adaptive graph convolution at time  $t$ . Let adaptive adjacency matrix  $\hat{\mathbf{A}} = (\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}) \odot \mathbf{W}$ , the AGCL formula is:

$$\mathbf{H}_t = \hat{\mathbf{A}} \mathbf{X}_t \theta_t + b_t. \quad (16)$$

For time-dependent graph data, it is necessary to extract features in the time domain, so we add a temporal feature

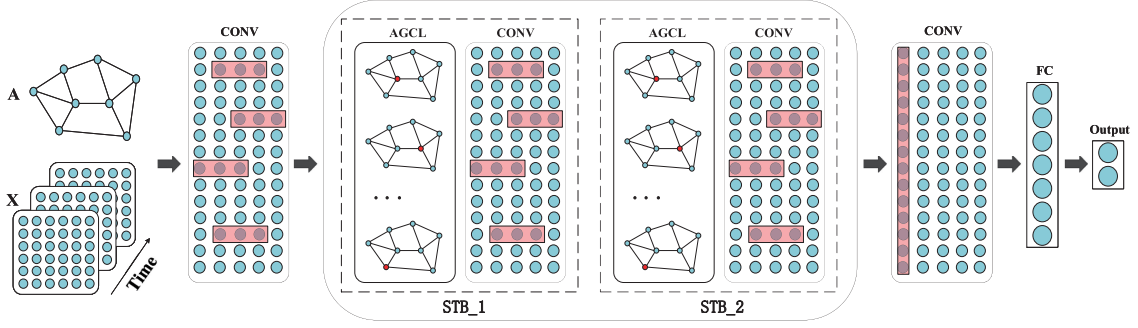


Fig. 1. The proposed adaptive spatiotemporal graph convolutional neural network architecture with the AGCL and spatiotemporal blocks.

extraction procedure based on the AGCL to form an adaptive spatiotemporal graph convolutional network (ASTGCN). The network structure is shown in Fig. 1. We call the AGCL and the convolutional layer used for temporal feature extraction as a spatiotemporal block (STB).

The kernel size of the first convolutional layer is (1,64), the stride is (1,1). Assuming that the input data dimension is  $(N, T)$  and the adjacency matrix dimension is  $(N, N)$ , where  $N$  is the number of nodes in the input graph. The feature map dimension obtained by the first convolution operation remains unchanged. An AGCL in STB performs adaptive graph convolution operation to aggregate the information of the central node and its neighbors, and does not change the dimension of the feature map. The kernel size of the convolutional layer in STB is (1,8), and the stride is (1,1), average pooling size is (1,2), the feature map dimension obtained by the average pooling is  $(N, T/2)$ . The dimension of the feature map obtained after two STBs is  $(N, T/4)$ . The kernel size of the next convolutional layer is  $(N, 1)$ , and the stride is (1,1). This layer combines the information of all nodes, which reduces the feature map dimension to  $(1, T/4)$ . After the (1,4) average pooling operation, we obtain the feature map with dimension  $(1, T/16)$ . Finally, classification is performed through the fully connected layer. The number of convolution kernels in the last convolution layer is 32, and 16 in other layers. The training parameters are initialized with Gaussian distribution.

Except the output layer which uses the Softmax activation function, the other layers use the ELU activation function defined as:

$$\text{ELU}(x) = \begin{cases} x, & x > 0 \\ \alpha * (e^x - 1), & x < 0. \end{cases} \quad (17)$$

Since ASTGCN is applied to the graph level binary classification task, the binary classification cross-entropy loss function is used, and the batch size is 20. The robust normalization strategy is adopted to accelerate the convergence. We use dropout to avoid overfitting. We adopt Adam optimizer with a learning rate of 0.001 for network training.

