BIOCYBERNETICS AND BIOMEDICAL ENGINEERING 42 (2022) 1023—1040

Available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/bbe

Original Research Article

EEG_GENet: A feature-level graph embedding method for motor imagery classification based on EEG signals

Huiyang Wang [a,b], *Hua Yu* [c], *Haixian Wang* [a,b,*]

[a] *Key Laboratory of Child Development and Learning Science of Ministry of Education, School of Biological Science & Medical Engineering, Southeast University, Nanjing 210096, Jiangsu, PR China*

[b] *Institute of Artificial Intelligence of Hefei Comprehensive National Science Center, Hefei 230094, Anhui, PR China*

[c] *Department of Cardiology, The First Affiliated Hospital of USTC, Division of Life Sciences and Medicine, University of Science and Technology of China, Hefei 230001, Anhui, PR China*

ARTICLE INFO

Article history:
Received 12 January 2022
Received in revised form
15 August 2022
Accepted 19 August 2022
Available online 29 August 2022

Keywords:
BCIs
EEG
Motor imagery
Deep learning
Graph embedding

ABSTRACT

In recent years, the success of deep learning has driven the development of motor imagery brain-computer interfaces (MI-BCIs) based on electroencephalography (EEG). However, unlike image or language data, motor imagery EEG signals are of multielectrodes with topology information. As a means of integrating graph topology information into feature maps, few studies studied motor imagery classification involving graph embeddings. To decode EEG signals more accurately, this paper proposes a feature-level graph embedding method and combines the method with EEGNet; this new network is called EEG_GENet. Specifically, time-domain features are obtained by convoluting raw EEG signals for each electrode. Then, the adjacent matrix, conceptualized as a graph filter, performs graph convolution and uses the time-domain features to embed the topology information. This process can also perform multi-order graph embeddings. In addition, the adjacency matrix in this paper can adapt to different brain network connectivities for different subjects. We evaluate the proposed method on two benchmark EEG datasets for motor imagery classification. Experimental results on the BCICIV-2a and High_Gamma datasets demonstrate that EEG_GENet achieves 79.57% and 96.02% classification accuracy, respectively. These results indicate that the proposed method is superior to state-of-the-art methods. In addition, various ablation experiments further verify the advantages of the feature-level graph embedding method. To conclude, the feature-level graph embedding method can improves the network's ability to decode raw motor imagery EEG signals.

* Corresponding author at: Key Laboratory of Child Development and Learning Science of Ministry of Education, School of Biological Science & Medical Engineering, Southeast University, Nanjing 210096, Jiangsu, PR China.
E-mail address: hxwang@seu.edu.cn (H. Wang).

https://doi.org/10.1016/j.bbe.2022.08.003

1. Introduction

Electroencephalography (EEG) signals are bioelectric information that reflects the physiological activities of human brain. They are widely applied to brain-computer interfaces (BCIs), which translate observed brain activities into meaningful information to communicate between brain and external environments [1,2]. There are two kinds of BCIs based on EEG signals: exogenous stimulation presentation, such as steady-state visual evoked potential (SSVEP) [3,4], and endogenous modulation, such as motor imagery [5,6].

Motor imagery BCIs based on EEG have gained popularity in clinical applications due to their advantages of low clinical risk, economy, convenience, and no requiring stimulus targets. For example, motor imagery BCIs have been applied to control intelligent devices such as electric wheelchairs [7], hand exoskeletons [8], artificial prosthetic limbs [9], and robot movement [10]. It is also developed in the motor recovery field for stroke patients [11,12] and Parkinson's disease patients [13]. However, motor imagery EEG signals have a low signal-to-noise ratio (SNR) and highly dynamic characteristics, which limit the discrimination ability and generalization performance of the motor imagery BCIs. The EEG decoding methods based on traditional machine learning and deep learning have emerged to solve these problems.

