GCNs-Net: A Graph Convolutional Neural Network Approach for Decoding Time-Resolved EEG Motor Imagery Signals

Yimin Hou, Shuyue Jia, Student Member, IEEE, Xiangmin Lun, Ziqian Hao, Yan Shi, Member, IEEE, Yang Li, Senior Member, IEEE, Rui Zeng, and Jinglei Lv

Abstract—Toward the development of effective and efficient brain–computer interface (BCI) systems, precise decoding of brain activity measured by an electroencephalogram (EEG) is highly demanded. Traditionally works classify EEG signals without considering the topological relationship among electrodes. However, neuroscience research has increasingly emphasized network patterns of brain dynamics. Thus, the Euclidean structure of electrodes might not adequately reflect the interaction between signals. To fill the gap, a novel deep learning (DL) framework based on the graph convolutional neural networks (GCNs) is presented to enhance the decoding performance of raw EEG signals during different types of motor imagery (MI) tasks while cooperating with the functional topological relationship of electrodes. Based on the absolute Pearson’s matrix of overall signals, the graph Laplacian of EEG electrodes is built up. The GCNs-Net constructed by graph convolutional layers learns the generalized features. The followed pooling layers reduce dimensionality, and the fully-connected (FC) softmax layer derives the final prediction. The introduced approach has been shown to converge for both personalized and groupwise predictions. It has achieved the highest averaged accuracy, 93.06% and 88.57% (PhysioNet dataset), 96.24% and 80.89% (high gamma dataset), at the subject and group level, respectively, compared to existing studies, which suggests adaptability and robustness to individual variability. Moreover, the performance is stably reproducible among repetitive experiments for cross-validation. The excellent performance of our method has shown that it is an important step toward better BCI approaches. To conclude, the GCNs-Net filters EEG signals based on the functional topological relationship, which manages to decode relevant features for brain MI. A DL library for EEG task classification including the code for this study is open source at https://github.com/SuperBruceJia/EEG-DL for scientific research.

Index Terms—Brain–computer interface (BCI), deep learning (DL), electroencephalography, graph convolutional neural networks, motor imagery (MI).

I. INTRODUCTION

RECENTLY, the brain–computer interface (BCI) has become one of the hottest research topics for broad applications in the field of therapeutic and medical engineering [1]. It refers to the establishment of an innovative technology that exchanges information directly between the brain and the surroundings, which does not rely on traditional methods such as human muscle tissue or peripheral nerves. BCI systems decode brain activity patterns to manipulate assistant devices, such as wheelchairs and artificial limbs [2]. An electroencephalogram (EEG) is extensively applied because of its high temporal resolution, noninvasiveness, and portability. The principle of the EEG is to record spontaneous, event-related, and stimulus-evoked electrical signals of the brain on time scales, which reveals variations for different brain activities [3]. The EEG decodes discriminable brain patterns while carrying out different types of actual movement or imagery [4], [5]. The motor imagery (MI)-based EEG mentally simulates multiple motor motions, such as imagining hand or foot movements. Controlling machines via only the MI without physical movements of the body is one of the elemental jobs in BCI [6]. To realize such BCI systems, accurate classification of MI brain activity is of great essence. Although previous studies have shown promising performances, there is still space to improve the classification accuracy toward building effective and efficient BCI applications. For instance, the adaptability and robustness to individual variability remain among the challenges of setting up an EEG MI-based wheelchair. Traditional approaches do not consider the topological relationship of electrodes while decoding EEG signals. However, a growing number of neuroscience research studies have emphasized brain network dynamics [7], [8], [9]. Thus, the interaction between signals might not be adequately reflected and represented via the Euclidean structure of EEG electrodes. To address the concern, graph convolutional neural networks (GCNs) are introduced to decode EEG signals,...
promoting the classification performance by cooperating with the functional topological relationship of EEG electrodes and implementing the convolutional neural networks (CNNs) on graphs.

II. Literature Survey

Traditional works manually designed features from EEG signals, for example, via the analytic intrinsic mode functions or wavelet transforms, and then employed machine-learning-based approaches to classify features [10], [11], [12], [13], [14]. Recently, deep learning (DL) has achieved superhuman performances across multiple domains [15], [16], [17]. The DL-based methods learned the underlying features from signals, which alleviated the need for hands-on feature engineering. The CNNs have been broadly employed to classify the Euclidean-structured signals on account of their ability to learn informative features via the local receptive fields. The CNN-based approaches [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29] were implemented to address the challenge of EEG task classification. Hou et al. [20] introduced an innovative approach by combining the Scout EEG source imaging (ESI) and CNNs to decode EEG tasks, which achieved competitive results, 94.5% maximum accuracy for ten subjects, and 92.5% for 14 subjects, on the PhysioNet dataset [30]. Zhang et al. [31] presented a cascade convolutional recurrent neural network and obtained 98.31% averaged accuracy on the PhysioNet dataset. Dose et al. [22] applied 1-D convolutional filters to learn the temporal and spatial features, and it reached 80.38%, 69.82%, and 58.58% accuracy on the PhysioNet dataset with two, three, and four MI tasks. Amin et al. [21], Schirrmeister et al. [24], Li et al. [26], and Zhang et al. [27] utilized variants of CNNs to decode EEG signals from the BCI Competition IV-2a Dataset [32] and achieved 73.70%, 75.70%, 79.90%, and 83.00% accuracy, respectively. Amin et al. [21], Alazzari et al. [28], and Tang and Zhang [29] obtained 92.50%, 93.70%, and 95.4% accuracy, respectively, at the subject level on the high gamma dataset [33]. Although the performance of the above CNN-based models was encouraging, there was still space to promote the classification accuracy to build a robust and reliable BCI system. The reasons why we applied the GCNs were as follows. The traditional CNNs cannot directly process the non-Euclidean structured data because the discrete convolutions cannot keep translation invariance on the non-Euclidean signals. However, the GCNs can directly extract features from the non-Euclidean data and process the graph-structured signals since the GCNs consider the relationship properties (e.g., correlations) between nodes [34], [35]. Through a novel graph convolutional neural networks approach (GCNs-Net) on two EEG MI benchmarks, it has achieved dominant performances, 98.72% accuracy on the PhysioNet dataset, and 96.24% on the high gamma dataset, for EEG MI decoding, which were far ahead of the results produced by the CNN-based approaches. Moreover, more and more neuroscience research studies have suggested that the topological information promotes the analysis of brain network dynamics [7], [8], [9]. Although the Euclidean distance was one of the similarity measurements, it might be superior to decode EEG signals from the non-Euclidean perspective by taking into consideration of the functional topological relationship of electrodes (e.g., the correlations and degree properties between electrodes) to enhance the decoding performance of EEG tasks.