### III. EXPERIMENTS AND RESULTS

#### A. Data Acquisition and Preprocessing

Subjects for this study were twenty-five right-handed students (twelve males and thirteen females, mean age: 25.3 years,

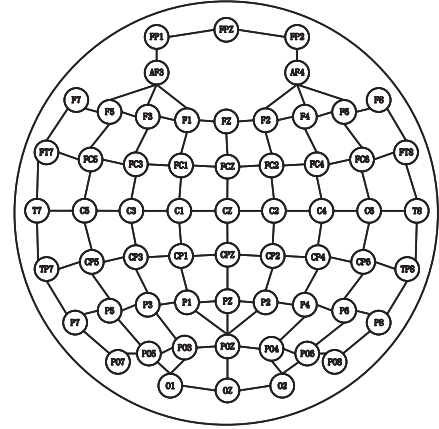


Fig. 2. Electrode channel connection diagram. Convert data into graph according to this diagram.

standard deviation (STD): 1.69 years, range: 19-32 years) recruited from Tianjin University. None of them had any history of neurological diseases. All EEG recording procedures were approved by the China Rehabilitation Research Center Ethics Committee (No. CRRC-IEC-RF-SC-005-01).

The EEG data acquisition uses the Neuroscan system, and the signal recording device has 64 Ag/AgCl scalp electrodes, which are arranged according to the international standard 10/20 system. The sampling frequency adopted in the experiment is 1000 Hz. Of the 64 electrodes, all but two reference electrodes and two electrodes that monitor eye movement can be used for data analysis.

The preprocessing steps included re-reference, artifact correction, filter, epoch, baseline correction, and artifact rejection. Then, we divided the signals into samples of 7-second trials without overlap. There are 320 trials per subject, of which the number for each class is 160. Specifically, downsampling the EEG signal to 128 Hz is beneficial for subsequent data analysis and helps reduce the need for computing resources. Since the frequency range associate with motor imagery is 8-30 Hz, we apply band-pass filtering on the EEG signals.

Next, the data of each trial is converted into a graph. Each node and its value on the graph represent the corresponding electrode channel and signal, edges represent the connection between the electrode channels, and the electrode channel connection diagram is shown in Fig. 2. So each subject gets 320 graphs, and each graph corresponds to its label.



TABLE I  
CLASSIFICATION ACCURACY (%) AND STANDARD DEVIATION (STD) RESULTS  
FOR CNN-SAE, EEGNET, ASTGCN AND STGCN METHODS