The methods based on traditional machine learning involve cumbersome data preprocessing, hand-crafted feature extraction, and classification. During the data preprocessing, primary tasks include removing artifacts [14] and bandpass filtering [15]. Many studies [16] have used the power spectral density (PSD) as the primary feature in the feature extraction stage. Neurophsiologically, during motor imagery and execution, an evident power spectrum change is established in the brain's sensory-motor rhythm [17], called event-related synchronization/desynchronization (ERS/ERD). The autoregressive (AR) model can be used to analyze the time-domain features of EEG signals [18], but this method is limited due to its non-stationarity characteristics. Aggarwal and Chugh [19] argue that common spatial pattern (CSP) is the most preferred feature extraction method by comprehensively comparing prominent feature extraction techniques. The algorithm has been used to construct a set of optimal spatial filters to project multielectrode EEG data into a low-dimensional spatial subspace to maximize the difference between the variance values of the two classes of EEG signals, and thereby, features with higher discrimination can be obtained. There are many variants of CSP, such as multi-time and multi-band CSP (MTF-CSP) [20], regularized CSP (RCSP) [21], filter bank regularized CSP (FBRCSP) [22] and correlation-based regularized CSP (corr-RCSP) [23], which all further improve classification performance. Granger causality is beneficial for expressing the dynamic features of brain networks and is widely used as a feature extracting method [24]. In addition, using pupil diameter as an additional feature can improve motor imagery classification results [25]. In the classification stage, the support vector machine (SVM) [26,27] and decision tree [28] are two of the most popular algorithms and are widely used as classifiers in motor imagery.

Notice that feature extraction and classifier optimization are separated, preventing the potential in which the two stages may promote each other. Additionally, hand-crafted features may ignore some underlying information from the EEG signals [29]. Unlike traditional machine learning, deep learning is an end-to-end method. Motor imagery BCIs based on deep learning can perform various dexterous movements. For instance, Alazrai et al. [30] designed a framework that can carry out 11 different motor imagery tasks of the same hand. Raw EEG signals require only simple preprocessing or transformation to spectrogram images [31–33], and networks can automatically mine data features and make decisions. Because the feature extraction modules and the classifiers are optimized in tandem, their classification performance is generally better than traditional machine learning methods. Deep learning has achieved massive success in computer vision (CV) and natural language processing (NLP), inspiring EEG signals classification methods based on deep learning. Schirrmeister et al. [34] designed a feature extraction module with a residual structure that obtained good results comparable with filter bank common spatial pattern (FBCSP) [35]. Lawhern et al. [36] proposed a novel network called EEGNet based on depthwise separable convolution with approximately 2000 trainable parameters, creating a balance between network fitting capability and generalization in the case of limited data. Tang and Zhang [37] refer to the densely connected structure [38] in the network design to improve the transmission of EEG features. Qiao and Bi [39] combined the gate recurrent unit (GRU) and the inception structure to propose the IncepCNN-BGRU for motor imagery classification. Their experiments demonstrated that mixing the two structures can significantly improve the network's classification accuracy. Izzuddin et al. [40] proposed Sinc-EEGNet based on the adaptation of SincNet, which was initially developed for speaker recognition tasks to enable network interpretability of decoding EEG signals. Above networks mainly focus on 2D convolution neural networks (CNN), Mattioli et al. [41] proposed a new network consisting entirely of 1D-CNN for motor imagery classification, outperforming other networks based on 2D-CNN.

However, many existing methods based on deep learning do not make good use of the multielectrode characteristics of EEG signals. Taking motor imagery tasks as an example, there may be more than 20 EEG signal electrodes; however, neuroscience research has emphasized brain network dynamics [42]. Therefore, addressing the topological relationship between electrodes is a challenge. The normal convolution process is the weighted sum of each electrode signal. From the graph theory perspective, the normal convolution process only considers nodes but ignores edges, which is not conducive to decoding multielectrode EEG signals. Graph embedding is an excellent way to solve the problem, but few studies regarding motor imagery BCIs involve graph embedding. This paper proposes a feature-level graph embedding method and combines the method with the existing EEGNet; the new network is called EEG_GENet (The

implementation code is publicly available[1]). First, based on the spatial position of the EEG electrodes or Pearson's correlation matrix, an adjacency matrix is built as a graph filter. Then, time-domain features are obtained by convoluting each electrode's raw motor imagery EEG signals.Furthermore, the normalized graph filter and the time-domain features perform matrix multiplication to embed the topology information into the features. Finally, the features continue to propagate forward through the remaining network layers. The primary contributions of this paper are summarized as follows:

1. This paper proposes feature-level graph embedding EEG signals for motor imagery classification tasks. Instead of carefully designing a specialized network, we directly introduce this method to EEGNet, which is widely applied to many studies. We call the combined networks EEG_GENet. We also apply the proposed method to other popular networks to verify our method's effectiveness. Our method can be used as an addable component of networks to improve their classification performance.
2. We conduct comprehensive experiments on two benchmark motor imagery datasets with subject-dependent testing. The experimental results show that our method outperformed the existing methods, demonstrating the advanced performance of the feature-level graph embedding.
3. This paper carefully explores the influence of the adjacency matrix. Various ablation experiments show that different adjacency matrix initialization methods and their trainability significantly influence the network performance. The results also show that the graph embedding with less order is more conducive for decoding motor image EEG signals with EEG_GENet.

2. Related works

Methods for embedding graphs in the network are a popular research topic. Graph embedding through graph neural networks (GNNs) is widely used.

In a GNN, the information in nodes is updated depending on adjacent nodes and the constraints imposed by connected edges. These relationships indicate that the labels of unknown nodes can be predicted by analyzing the data in neighboring nodes. However, the layers share weights, which results in poor classification performance. Until the rise of deep learning, Bruna et al. [43] combined CNNs and spectral graph theory to lay the theoretical foundation and structural framework of today's GNN. Generally, the graph convolution between two graph signals is converted to a corresponding graph filtering operation, and the Laplacian matrix is used as a graph filter. To make the graph filter trainable like a CNN, the graph filter is parameterized, i.e., the eigenvalues of the Laplacian matrix can be learned. However, this method is limited to graph structures with too many nodes. Defferrard et al. [44] adopted a polynomial learning method to reduce the trainable parameters. Only polynomial coefficients are trainable, and a recursive formulation of the k-order Chebyshev polynomials is further used to approximate the filters, which can significantly improve the operation efficiency. Kipf and Welling [45] further limited the order of the graph filter to one since the adequate information in graph data is primarily contained in the low-order parts [46], and the computational complexity is significantly reduced.

Lun et al. [47] proposed GCNs_Net to decode motor imagery EEG signals in the motor imagery field. Except for the last layer, which is a fully connected layer, all the other layers are GNNs. Therefore, graph embedding is carried out in the entire network. However, GCNs_Net only considers the information in the spatial domain while discarding the information in the time domain so that each electrode's input data is a single dimension. Zhang et al. [48] utilized the spatial positioning of EEG electrodes to build up the adjacency matrix and perform data-level graph embedding by calculating the matrix product of the adjacency matrix and raw motor imagery EEG signals. Notably, the data-level graph embedding may enhance noise, and discarding raw EEG signals may sacrifice the network's feature extraction ability. Li et al. [49] calculated mutual information between channels to get an adjacency matrix, and the inputs of networks are hand-crafted features that may ignore some underlying information from EEG signals.

We use GNN for graph embedding in this paper, but the specific embedding methods differ from the above studies. The particular method used in this study is introduced in Section 3.

3. Materials and methods

3.1. Definitions and notations

An EEG dataset for a specific subject can be defined as$S = \{(x_i, y_i), i = 0, 1, 2, \cdots, M-1\}$, where $x_i \in R^{C\times T}$ represents a raw sample from the $i-th$ trial and has T discretized time points and C electrodes, M is the total number of trials, and $y_i \in \{0, 1, \cdots N-1\}$ is the corresponding label of x_i where N is the total number of classes. For a motor imagery task, it is ideal to find a perfect mapping that automatically assigns the correct label to x_i. The decoder model can be mathematically formalized as:

$$\tilde{y} = f(x_i; \theta) \tag{1}$$

where $\tilde{y}$ is the predicted label corresponding to x_i and θ represents all of the trainable parameters in the decoder model. If the decoder model is based on a neural network, formula (1) can be rewritten as:

$$h^l = \sigma\left(h^{l-1} * W^l + b^l\right) \tag{2}$$

$$\tilde{y} = softmax\left(h^L \times W^* + b^*\right) \tag{3}$$

where h^l represents the output from the $l-th$ convolution layer whose weights and bias are defined by W^l andb^l, respec-

[1] code: https://github.com/stickOverCarrot/EEGGENET.