Considering the topological relationship of EEG electrodes, the graph in the non-Euclidean space was put forward [34], [36], [37], [38], [39], [40]. Researchers have explored the CNNs with the graph theory, intending to apply the convolutional operation on graphs. Two strategies were presented to define convolutions, that is, either from the spatial domain or from the spectral domain. At first, the spatial GCNs were proposed [41], [42], [43], [44]. However, Defferrard et al. [35] and Bruna et al. [45] indicated that they faced the challenge of matching the local neighborhoods. On the other hand, the spectral method provided a well-defined localized operator on graphs [46]. Thus, an innovative approach to the GCNs was proposed by implementing the CNNs based on the spectral graph theory. Specifically, Bruna et al. [45] first performed the GCNs from the spectral perspective. A few works [36], [37], [38], [39] have applied the above model to decode EEG tasks in the field of EEG-based emotion recognition. In detail, Zhang et al. [36] combined the GCNs with a broad learning system and put forward the graph convolutional broad network, which achieved 94.24% accuracy on the SJTU Emotion EEG Dataset [47]. Wang et al. [37] presented a phase-locking value-based GCNs. The dynamical GCNs were proposed by Song et al. [38], which can dynamically learn the topological relationship of EEG electrodes during training. Wang et al. [39] improved the above method via a broad learning system. Moreover, Hou et al. [34] introduced a two-stage method in which it first extracted the spatial-temporal EEG features via an attention-based bidirectional long short-term memory model and then employed the graph representation learning to classify deep features. Jia et al. [40] proposed an end-to-end approach, a graph residual network, to classify EEG signals. The highly competitive performance of these approaches [34], [36], [37], [38], [39], [40] indicated the superiority of applying the graph learning methods.

In this work, a novel structure of the GCNs is introduced to decode EEG MI signals. First of all, based on the absolute Pearson’s matrix (PCC) of overall signals, the graph Laplacian is built up to represent the topological relationship of EEG electrodes. Besides, the GCNs-Net built on the graph convolutional layers learn the generalized features. The followed pooling layers reduce dimensionality. And the fully-connected (FC) softmax layer derives the final prediction. Furthermore, the Chebyshev polynomial is applied to approximate the graph convolutional filters, which significantly promotes computational efficiency. Last but not least, the GCNs-Net decode time-resolved EEG MI signals, which paves the road toward effective and efficient BCI applications. The main contributions of this work are summarized as follows.

1) A novel structure of the GCNs is introduced to detect four-class MI intentions while cooperating with the functional topological relationship of EEG electrodes.
2) The individual and groupwise performance of the GCNs-Net on two benchmark datasets of EEG MI outperform the existing studies, which validates that the method can decode relevant features for brain motor imagination.

3) The introduced GCNs-Net framework can be easily transferred and implemented for other MI-related tasks and potentially for other EEG BCI tasks.

III. MATERIALS AND METHODS

A. Overview

The framework of this work is shown in Fig. 1.

1) 64-channel raw EEG signals are acquired as one of the inputs of the GCNs-Net.

2) The PCC matrix, absolute PCC matrix, adjacency matrix, and graph Laplacian are introduced to represent the correlations between electrodes.

3) The graph representation, another input of the GCNs-Net, is represented by the graph Laplacian.

4) The GCNs-Net is applied to decode EEG MI signals, where N denotes the number of electrodes and l denotes the lth graph pooling layer.

B. Dataset Description

In this work, two benchmark datasets are used to evaluate the effectiveness and robustness of our presented method.

1) PhysioNet Dataset: The EEG Motor Movement/Imagery Dataset consists of over 1500 EEG records from 109 subjects [30]. There are 64 electrodes based on the international 10-10 system. Each subject performs 84 trials (3 runs × 7 trials × 4 tasks), 160-Hz sampling rate and 4-second signals, that is, 640 time points per trail, are utilized considering the duration of the experiments [20]. Four MI tasks are termed L (image left fist), R (image right fist), B (image both fists), and F (image both feet), respectively.

2) High Gamma Dataset: Collected from 14 subjects, the public high gamma dataset performs four EEG tasks, that is, left-hand movement, right-hand movement, both feet movement, and rest [33]. The data of 44 electrodes and 0–125-Hz frequency are applied in our experiments, and the dataset is resampled to 250 Hz [24]. Each subject performs approximately 880 trials for training and 160 trials for testing.

Previous studies apply segments (windows) of time points as samples [34], [48], [49]. However, neuroscience research indicates that the brain is one of the most complicated systems, and its state at every moment is changing. Time-resolved signals can represent the condition of the brain at an instant moment, which reflects the network patterns of brain dynamics. Therefore, in this work, every time point is recognized as a sample to map the brain state to the corresponding MI task. Compared with applying the time-window signals as samples, the method is time-resolved, which is superior to pave the road toward developing real-time and efficient EEG MI applications [40].
To approximate filters [51].