Subject	Accuracy % (mean $\pm$ std)			
	CNN-SAE	EEGNet	STGCN	ASTGCN
No.1	78.8 $\pm$ 4.8	85.6 $\pm$ 4.0	89.1 $\pm$ 6.1	<b>95.9<math>\pm</math>4.2</b>
No.2	79.7 $\pm$ 5.3	<b>98.8<math>\pm</math>1.8</b>	91.4 $\pm$ 4.7	96.7 $\pm$ 3.7
No.3	73.8 $\pm$ 7.7	95.3 $\pm$ 2.1	93.4 $\pm$ 3.3	<b>96.6<math>\pm</math>2.2</b>
No.4	90.6 $\pm$ 8.2	<b>96.5<math>\pm</math>5.0</b>	83.6 $\pm$ 9.2	85.6 $\pm$ 8.9
No.5	73.0 $\pm$ 5.9	93.7 $\pm$ 4.2	91.5 $\pm$ 4.1	<b>99.7<math>\pm</math>0.9</b>
No.6	71.3 $\pm$ 4.6	73.2 $\pm$ 6.4	76.9 $\pm$ 5.8	<b>77.2<math>\pm</math>4.2</b>
No.7	71.5 $\pm$ 5.3	71.8 $\pm$ 5.2	77.7 $\pm$ 3.5	<b>77.7<math>\pm</math>3.5</b>
No.8	74.2 $\pm$ 5.7	90.5 $\pm$ 4.2	90.2 $\pm$ 2.2	<b>96.2<math>\pm</math>3.6</b>
No.9	73.1 $\pm$ 4.7	<b>76.6<math>\pm</math>4.3</b>	75.9 $\pm$ 5.7	76.3 $\pm$ 3.7
No.10	73.2 $\pm$ 3.2	92.2 $\pm$ 3.8	96.1 $\pm$ 3.0	<b>99.6<math>\pm</math>1.2</b>
No.11	73.4 $\pm$ 6.6	87.9 $\pm$ 5.8	93.4 $\pm$ 4.2	<b>98.5<math>\pm</math>2.5</b>
No.12	75.8 $\pm$ 5.4	90.5 $\pm$ 7.9	91.6 $\pm$ 5.5	<b>99.5<math>\pm</math>1.5</b>
No.13	76.2 $\pm$ 4.1	81.7 $\pm$ 5.2	92.9 $\pm$ 3.7	<b>97.1<math>\pm</math>2.6</b>
No.14	66.6 $\pm$ 6.1	69.1 $\pm$ 5.8	74.7 $\pm$ 4.7	<b>75.3<math>\pm</math>2.9</b>
No.15	72.3 $\pm$ 5.5	74.0 $\pm$ 6.2	78.6 $\pm$ 3.5	<b>80.0<math>\pm</math>3.8</b>
No.16	93.8 $\pm$ 11.4	95.3 $\pm$ 7.8	94.7 $\pm$ 9.2	<b>97.2<math>\pm</math>6.0</b>
No.17	95.9 $\pm$ 5.4	98.1 $\pm$ 5.6	96.9 $\pm$ 4.6	<b>98.1<math>\pm</math>4.7</b>
No.18	65.7 $\pm$ 6.0	67.0 $\pm$ 5.2	77.3 $\pm$ 5.0	<b>81.1<math>\pm</math>3.7</b>
No.19	81.9 $\pm$ 5.7	88.8 $\pm$ 4.9	88.4 $\pm$ 4.0	<b>93.1<math>\pm</math>4.1</b>
No.20	73.4 $\pm$ 2.5	91.6 $\pm$ 6.4	94.7 $\pm$ 3.1	<b>97.5<math>\pm</math>2.7</b>
No.21	64.1 $\pm$ 4.7	96.6 $\pm$ 2.2	99.4 $\pm$ 1.3	<b>100.0<math>\pm</math>0.0</b>
No.22	68.1 $\pm$ 5.7	75.0 $\pm$ 7.4	83.8 $\pm$ 3.9	<b>89.4<math>\pm</math>3.5</b>
No.23	68.8 $\pm$ 5.8	65.9 $\pm$ 6.0	<b>73.4<math>\pm</math>3.2</b>	72.5 $\pm$ 3.6
No.24	67.8 $\pm$ 4.4	97.7 $\pm$ 2.1	<b>99.7<math>\pm</math>1.0</b>	99.4 $\pm$ 1.3
No.25	69.7 $\pm$ 7.0	68.4 $\pm$ 5.7	78.5 $\pm$ 5.0	<b>84.9<math>\pm</math>4.8</b>
Average	74.9 $\pm$ 5.7	84.9 $\pm$ 5.0	87.4 $\pm$ 4.4	<b>90.6<math>\pm</math>3.4</b>
Inter-subject std	8.2	11.3	8.4	9.5

## B. Results

In this work, the model performance is evaluated in a python 3.5 environment. Computer hardware resources include 32G RAM, 3.60 GHz Intel Core I7 CPU, and an NVIDIA GPU (Titan Xp). Pytorch [22] is used for designing and testing the proposed network. For each subject, the classifier uses the data separately to train the network and test the network performance. The performance of the proposed method is evaluated using the classification accuracy metric. Two baseline methods including convolutional neural networks with stacked autoencoders (CNN-SAE) [10] and EEGNet [11] are chosen for performance comparison with the proposed ASTGCN method. 10-fold cross-validation is used in each session, i.e., the trials are randomly partitioned into ten equal-sized parts. Of the ten parts, a single part is retained as the validation data for testing the method, while the remaining nine parts are used as the training set. The cross-validation process is repeated ten times, with each of the ten parts used exactly once as the validation data. After the repetitions, the ten results are averaged to produce a single estimate.