tively; $h^0 = x$ and σ include other common operations such as batch normalization, linear or nonlinear activation, and pooling, among others; L is the total number of convolution layers; W^* and b^* are the weights and bias of the fully connected layer, respectively; $*$ represents the convolution operation; $\times$ represents matrix multiplication. Therefore, $\theta = \left\{W^l, W^*, b^l, b^*, l = 1, 2, \cdots L\right\}$. The softmax function activates the final output, a vector with N dimensions representing the probability distribution for N classes. In the supervised classification task for motor imagery EEG signals decoding, we can minimize the cross-entropy loss L, as shown in formula (4), to optimize the decoder model.

$$L = \frac{1}{B}\sum_{i=0}^{B-1}\sum_{j=0}^{N-1} -\delta(j == y_i)\log \tilde{y}_j + \lambda|\theta|^2 \tag{4}$$

where B is the batch size, δ represents the signal function, $|\cdot|^2$ represents the regularization component, and λ is the trade-off regularization weight.

3.2. *Graph filtering*

Given an undirected and weighted graph $G = (V, E)$, V denotes a set of nodes and E represents a set of edge weights between the nodes in V. According to E, an adjacency matrix $A \in R^{C\times C}$ can be built where C is the total number of nodes, A_{ij} means the edge weights between the $i-th$ and $j-th$ nodes, and $A_{ii} = 0$. Furthermore, the degree matrix D can be obtained, i.e. $D_{ii} = \sum_j |A_{ij}|$, when $i \neq j, D_{ii} = 0$. Thus, the normalized adjacency matrix $\tilde{A}$ can be obtained:

$$\tilde{A} = D_{ii}^{-1/2} \times A \times D_{ii}^{-1/2} \tag{5}$$

Then, the graph filter H is defined by the normalized adjacency matrix polynomials:

$$H = h_0\tilde{A}^0 + h_1\tilde{A}^1 + \cdots + h_k\tilde{A}^k \tag{6}$$

where $h_0, h_1, \cdots, h_k$ are the polynomial coefficients and k is the order of the polynomials, i.e., the order of the graph filter. Some references [43–45] replace the adjacency matrix with the Laplacian matrix $L = I - \tilde{A}$, where I is the identity matrix that constructs the graph filter in formula (6). More importantly, L is a real symmetric matrix in which the k-order Chebyshev polynomials are often used to approximate H to reduce computational complexity. Finally, for node data $x \in R^{C\times T}$, graph filtering is defined by formula (7):

$$\tilde{x} = H \times x \tag{7}$$

3.3. *EEG_GENet*

It is difficult for CNNs to take advantage of the topology information in EEG signals, and representing topology information is vital to decoding these signals. We propose a feature-level graph embedding method to solve this problem. Instead of carefully designing a specialized network, we directly introduce this method to EEGNet, which is widely applied in many studies. Fig. 1 shows the proposed overall framework, and Table 1 shows the EEG_GENet implementation details.

We use the normalized adjacency matrix polynomials in the graph embedding stage as the graph filter. On the one hand, the number of EEG electrodes is generally not very large, and the order k set in this paper is relatively small to avoid significant computational overhead. On the other hand, there may be a negative correlation between the EEG electrodes. In our experiments, node self-loops are contained in A, i.e., $A_{ii} = 1$ to include the signals of the electrode itself. The method chosen to initialize the adjacency matrix A is essential for training EEG_GENet. This paper adopts two methods for initializing the adjacency matrix:

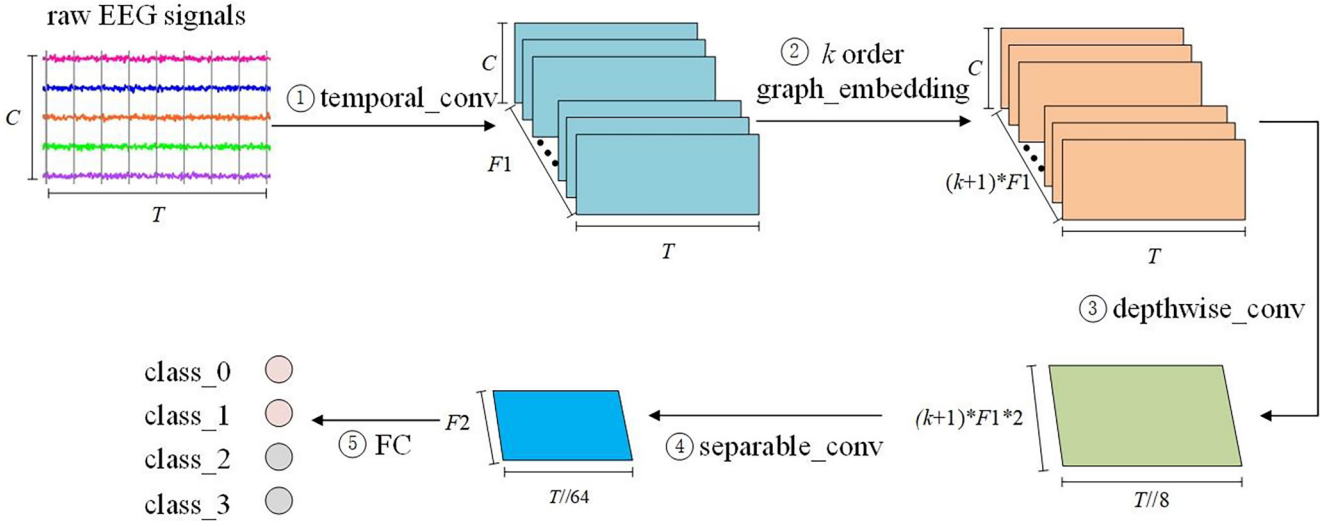


Fig. 1 – The overall EEG_GENet framework. C is the number of EEG electrodes, T is the number of discretized time points, the raw EEG signal channel is 1, F1 and F2 are the feature channel numbers, and k is an order of the graph filter. temporal_conv, depthwise_conv and separable_conv represent convolution layers with different implementations, and FC represents a fully connected layer.

Table 1 – The EEG_GENet implementation details.

Layer	Type	Filter number	Kernel size/Stride/Groups/Max_norm	Output size
Input	None	None	None	$1 \times C \times T$
Temporal_conv	Conv2d	F1	$(1,64)/(1,1)/$ 1/None	$F1 \times C \times T$
	Batchnorm	F1	None	
Graph_embedding	None	$F1 * (k+1)$	None	$F1 * (k+1) \times C \times T$
Depthwise_conv	Conv2d	$F1 * 2 * (k+1)$	$(C,1)/(1,1)/F1 * (k+1)/1.0$	$F1 * 2 * (k+1) \times 1 \times T$
	Batchnorm	F1	None	
	Elu	None	None	
	Average_pool	None	$(1,8)/(1,8)/1$/None	$F1 * 2 * (k+1) \times 1 \times T//8$
	Dropout	None	None	
Separable_conv	Conv2d	$F1 * 2 * (k+1)$	$(1,16)/(1,1)/F1 * 2 * (k+1)/$ None	
	Conv2d	F2	$(1,1)/(1,1)/1/$ None	$F2 \times 1 \times T//8$
	Batchnorm	F2	None	
	Elu	None	None	
	Average_pool	None	$(1,8)/(1,8)/1$/None	$F2 \times 1 \times T//64$
FC	None	4	$(1,T//64)/(1,1)/1/0.25$	$4 \times 1 \times 1$

F1 is the number of temporal filters, F2 is the number of spatial filter, and k is the order of the graph filter.

1). Method based on the spatial position of EEG electrodes (SPEE). The connection between brain regions' strength decreases as the physical distance increases. Therefore, for convenience, the edge between two naturally neighboring EEG nodes is set to 1, while the edge that is not between two naturally neighboring EEG nodes is set to 0. Fig. 2 shows the spatial position of EEG electrodes from the BCICIV-2a dataset [50], Graz University of Technology). Each node has several natural neighbors (up-left, up, up-right, left, right, bottom-left, bottom, and bottom-right). Using the $4-th$ electrode as an example, the $(1,3,4,5,9,10,11)-th$ electrodes are all its neighbors. Based on SPEE, A can be defined:

$$A_{ij} = \begin{cases} 1, & if\ v_i\ and\ v_j\ are\ neighbors\ or\ i=j \\ 0, & if\ v_i\ and\ v_j\ are\ not\ neighbors \end{cases} \tag{8}$$