The convolution on graph $G$ is defined as follows:

$$x \ast_G g = U((U^T x) \odot (U^T g))$$

in which $\odot$ denotes the elementwise Hadamard product and $g \in \mathbb{R}^N$ is a convolutional filter. Here, $g$ is nonparametric and it is denoted as $g_\theta(A) = \text{diag}(\theta)$, where $\theta \in \mathbb{R}^N$ is the vector of the Fourier coefficients. The convolutional operation implemented in the GCNs is the following:

$$x \ast_G g_\theta = g_\theta(UAU^T)x = Ug_\theta(A)U^T x.$$  

The difference in the spectral graph convolution lies in the choice of the filter $g_\theta$. Since the nonparametric filter is not localized in space, and its computational complexity is too high, we use the polynomial approximation to address the problem. The Chebyshev polynomials are popularly utilized to approximate filters [51]. $g_\theta$ is parameterized as a truncated
TABLE I
IMPLEMENTATION DETAILS OF THE PROPOSED GCNs-NET ON THE PHYSIONET DATASET

| Layer       | Type         | Maps   | Size       | Edges | Polynomial Order | Pooling Size | Activation     | Weights | Bias |
|-------------|--------------|--------|------------|-------|------------------|--------------|----------------|---------|------|
| Softmax     | Fully-connected | -      | \(\frac{N}{64} \times \frac{N}{64} \times F_6\) | -     | -                | -            | Softmax        | \(\frac{N}{64} \times \frac{N}{64} \times F_6 \times O\) | -      | -    |
| Flatten     | Flatten      | -      | \(\frac{N}{64} \times \frac{N}{64} \times F_6\) | -     | -                | -            | -              | -       | -    |
| P6          | Max-pooling  | \(F_6\) | \(\frac{N}{32}\) | \(\sum_{i=1}^{K} \theta_i T_i(\tilde{A})\) | -       | 2              | -              | -       | -    |
| C6          | Convolution  | \(F_6\) | \(\frac{N}{32}\) | \(\sum_{i=1}^{K} \theta_i T_i(\tilde{A})\) | K       | -              | Softplus       | \(F_5 \times F_6 \times K\) | \(\frac{N}{32} \times F_6\) | -    |
| P5          | Max-pooling  | \(F_5\) | \(\frac{N}{32}\) | \(\sum_{i=1}^{K} \theta_i T_i(\tilde{A})\) | -       | 2              | -              | -       | -    |
| C5          | Convolution  | \(F_5\) | \(\frac{N}{32}\) | \(\sum_{i=1}^{K} \theta_i T_i(\tilde{A})\) | K       | -              | Softplus       | \(F_4 \times F_5 \times K\) | \(\frac{N}{32} \times F_5\) | -    |
| P4          | Max-pooling  | \(F_4\) | \(\frac{N}{32}\) | \(\sum_{i=1}^{K} \theta_i T_i(\tilde{A})\) | -       | 2              | -              | -       | -    |
| C4          | Convolution  | \(F_4\) | \(\frac{N}{32}\) | \(\sum_{i=1}^{K} \theta_i T_i(\tilde{A})\) | K       | -              | Softplus       | \(F_4 \times F_4 \times K\) | \(\frac{N}{32} \times F_4\) | -    |
| P3          | Max-pooling  | \(F_3\) | \(\frac{N}{32}\) | \(\sum_{i=1}^{K} \theta_i T_i(\tilde{A})\) | -       | 2              | -              | -       | -    |
| C3          | Convolution  | \(F_3\) | \(\frac{N}{32}\) | \(\sum_{i=1}^{K} \theta_i T_i(\tilde{A})\) | K       | -              | Softplus       | \(F_3 \times F_3 \times K\) | \(\frac{N}{32} \times F_3\) | -    |
| P2          | Max-pooling  | \(F_2\) | \(\frac{N}{32}\) | \(\sum_{i=1}^{K} \theta_i T_i(\tilde{A})\) | -       | 2              | -              | -       | -    |
| C2          | Convolution  | \(F_2\) | \(\frac{N}{32}\) | \(\sum_{i=1}^{K} \theta_i T_i(\tilde{A})\) | K       | -              | Softplus       | \(F_2 \times F_2 \times K\) | \(\frac{N}{32} \times F_2\) | -    |
| P1          | Max-pooling  | \(F_1\) | \(\frac{N}{32}\) | \(\sum_{i=1}^{K} \theta_i T_i(\tilde{A})\) | -       | 2              | -              | -       | -    |
| C1          | Convolution  | \(F_1\) | \(\frac{N}{32}\) | \(\sum_{i=1}^{K} \theta_i T_i(\tilde{A})\) | K       | -              | Softplus       | \(1 \times F_1 \times K\) | \(\frac{N}{32} \times F_1\) | -    |
| Input       | Input        | 1      | \(\frac{N}{32}\) | \(\sum_{i=1}^{K} \theta_i T_i(\tilde{A})\) | -       | -              | -              | -       | -    |

expansion as follows:

\[
g_\theta(\mathbf{A}) = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{A})
\]

in which \(\theta \in \mathbb{R}^K\) is a set of Chebyshev coefficients, \(T_k(\tilde{A}) \in \mathbb{R}^K\) is the \(k\)th-order Chebyshev polynomial evaluated at \(\tilde{A} = \frac{2A}{\Lambda_{\text{max}}} - I_N\), and \(I_N\) is a diagonal matrix of the scaled eigenvalues.

Then, the signal \(\mathbf{x}\) is convolutioned by the defined filter \(g_\theta\) as follows:

\[
\mathbf{x} \ast G_{\theta} = U \sum_{k=0}^{K-1} \theta_k T_k(\tilde{L}) U^T \mathbf{x} = U \sum_{k=0}^{K-1} \theta_k T_k(\tilde{L}) \mathbf{x}.
\]

3) Graph Coarsening and Fast Pooling: Compared with the pooling operation in CNNs, on graphs, it involves nodes clustering and 1-D pooling. To carry out pooling to reduce dimensionality, the Graclus multilevel clustering algorithm is performed [52]. A greedy algorithm is employed to measure the consecutive coarser of the graph and minimize the objective of the spectral clustering [53]: 1) at each level, multiple numbers of the coarser graphs are given; 2) it picks an unmarked node \(i\) and matches with its unmarked neighborhood \(j\), which needs to maximize the local normalized cut \(W_{ij}(1/d_i + 1/d_j)\). \(W_{ij}\) denotes the edge weight between nodes \(i\) and \(j\), and \(d_i\) and \(d_j\) are the distances between the coarsened node and nodes \(i\) and \(j\), respectively [52], [53]; 3) it will mark the two matched nodes, and the sum of their weights will be the coarsened weight; and 4) all the nodes will undergo the same procedure. Noticeably, at the coarsest level, the nodes will be arbitrarily ordered. Then, the ordered nodes will be propagated to the finest level. Finally, the graph signal is pooled in a 1-D manner [35]. This algorithm cuts the number of nodes by two between two levels.