To evaluate the proposed ASTGCN, we first compare it with CNN-SAE and EEGNet on datasets of all 25 subjects. The experimental results are shown in Table I. We observe that ASTGCN outperforms the other two baseline methods, where the average accuracies of ASTGCN, CNN-SAE, and EEGNet are 90.6%, 74.9%, and 84.9% respectively. The results indicate that ASTGCN provides a 30.0% improvement with respect to

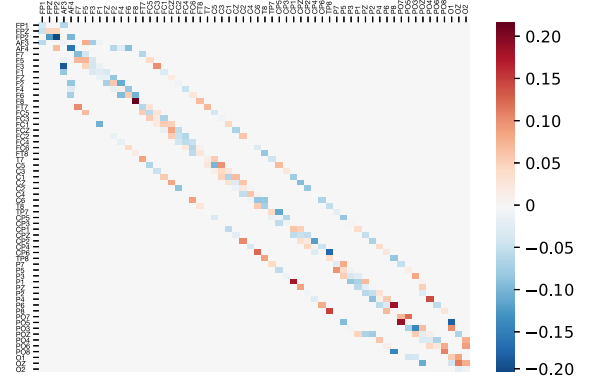


Fig. 3. The heatmap of adaptive adjacency matrix  $\hat{A}$ .

CNN-SAE, and a 6.7% improvement with respect to EEGNet in terms of average accuracy. Especially, ASTGCN consistently outperforms CNN-SAE for 24 out of 25 subjects. It also outperforms EEGNet in 22 out of 25 subjects. Although ASTGCN cannot outperform baseline methods for some subjects which is mainly due to the individual differences in EEG signals, the average standard deviation of accuracy for ASTGCN is 3.4, which is less than that of CNN-SAE (std = 5.7) and EEGNet (std = 5.0), demonstrating that ASTGCN is more robust to cross-trial variation than baselines. Inter-subject standard deviations of the three methods are also shown in Table I. We observe that the inter-subject standard deviations of GNN-based methods are lower than that of baseline methods, indicating the robustness to cross-subject variation of the proposed method. Although the inter-subject standard deviation of CNN-SAE is slightly lower than that of GNN-based methods, the poor classification accuracy impedes its application to real BCI scenarios.

To evaluate the importance of AGCL in ASTGCN, we consider another case, i.e., replacing AGCL with a typical graph convolutional layer (termed as STGCN) for performance comparison. We evaluate the classification accuracies in the case for all subjects and show the results in Table I. We observe that if the two AGCL are replaced, the performance decreases. The average accuracy of ASTGCN is  $90.6 \pm 3.4\%$ , indicating 3.7% improvements compared to STGCN ( $87.4 \pm 4.4\%$ ). We saved the parameters of the ASTGCN model, calculated the value of  $\hat{A}$ , and then drew the heatmap of  $\hat{A}$ , as shown in Figure 3. Results show that AGCL adaptively adjusts the weights on the edges, thus making ASTGCN more flexible and improving the performance of the model [23].

## IV. CONCLUSION

This letter proposes a graph neural network-based framework for MI-EEG classification. The proposed method has the capability of aggregating information from EEG channels dynamically and extracting the time-domain feature simultaneously, thus enhancing the classification performance. Experimental results on 25 healthy subjects show that the proposed method has high classification accuracy and robustness to cross-trial and cross-subject variations.