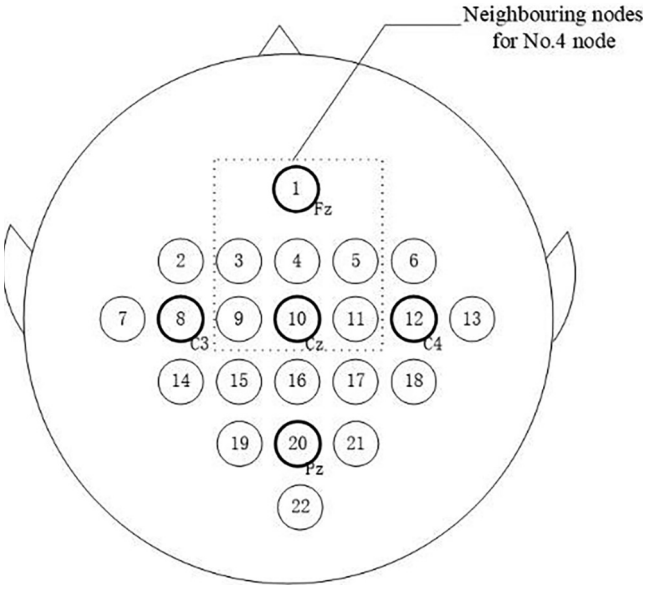


Fig. 2 – The positions of 22 EEG electrodes from the BCICIV-2a dataset.

Then, according to formula (5), the normalized adjacency matrix $\tilde{A}$ is obtained.

2). Method based on Pearson's correlation matrix (PCM). Reference [47] demonstrated that Pearson's correlation matrix could effectively represent the topological relationship between electrodes by conducting comprehensive experiments. For a dataset $X \in R^{C\times(M\times T)}$, the calculation of the correlation coefficient between the electrodes is shown in formula (9):

$$p_{ij} = \frac{cov(X_i, X_j)}{\sqrt{var(X_i)var(X_j)}} \tag{9}$$

where M is the number of trials, $(M \times T)$ represents the sum of the discretized time points from all trials for each electrode, $cov(X_i, X_j)$ represents the covariance between the $i-th$ electrode and $j-th$ electrode, and $var(X_i)$ represents the variance of the $i-th$ electrode. Therefore, A can be defined as:

$$A_{ij} = p_{ij} \tag{10}$$

Then, according to formula (5), the normalized adjacency matrix $\tilde{A}$ is obtained.

Graph embedding is carried out after convolution and is based on the EEG electrodes. The time-domain features are lower-noise and more robust than the raw EEG signals. Moreover, the adjacency matrix is learnable because different subjects may have different brain network connections [42]. On the one hand, the adjacency matrix is symmetric. On the other hand, to reduce the number of learnable parameters, this paper uses only $\frac{n(n-1)}{2}$ (n is the number of nodes) parameters to learn the adjacency matrix, which means that the diagonal element A_{ii} is untrainable and is 1 by default. More importantly, the features obtained by the k-order graph filtering are not added according to formula (6) but are concatenated based on the feature channels, as shown in formula (11):

$$\tilde{x} = \tilde{A}^0 \times x \oplus \tilde{A}^1 \times x \oplus, \cdots, \oplus \tilde{A}^k \times x \tag{11}$$

where $\oplus$ represents channel-based concatenation. According to the formula (11), the features obtained by every order graph filtering can be retained to enhance the feature representations. In addition, the number of feature channels is$(k+1)*F1$. Therefore, for the first convolution layer, combined with formula (11), the mathematical formula (2) is changed to:

$$h^1 = \left(\tilde{A}^0 \times \sigma\left(x * W^1 + b^1\right)\right) \oplus \left(\tilde{A}^1 \times \sigma\left(x * W^1 + b^1\right)\right) \oplus, \cdots, \oplus \left(\tilde{A}^{k-1} \times \sigma\left(x * W^1 + b^1\right)\right) \tag{12}$$

3.4. Datasets and implementation details

Fig. 3 shows the timing scheme of a typical motor imagery experiment.