D. Model Initialization

A novel structure of the GCNs is introduced to classify EEG MI tasks. Based on the absolute Pearson’s matrix of overall signals, the graph Laplacian is built up to represent the topological relationship of EEG electrodes. The graph convolutional layers learn the generalized features. Built on a maximum of \(\log_2 N\) graph pooling layers regarding \(N\) EEG channels, the pooling operation reduces dimensionality, and the FC softmax layer derives the final prediction. With regard to the 64-channel international 10-20 EEG system, the maximum number of pooling layers is six. The implementation details are listed in Table I, where \(N\) denotes the input size of the EEG signals, \(F_i \in [F_1, F_2, F_3, F_4, F_5, F_6]\) donates the number of filters at the \(i\)th graph convolutional layer, \(K\) denotes the polynomial order for filters, and \(O\) is the number of MI tasks.

The hyperparameters in our work (e.g., learning rate, dropout rate, and weight decay rate) are mainly empirically chosen over all the experiments, which are not task-oriented tuned. The network parameters, that is, weights and biases, are updated by the Adam iterative solver [54] with a 0.01 learning rate. Biases are applied to every node of the graph. The batch size is 1024 to maximize the usage of GPU resources. For FC layers in Section IV-A, a 50% dropout rate is applied [55]. The batch normalization (BN) is employed for graph convolutions. The BN normalizes the input graph signals by subtracting the.
Fig. 3. Accuracy of some selected models regarding different polynomial approximation order. The models are selected from Table II. (a) Accuracy of the model C1-P1 (model 1). (b) Accuracy of the model C2-P2 (model 3). (c) Accuracy of the model C3-P3 (model 6). (d) Accuracy of the model C4-P4 (model 10). (e) Accuracy of the model C5-P5 (model 14). (f) Accuracy of the model C6-P3 (model 16). (g) Accuracy of the model C6-P5 (model 19). (h) Accuracy of the model C6-P6 (model 20).

mean and dividing the standard deviation of the mini-batch. Then, it scales and shifts the normalized signals to align with the original distribution. It not only alleviates the problem of internal covariate shift, but also prevents gradient vanishing [56]. The nonlinear smooth rectified linear unit (Softplus) activation function is applied to the graph convolutional layers and FC layers to prevent gradient vanishing [57], where $x$ are the input signals

$$\text{Softplus}(x) = \log(1 + \exp(x)).$$

The softmax function is utilized to derive the final prediction

$$\text{Softmax}(\hat{y}_i) = \frac{\exp(\hat{y}_i)}{\sum_{i=1}^{O} \exp(\hat{y}_i)}$$

where $\hat{y}_i \in [\hat{y}_1, \ldots, \hat{y}_O]$ is the predicted probability of the $i$th MI task. In (7), the cross-entropy loss with the L2 regularization (weight decay) is employed as the loss function. Meanwhile, the weight decay rate $\rho$ is set to $1 \times 10^{-2}$

$$\text{Loss} = -\sum_{i=1}^{O} y_i \log(\hat{y}_i) + \frac{\rho}{2N_P} \|W\|^2$$

where $y_i$, $W$, and $N_P$ are the corresponding MI task, network parameters, and the number of parameters, respectively.

E. Evaluation Metrics

To evaluate performance, multiple metrics are adopted, including accuracy, Cohen’s Kappa coefficient (Kappa) [58], single class accuracy on each task, macro-averaged precision, recall, F1-score, receiver operating characteristic curve (ROC curve), and the area under roc curve (AUC). Besides, to validate whether the performance difference between methods is statistically significant, the pairwise $t$-test is applied. In this work, the significance level of the $t$-test, that is, the $p$-value, is set to 0.05 [59].

IV. RESULTS AND DISCUSSION

A. Novel DL Framework of the GCNs

To explore an optimal model for the GCNs-Net, as detailed in Table II, the decoding performance of multiple structures is investigated by changing some network hyperparameters, such as the number of graph convolutional (Conv) and pooling layers, the polynomial order of the Chebyshev polynomials for filters, and the number of convolutional filters. C denotes a graph convolutional layer, P denotes a graph pooling layer, F denotes an FC layer, and S denotes a softmax layer. The PhysioNet dataset is used to compare the performance of different architectures, as it contains the largest number of participants in the field of EEG MI. The amount of data makes it particularly well suited for training DL models. The dataset of 20 subjects (S1–S20) with 64-channel 1 075 200 samples (640 time points $\times$ 84 trials $\times$ 20 subjects) is utilized to train and evaluate different architectures.

First of all, while holding the number of graph Conv and pooling layers, experiments are carried out by changing the Chebyshev polynomial order from the first to the fifth as described in Table II. Fig. 3(a) displays that when there is only one graph Conv layer followed by a graph pooling layer, the order of the Chebyshev does not make a difference. The accuracy of each order is less than 58%. They overlap with each other and fluctuate during training. Additionally, when the number of graph Conv layers is greater than one, the accuracy of the model with the first-order approximation ascends. In the later epochs, it enters a period of dormancy. The accuracy of models with the first-order witnesses a rugged and abrupt ascent when there are more graph Conv layers. Apart from the number of graph Conv layers, the accuracies of models move upward smoothly with the increasing number of pooling layers.