## REFERENCES

- [1] S. Siuly and Y. Li, "Improving the separability of motor imagery EEG signals using a cross correlation-based least square support vector machine for brain-computer interface," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 20, no. 4, pp. 526–538, Apr. 2012.
- [2] M. Miao, H. Zeng, A. Wang, C. Zhao, and F. Liu, "Discriminative spatial-frequency-temporal feature extraction and classification of motor imagery EEG: An sparse regression and weighted naïve Bayesian classifier-based approach," *J. Neurosci. Methods*, vol. 278, pp. 13–24, 2017.
- [3] J. Fedjaev, "Decoding EEG brain signals using recurrent neural networks," Master's thesis, Dept. Neurosci. Syst. Theory Technische Universität München, München, Germany, 2017.
- [4] B. Sun, X. Zhao, H. Zhang, R. Bai, and T. Li, "EEG motor imagery classification with sparse spectrotemporal decomposition and deep learning," *IEEE Trans. Automat. Sci. Eng.*, early access, Sep. 17, 2020, doi: 10.1109/TASE.2020.3021456.
- [5] H. Zhang, X. Zhao, Z. Wu, B. Sun, and T. Li, "Motor imagery recognition with automatic EEG channel selection and deep learning," *J. Neural Eng.*, 2020. [Online]. Available: <http://iopscience.iop.org/article/10.1088/1741-2552/abca16>
- [6] X. An, D. Kuang, X. Guo, Y. Zhao, and L. He, "A deep learning method for classification of EEG data based on motor imagery," in *Proc. Int. Conf. Intell. Comput.*, 2014, pp. 203–210.
- [7] H. Yang, S. Sakhavi, K. K. Ang, and C. Guan, "On the use of convolutional neural networks and augmented CSP features for multi-class motor imagery of EEG signals classification," in *Proc. 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, 2015, pp. 2620–2623.
- [8] P. Bashivan, I. Rish, M. Yeasin, and N. Codella, "Learning representations from EEG with deep recurrent-convolutional neural networks," in *Proc. 4th Int. Conf. Learn. Representations (ICLR-16)*, 2016.
- [9] S. Jirayucharoensak, S. Pan-Ngum, and P. Israsena, "EEG-based emotion recognition using deep learning network with principal component based covariate shift adaptation," *Sci. World J.*, vol. 2014, 2014, Art. no. 627892.
- [10] Y. R. Tabar and U. Halici, "A novel deep learning approach for classification of eeg motor imagery signals," *J. Neural Eng.*, vol. 14, no. 1, 2016, Art. no. 016003.
- [11] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "EEGNet: A compact convolutional neural network for EEG-based brain-computer interfaces," *J. Neural Eng.*, vol. 15, no. 5, 2018, Art. no. 056013.
- [12] G. Dai, J. Zhou, J. Huang, and N. Wang, "HS-CNN: A CNN with hybrid convolution scale for EEG motor imagery classification," *J. Neural Eng.*, vol. 17, no. 1, 2020, Art. no. 16025.
- [13] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," *Learning*, 2013, *arXiv:1312.6203*.
- [14] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.
- [15] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5115–5124.
- [16] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5177–5186.
- [17] T. N. Kipf and M. Welling, "Semi-supervised classification with graph 340 convolutional networks," in *Proc. 5th Int. Conf. Learn Representations (ICLR-17)*, 2017, *arXiv:1609.02907*.
- [18] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," *AAAI*, vol. 32, no. 1, Apr. 2018.
- [19] H. Zhang, Y. Song, and Y. Zhang, "Graph convolutional LSTM model for skeleton-based action recognition," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2019, pp. 412–417.
- [20] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. 6th Int. Conf. Learn. Representations (ICLR-18)*, 2018.
- [21] Z. Diao, X. Wang, D. Zhang, Y. Liu, K. Xie, and S. He, "Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 890–897.
- [22] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.
- [23] Y. Yang, O. Kyrgyzov, J. Wiart, and I. Bloch, "Subject-specific channel selection for classification of motor imagery electroencephalographic data," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 1277–1280.