BCICIV-2a Dataset[50]: This dataset is widely applied to study the performance of different decoder models for motor imagery EEG signals. Twenty-two EEG electrodes recorded the EEG signals at a 250 Hz sampling rate, and the signal frequency range is 0.5 Hz–100 Hz. There are a total of 9 subjects and each subject participated in 2 sessions. Every session included 288 four-class motor imagery (left hand, right hand, feet, and tongue) trials. The first session is used for training, and the second session is used for testing. For the network to better decode the signals, previous studies [34] used trials of 4.5 s, i.e., each trial contains 0.5 s before the motion cue onset. Thus, there are 2592 trials $x \in R^{22\times1125}$ for the training set and testing set. We do not use any preprocessing or data augmentation for the BCICIV-2a dataset.

High-Gamma Dataset (HGD) [34]: HGD is larger than the BCICIV-2a dataset, including four-class motor imagery tasks (left hand, right hand, foot, and rest). A total of 128 electrodes recorded the EEG signals at a 500 Hz sampling rate. There are a total of 14 subjects. For each subject, approximately 880 trials are included in the training set, and approximately 160 trials are included in the testing set by the dataset provider before release. Based on the strategy in [34], we select 44 EEG electrodes with a high correlation to the motor imagery tasks, resample the EEG signals to 250 Hz, and use a 0.5 Hz high-pass filter to filter the signals. We also use trials of 4.5 s as in BCICIV-2a. Thus, for HGD, $x \in R^{22\times1125}$. The dataset is preprocessed with the exponentially weighted moving average method, and we do not use any data augmentation.

Implementation Details: The k,F1, and F2 in Table 1 correspond to 1, 8, and 16, respectively, and the PCM initialization method is used by default. Stochastic gradient descent with the Adam update rule minimizes formula (4). The learning rate of the Adam optimizer parameters is 0.001, and the dropout probability is 0.2 for all dropout layers. The epochs are 1600, and an early stopping strategy is adopted. For the BCICIV-2a dataset, the regularization weight λ is 0, and the batch size is 32. For HGD, the regularization weight λ is 0.0001, and the batch size is 16. Models are trained and tested in a PyTorch environment on an NVIDIA RTX 3080 graphics processing unit (GPU) for high-performance computing. In addition, there are 2311 + 864 k trainable parameters for BCICIV-2a and 3378 + 1216 k trainable parameters for HGD.

Evaluation Metric: Classification accuracy, the Area Under ROC Curve (ROC-AUC), and k-score (kappa score) are used to evaluate the proposed method. The k-score is used to evaluate the classification performance of the algorithm and remove the impact of random classification. It's calculation expression is as follows:

$$k - score = \frac{acc - p_e}{1 - p_e} \tag{13}$$

where acc is mean accuracy and P_e represents the probability or accuracy of a random guess.

The standard deviation is calculated for all the classification accuracy, ROC-AUC, and k-score.

We also use the total number of model parameters (#parameters) and multiply-accumulate operations (MACCs) to measure model complexity at inference time. MACCs for different CNN layers is calculated by:

temporal_conv (i.e., conv2d):

$$I_1 \times I_2 \times F_{in} \times F_{out} \times H_{out} \times W_{out} \tag{14}$$

depthwise_conv:

$$I_1 \times I_2 \times F_{in} \times 2 \times H_{out} \times W_{out} \tag{15}$$

separable_conv:

$$(I_1 \times I_2 + F_{out}) \times F_{in} \times H_{out} \times W_{out} \tag{16}$$

conv1d:

$$I_1 \times F_{in} \times F_{out} \times W_{out} \tag{17}$$

dense:

$$F_{in} \times F_{out} \tag{18}$$

where I_1 and I_2 stand for the width and height of the kernel size, respectively. F_{in} and F_{out} stand for the number of chan-

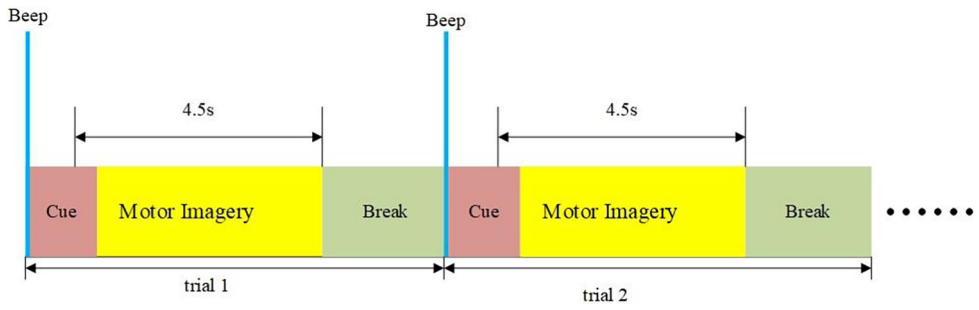


Fig. 3 – A timing scheme of a motor imagery EEG acquisition experiment. The beep and cue are used to notice and indicate the subject to perform the motor imagery task. The window of 4.5 s is the duration of interest for our experiments.

nels of the tensors. And, the H_{out} and W_{out} are the height and width of the tensors, respectively.