As illustrated in Fig. 3, the accuracy regarding models with the first polynomial order is unsatisfactory. By contrast, the
TABLE II

| Model | Num. of Core Layers | Num. of Pooling Layers | Num. of Filters | Model Framework | Accuracy w.r.t. 1st order | Accuracy w.r.t. 2nd order | Accuracy w.r.t. 3rd order | Accuracy w.r.t. 4th order | Accuracy w.r.t. 5th order |
|-------|---------------------|------------------------|----------------|----------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| 1     | 1                   | 1                      | 16             | C-P-S           | 55.63%                   | 55.30%                   | 58.70%                   | 56.60%                   | 56.04%                   |
| 2     | 2                   | 2                      | 16, 32         | C-P-S           | 57.86%                   | 61.90%                   | 63.17%                   | 63.16%                   | 63.37%                   |
| 3     | 2                   | 2                      | 32             | (C-P)×2-S       | 60.04%                   | 62.32%                   | 62.55%                   | 62.96%                   | 62.08%                   |
| 4     | 3                   | 1                      | 16, 32, 64     | C-C-P-S         | 58.07%                   | 69.18%                   | 69.66%                   | 71.17%                   | 70.98%                   |
| 5     | 3                   | 2                      | 16, 32, 64     | C-C-P-S×2-S     | 61.65%                   | 69.73%                   | 70.19%                   | 71.06%                   | 71.45%                   |
| 6     | 3                   | 3                      | 16, 32, 64     | (C-P)×3-S       | 65.03%                   | 70.50%                   | 69.12%                   | 70.36%                   | 71.30%                   |
| 7     | 4                   | 2                      | 16, 32, 64, 124| C-C(C-P)×2-S   | 65.27%                   | 77.09%                   | 77.04%                   | 78.42%                   | 78.15%                   |
| 8     | 5                   | 2                      | 16, 32, 64, 124| C-C-P-S×2-S     | 63.96%                   | 77.59%                   | 77.12%                   | 79.28%                   | 77.03%                   |
| 9     | 4                   | 3                      | 16, 32, 64, 124| C-P-P-S×3-S     | 67.50%                   | 77.63%                   | 77.67%                   | 79.60%                   | 78.36%                   |
| 10    | 4                   | 4                      | 16, 32, 64, 124| (C-P)×4-S       | 71.22%                   | 77.61%                   | 78.30%                   | 78.22%                   | 78.20%                   |
| 11    | 5                   | 3                      | 16, 32, 64, 124, 256| C-C-C-P-S×3-S | 70.11%                   | 83.06%                   | 82.78%                   | 84.29%                   | 84.19%                   |
| 12    | 5                   | 4                      | 16, 32, 64, 124, 256| C-P-P-P-S×3-S   | 70.12%                   | 83.05%                   | 82.49%                   | 84.26%                   | 84.34%                   |
| 13    | 5                   | 5                      | 16, 32, 64, 124, 256| C-C-C-P-S×4-S   | 75.95%                   | 84.17%                   | 85.90%                   | 84.74%                   | 84.57%                   |
| 14    | 6                   | 3                      | 16, 32, 64, 124, 256, 512| C-C-C-P-S×3-S | 77.79%                   | 84.39%                   | 84.30%                   | 83.80%                   | 85.08%                   |
| 15    | 6                   | 4                      | 16, 32, 64, 124, 256, 512| C-C-C-P-S×4-S   | 77.69%                   | 87.63%                   | 87.18%                   | 88.18%                   | 87.34%                   |
| 16    | 6                   | 5                      | 16, 32, 64, 124, 256, 512| C-P-P-P-P×2-C-P-S | 78.38%                   | 87.65%                   | 87.70%                   | 87.80%                   | 87.36%                   |
| 17    | 6                   | 6                      | 16, 32, 64, 124, 256, 512| C-P-P-P-P×3-S   | 81.89%                   | 88.60%                   | 88.09%                   | 88.45%                   | 88.91%                   |
| 18    | 6                   | 6                      | 16, 32, 64, 124, 256, 512| C-P-P-P-P×3-S   | 84.88%                   | 88.85%                   | 88.23%                   | 88.37%                   | 87.90%                   |

TABLE III

| Model | Num. of Filters | Num. of FC Layers | Num. of Neurons at FC Layer | Model Framework | Accuracy |
|-------|----------------|------------------|----------------------------|----------------|----------|
| Base  | 16, 32, 64, 128, 256, 512 | 1                | 4                          | (C-P)×6-S        | 88.85%   |
| 1     | 32, 64, 128, 256, 512, 1024 | 1                | 4                          | (C-P)×6-S        | 90.60%   |
| 2     | 64, 128, 256, 512, 1024, 1536 | 1               | 4                          | (C-P)×6-S        | 90.89%   |
| 3     | 16, 32, 64, 128, 256, 512 | 2                | 64, 4                      | (C-P)×6-F×S      | 88.08%   |
| 4     | 16, 32, 64, 128, 256, 512 | 2                | 512, 4                     | (C-P)×6-F×2-S    | 88.64%   |
| 5     | 16, 32, 64, 128, 256, 512 | 3                | 512, 64, 4                 | (C-P)×6-F×3-S    | 88.35%   |
| 6     | 16, 32, 64, 128, 256, 512 | 3                | 512, 256, 4                | (C-P)×6-F×3-S    | 89.45%   |
| 7     | 32, 64, 128, 256, 512, 1024 | 3               | 512, 64, 4                 | (C-P)×6-F×3-S    | 90.45%   |

Fig. 4. Accuracy of different models while applying for the same polynomial order. (a) Accuracy of different models regarding the first-order Chebyshev polynomial. (b) Accuracy of different models regarding the second-order Chebyshev polynomial. (c) Accuracy of different models regarding the fifth-order Chebyshev polynomial. (d) Accuracy of the top ten models.

Fig. 5 shows that when there are more filters at the graph Conv layers, the accuracy ascends marginally. However, as indicated in Fig. 5(b), the loss value rises slightly after a fall. It means that the model with more filters has caused accuracies of models regarding the second to fifth polynomial orders are making a different climb. But they almost overlap and parallel with each other during training. It indicates that when the order of polynomial approximation is greater than one, there is a minor impact on the EEG MI decoding. As a result, the second-order Chebyshev approximation for filters is employed to not only achieve a superior performance but also reduce the model complexity.

Besides, the impacts of performance by changing the number of graph Conv and pooling layers are ablated at a specific polynomial approximation order. Fig. 4 demonstrates that when the number of graph Conv layers increases, the accuracies take a steep climb. Notably, as for the second polynomial order in Fig. 4(b), it illustrates that the number of Conv layers does affect performance. While applying a deeper model, including extra Conv layers, features can be better extracted from EEG signals. Meanwhile, the effects of the number of graph pooling layers are also investigated. The number of pooling layers promotes and enhances the decoding performance, but with a modest increment. As detailed in Fig. 4(d), when the polynomial order is second, the accuracies are 88.60% (Model C6-P5-K2) and 88.85% (Model C6-P6-K2), respectively.

Furthermore, based on the optimal C6-P6-K2 model that contains six graph Conv layers with the second polynomial order for filters and six graph pooling layers, the influence on performances by changing the number of convolutional filters at every Conv layer and the number of FC layers is also explored in Table III.
overfitting. The reason is that the model structure becomes much more complicated while applying more filters. Consequently, regarding the PhysioNet dataset, 16, 32, 64, 128, 256, and 512 filters are used for the six-layer GCNs-Net to prevent overfitting. Meanwhile, there is a gentle descent of performances while adding more FC layers. As a result, a softmax layer is directly implemented without applying extra FC layers.

**B. Subject-Level Validation**

The GCNs-Net is validated on ten subjects from the PhysioNet dataset, each with 64-channel 53,760 samples (640 time points × 84 trials × 1 subject). The decoding accuracies are listed as follows: S1 (97.08%), S2 (90.70%), S3 (97.92%), S4 (96.86%), S5 (80.49%), S6 (89.55%), S7 (84.82%), S8 (97.40%), S9 (97.02%), and S10 (98.72%).

According to Fig. 6, the PCC matrix, absolute PCC matrix, adjacency matrix, and graph Laplacian for Subjects 10 and 5 are shown, which achieve the highest and the lowest accuracy, respectively. There are quite a lot of variations underlying the intersubject EEG signals. For each one of the ten subjects, 98.72% maximum accuracy is achieved. As for the model of Subject 10, the AUC is 0.99. The single-class accuracies on the L, R, B, and F are 99.92%, 97.96%, 98.08%, and 98.93%, respectively. For ten subjects, the highest F1-score is 98.71%, and the lowest is 80.19%.

Meanwhile, the presented GCNs-Net is validated on the high gamma dataset. The model containing 14 subjects is separately trained and evaluated. Since there are 44 electrodes [24], the maximum number of pooling layers is 2. We use the (C-C-C-P) × 2 architecture of the GCNs-Net to decode EEG tasks. From Subjects 1 to 14, 96.43%, 95.63%, 93.04%, 99.18%, 98.65%, 94.77%, 93.49%, 97.91%, 95.48%, 96.77%, 98.55%, 98.69%, 98.34%, and 90.43% accuracies are achieved, respectively. The mean accuracy is 96.24%. The results indicate that the GCNs-Net manages to handle the individual variability due to its robustness and effectiveness.

**C. Classification at the Group Level**

Next, the GCNs-Net is evaluated at a group of 20 subjects (S1–S20) from the PhysioNet dataset. The accuracy, Kappa, precision, recall, and F1-score are 88.35%, 84.47%, 88.39%, 88.35%, and 88.34%, respectively. Furthermore, the single-class accuracies on the L, R, B, and F are 83.45%, 86.72%, 83.96%, and 99.42%, separately. The method performs well in classes F and B on the PhysioNet dataset. The AUC is 0.92. Besides, it is evaluated on the high gamma dataset. The data of 14 subjects are used in the experiment. 80.89% accuracy and 80.78% F1-score are achieved. The reason for the accomplishment is that the GCNs-Net converges for the groupwise predictions and succeeds in extracting relevant features from EEG signals.

**D. Tenfold Cross-Validation for Reliability**

The GCNs-Net is trained at the group level of 20 subjects (S1–S20) from the PhysioNet dataset, following the tenfold cross-validation to validate the stability and reliability. We divide the dataset into ten pieces, use one of them as the testing set, and the left nine pieces as the training set in turn.

With the results from Section IV-A (88.85% accuracy, Model C6-P6-K2), 11 results are listed in Fig. 7. 89.39% maximum accuracy is achieved, and the lowest is 87.90%. The averaged accuracy and F1-score are both 88.57%. At the group level, the performance is stably reproducible through repetitive experiments for cross-validation, showing the reliability and stability of the GCNs-Net.

**E. Robustness to Data Size**

It has also been trained and evaluated on different amounts of participants on the PhysioNet Dataset. The dataset of 50 subjects (S1–S50) with 64-channel 2688,000 samples (640 time points × 84 trials × 50 subjects) and the dataset of 100 subjects (S1–S100) with 64-channel 5,376,000 samples (640 time points × 84 trials × 100 subjects) are used. Accuracies and losses are illustrated in Fig. 8.

The GCNs-Net has achieved 89.75% (testing set) and 94.99% (training set) accuracies for a group of 50 subjects. Furthermore, it also achieves 88.14% (testing set) and 93.24% (training set) accuracies for 100 subjects. The results have shown that it learns the generalized features from subjects at a larger scale. The reason for this phenomenon is that the GCNs-Net handles a larger amount of subjects, suggesting the adaptability and robustness to individual variability. The presented method is much more effective and robust in processing the graph-structured EEG-based MI signals since it has considered the functional topological relationship of EEG electrodes.

**F. Comparison With State-of-the-Art**

Two-level experiments are applied to compare the performance of the current models, that is, either from the subject level or the group level. The decoding performance is mainly measured by the maximum accuracy (Max. Accuracy) and the averaged accuracy (Avg. Accuracy) on two datasets. First of all, the performance on the PhysioNet dataset is compared in Table IV.
TABLE IV
PERFORMANCE COMPARISONS ON THE PHYSIO NET DATASET

| Related Work | Max. Accuracy | Avg. Accuracy | p-value | Level | Approach | Num. of Subjects |
|--------------|---------------|---------------|---------|-------|----------|------------------|
| Dose et al. (2018) [22] | 80.38% | 68.51% | < 0.05 | Group | CNNs | 105 |
| Ma et al. (2018) [60] | 82.65% | 68.20% | — | Group | RNNs | 12 |
| Hou et al. (2020) [20] | 94.50% | 96.00% | > 0.05 | Subject | ESI-CNNs | 1 |
| Hou et al. (2022) [34] | 98.81% | 95.48% | > 0.05 | Subject | BiLSTM-GCN | 20 |
| Jia et al. (2022) [40] | 94.16% | 93.78% | > 0.05 | Group | Graph ResNet | 1 |
| **Author** | **89.39%** | **88.79%** | — | Group | GCNs-Net | **100** |
| **88.14%** | **98.72%** | **93.06%** | — | Subject | | 1 |

Fig. 7. Accuracy and F1-score of the repetitive experiments for 20 subjects from the PhysioNet dataset. (a) Accuracy of the repetitive experiments for 20 subjects. (b) F1-score of the repetitive experiments for 20 subjects.

The ESI-CNN approach has achieved 94.50% maximum accuracy at the group level (ten subjects) [20]. Lately, the graph learning-based methods, for example, BiLSTM-GCN and Graph ResNet, achieve highly competitive performances. This phenomenon is due to the superiority of graph representation learning for EEG signal processing [34], [40]. The GCNs-Net also obtains competing performances, 89.39% maximum accuracy for a group of 20 participants, and 88.14% for 100 participants. Meanwhile, at the subject level, the p-value between the GCNs-Net and the CNN models [22] is significantly less than 0.05. It indicates a significant difference in the predictive performance between the two models, and the GCNs-Net is superior to predicting EEG tasks, with a 30.21% maximum accuracy increment. However, compared with the ESI-CNN approach [20], there is no significant difference in classification performance at a 95% confidence interval as the p-value is greater than 0.05. Furthermore, the GCNs-Net attains the best state-of-the-art performance at the hundred-subject level on the PhysioNet dataset, which far exceeds current studies. The reason for the outcome is that the presented approach maintains a robust and effective...
dataset with a larger amount of participants, regardless of the inter-trial and intersubject variations.

In Table V, we compare the classification performance of some representative works [24], [29], [61] on the high gamma dataset. Evaluated on the high gamma dataset, the p-values are both less than 0.05 while comparing the introduced method with the CNNs-based methods [24], [61]. The performance is significantly different among the models. The GCNs-Net successfully predicts MI tasks with dominant performances, that is, 99.18% maximum accuracy and 96.24% averaged accuracy. The p-value compared with the DAN approach [29] is greater than 0.05. The performance of the two models is statistically less different, and both models achieve competing performances. Last but not least, the dominant classification accuracy has verified the robustness and effectiveness of our presented GCNs-Net.

### V. CONCLUSION

In order to deeply extract network patterns of brain dynamics, the GCNs-Net, a novel DL framework based on the GCNs, is presented to distinguish four-class MI intentions by cooperating with the functional topological relationship of EEG electrodes. The introduced method has been proved to converge for both personalized and groupwise predictions. Trained with individual data, the approach has achieved an averaged accuracy of 93.06% (PhysioNet dataset) and 96.24% (high gamma dataset) in predicting the independent trials of the same participant, which is dominant in existing studies, indicating that the GCNs-Net converges well for individuals. Moreover, it has reached the uppermost accuracy on numerous sizes of group-level prediction on the PhysioNet dataset, that is, with 89.39% accuracy for 20 subjects, 89.75% for 50 subjects, and 88.14% for 100 subjects, which implies that it is considerably robust to individual variability. Furthermore, it holds an average accuracy of 88.57% after tenfold cross-validation showing reliability and stability. On the other hand, it predicts all four MI tasks with superior accuracy, the best among which is the two feet prediction with an accuracy of 99.42%. It indicates that the introduced method is able to build a generalized representation against both personalized and groupwise variations. In conclusion, we have developed a novel GCNs-Net method for EEG data classification. The outstanding performance of our method in MI identification is an important step toward better BCI approaches and neuroscience research.

### ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their constructive comments to substantially improve this article.
[21] S. U. Amin, M. Alsulaiman, G. Muhammad, M. A. Mekhtiche, and M. S. Hossain, “Deep learning for EEG motor imagery classification based on multi-layer CNNs feature fusion,” Future Gener. Comput. Syst., vol. 101, pp. 542–554, Dec. 2019.

[22] H. Dose, J. S. Møller, H. I. Versen, 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, Dec. 2018.

[23] S. Tan, W. Zhang, T. Tan, Y. Bajai, and A. Sengur, “Convolutional neural network based approach towards motor imagery tasks EEG signal classification,” IEEE Sensors J., vol. 19, no. 12, pp. 4494–4500, Jun. 2019.

[24] R. T. Schirrmeister et al., “Deep learning with convolutional neural networks for EEG decoding and visualization,” Hum. Brain Mapping, vol. 38, no. 11, pp. 5391–5420, Aug. 2017.

[25] C. J. Ortiz-Echeverri, S. Salazar-Colores, J. Rodríguez-Reséndiz, and R. A. Quintana-Zoel, “A new approach for motor imagery classification based on sorted blind source separation, continuous wavelet transform, and convolutional neural network,” Sensors, vol. 19, no. 20, pp. 4541–4565, Oct. 2019.

[26] D. Li, J. Wang, J. Xu, and X. Fang, “Densely feature fusion based on convolutional neural networks for motor imagery EEG classification,” IEEE Access, vol. 7, pp. 32720–32730, 2019.

[27] R. Zhang, Q. Zong, L. Dou, and X. Zhan, “A novel hybrid deep learning scheme for four-class motor imagery classification,” J. Neural Eng., vol. 16, no. 6, Oct. 2019, Art. no. 066004.

[28] R. Alazzari, M. Abuhijleh, H. Alwanni, and M. I. Daoud, “A deep learning framework for decoding motor imagery tasks of the same hand using EEG signals,” IEEE Access, vol. 7, pp. 109612–109627, 2019.

[29] X. Tang and X. Zhang, “Conditional adversarial domain adaptation neural network for motor imagery EEG decoding,” Entropy, vol. 22, no. 1, p. 96, Jan. 2020.

[30] A. L. Goldberger et al., “PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals,” Circulation, vol. 101, no. 23, pp. e215–e220, Jun. 2000.

[31] D. Zhang, L. Yao, K. Chen, S. Wang, X. Chang, and Y. Liu, “Making sense of spatio-temporal preserving representations for EEG-based human intention recognition,” IEEE Trans. Cybern., vol. 50, no. 7, pp. 3033–3044, Jul. 2020.

[32] C. Brunner, R. Leeb, G. Müller-Putz, A. Schügol, and P. Pfurtscheller, “BCI competition 2008-Graz data set A,” Inst. Knowl. Discovery, Lab. Brain Comput. Interface, Graz Univ. Technol., Graz, Austria, Tech. Rep., 2008, pp. 1–6. [Online]. Available: http://www.bbci.de/competition/iv/desc_2a.pdf.

[33] M. Tangermann et al., “Review of the BCI competition IV,” Frontiers Neurosci., vol. 6, p. 55, Jul. 2012.

[34] Y. Hou et al., “Deep feature mining via the attention-based bidirectional long short term memory graph convolutional neural network for human motor imagery recognition,” Frontiers Bioeng. Biotechnol., vol. 9, p. 1525, Feb. 2022.

[35] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in Proc. Adv. Neural Inf. Process. Syst., vol. 29, Red Hook, NY, USA: Curran Associates, Dec. 2016, pp. 3844–3852.

[36] T. Zhang, X. Wang, X. Xu, and C. L. P. Chen, “GCN-Net: Graph convolutional broad network and its application in emotion recognition,” IEEE Trans. Affect. Comput., vol. 13, no. 1, pp. 379–388, Jan. 2022.

[37] Z. Fang, T. Tang, and X. Heng, “Phase-locking value based graph convolutional neural networks for emotion recognition,” IEEE Access, vol. 8, pp. 93711–93722, May 2020.

[38] S. Jia, Y. Hou, Y. Shi, and Y. Li, “Attention-based graph ResNet for motor intent detection from raw EEG signals,” 2020, arXiv:2007.13484.

[39] W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in Proc. Adv. Neural. Inf. Process. Syst., vol. 30, Red Hook, NY, USA: Curran Associates, Dec. 2017, pp. 1024–1034.

[40] 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., Jul. 2017, pp. 5115–5124.

Yimin Hou received the Ph.D. degree from North-western Polytechnical University, Xi’an, China, in 2007. He is currently a Professor and the Vice Dean of the School of Automation Engineering, North-east Electric Power University, Jilin, China. He has authored over 30 articles in international journals and international conferences in the fields of image processing, computer vision, and EEG analysis. His current research interests include signal processing, BCI design, and DL.
Shuyue Jia (Student Member, IEEE) received the B.Eng. degree from Northeast Electric Power University, Jilin, China, in 2020. He is currently pursuing the M.Phil. degree in computer science with the City University of Hong Kong, Hong Kong.

In 2020 and 2022, he interned with Philips Research, Shanghai, China, and Samsung Research, Beijing, China. In the summer of 2017, he was a Visiting Student with the University of California at Irvine, Irvine, CA, USA. His current research interests include image and video processing and computer vision.

Xiangmin Lun received the Ph.D. degree from the Changchun University of Science and Technology, Changchun, China, in 2022.

She is currently a Lecturer with the School of Automation Engineering, Northeast Electric Power University, Jilin, China. She has authored six articles in international journals and international conferences in the fields of EEG analysis, motor imagination, and BCI. Her current research interests include BCI design and DL.

Ziqian Hao received the M.E. degree from the Aeronautical Engineering Institute, Yantai, China, in 2007.

She is working as a Lecturer with the School of Finance and Business, Jinan Vocational College, Jinan, China. She is dedicated to both research and teaching. Her research fields include digital image and signal processing, control science, and data science.

Yan Shi (Member, IEEE) was born in Jilin, China. He received the B.E., M.E., and Ph.D. degrees from the School of Automation Engineering, Northeast Electric Power University, Jilin, in 2015, 2018, and 2022, respectively.

He is currently a Lecturer in automation engineering with Northeast Electric Power University. He has published 15 articles in the fields of electrical engineering, instrument science, and pattern recognition. His research interests include sensor fabrication, feature mining, quality inspection, and DL.

Yang Li (Senior Member, IEEE) was born in Nanyang, China. He received the Ph.D. degree in electrical engineering from North China Electric Power University (NCEPU), Beijing, China, in 2014.

From January 2017 to February 2019, he was a postdoc with the Argonne National Laboratory, Lemont, IL, USA. He is currently a Professor with the School of Electrical Engineering, Northeast Electric Power University, Jilin, China. His research interests include renewable energy integration, AI-driven power system stability analysis, and microgrids.

Dr. Li serves as an Associate Editor for the IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS, IET Renewable Power Generation, and Energy Reports.

Rui Zeng received the B.S. degree in computer science from the Wuhan Institute of Technology, Wuhan, China, in 2012, the M.S. degree in electronic telecommunications from the University of Rennes, Rennes, France, in 2015, the M.Eng. degree in computer science from Southeast University, Nanjing, China, in 2015, and the Ph.D. degree from the Queensland University of Technology, Brisbane, QLD, Australia, in 2020.

He was a Research Fellow with the Australian Centre for Robotic Vision and eResearch Centre, Monash University, Melbourne, VIC, Australia, and a Post-Doctoral Research Associate with the Brain and Mind Centre, University of Sydney, Sydney, NSW, Australia. His research interests include multiview geometry, DL, and medical imaging, and diffusion MRI.

Jinglei Lv received the Ph.D. degree in engineering from a joint Ph.D. Program between the Northwestern Polytechnical University, Xi’an, China, and the University of Georgia, Athens, GA, USA, in 2016.

He worked as a Research Fellow with the Melbourne Neuropsychiatry Centre, The University of Melbourne and the Translational Neuroscience Group, QIMR Berghofer, Brisbane, QLD, Australia. He has studied and worked as a Researcher with the Computer Science Department, University of Georgia. He is currently a Senior Lecturer dedicated to both research and teaching with the School of Biomedical Engineering and the Brain and Mind Centre, University of Sydney, Sydney, NSW, Australia. He has authored or coauthored more than 100 peer-reviewed articles. His research interest includes neuroimaging analysis, neural signal analysis, machine learning, neuroscience, and mental health.