Information density is used to evaluate the balance between accuracy and the number of parameters. It's calculated as the ratio of the accuracy (between 0 and 1) and #parameters in millions, as shown by:

$$Information\ density = \frac{Acc}{\#parameters} \tag{19}$$

Furthermore, we report the p-value via the one-tailed Wilcoxon signed-rank test of classification accuracy, whose null hypothesis is that the performance of our method is not better than the other method. The statistics are the sum of the ranks of the differences above zero.

4. Experiment results

4.1. Comparison with state-of-the-art methods

Shallow ConvNet [34]: This network only contains three layers (two convolution layers and one fully connected layer), but the classification accuracy exceeds FBCSP on the BCICIV-2a dataset. This structure's advantage derives from using fewer parameters in exchange for generalizability. Inspired by this strategy, the design of shallow structures is popular in EEG signal classification tasks.

SCCNet [51]: It also only contains two convolution layers. However, the first layer is the spatial convolution layer, and the second is the temporal convolution layer, which is the reverse order of the two convolution layers of Shallow ConvNet.

MCNN [52]: This network implements a feature fusion method based on multi-branch CNNs. The branches are responsible for dealing with EEG signals at various frequency bands and have different layers. This strategy proves that multi-branch feature fusion enhances the feature representations that decode EEG signals.

CNN + LSTM (serial)[53]: CNN and LSTM are combined with being serial. It combines a one-dimensional convolutional neural network (1D CNN) with long short-term memory (LSTM) to classify motor imagery tasks. This method abandons the convolution in the space domain and focuses on extracting the time representation.

CNN + LSTM (parallel) [54]: The structure is parallel. In particular, features from CNN and LSTM are fused into the classifier layer to make decisions.

EEG_TCNet [55]: This network combines EEG_Net with a temporal convolutional network (TCN) capable of exponentially extending the receptive field size while increasing the number of parameters linearly without suffering from exploding or vanishing gradient issues, which contrasts other time-series networks such as LSTM networks.

NG-CRAM [48]: There is data-level graph embedding in NG-CRAM. The adjacency matrix is initialized based on SPEE and untrainable.

CDAN [37]: This network is a conditional domain adaptation neural network whose conditional domain discriminator is adversarial with the label classifier to learn commonly shared intra-subject EEG features. During training, this strategy is equivalent to simultaneously carrying out unsupervised training (in which the training set is composed of all subjects) and supervised training (in which the training data comprises only a targeted subject). Thus, the model is trained with sufficient EEG signals.

GCNs-Net [47]: This network is the first to be wholly composed of GNNs (except for the last layer) for motor imagery classification tasks, proving that GNNs are beneficial for decoding EEG signals and exhibit advanced performance.

Table 2 shows the comparison results of the six methods on the BCICIV-2a dataset. SCCNet and MCNN do not provide k-scores. EEG_GENet achieves the highest mean accuracy of 79.59 % and the highest mean k-score of 0.73 due to its ability to extract robust EEG representations from the embedded graph information. From Table 2, it is also observed that the mean accuracy of EEG_GENet improved by at least 2.24 %, and the average k-score improved by at least 0.03 when compared to other methods. Specifically, EEG_GENet obtains the best classification results for subjects 3, 4, 8, and 9, and the accuracy improved by at least 0.63 %, 1.66 %, 3.75 %, and 1.73 %, respectively. Note that the accuracy of subjects 3, 7, and 9 exceeds 90 %. In addition, EEGNet is equivalent to EEG_GENet with $k = 0$ and achieves 76.51 % mean accuracy and 0.68 mean k-score. When EEGNet is embedded in 1-order graph information, i.e., EEG_GENet, the mean accuracy and the mean k-score increase by 3.09 % and 0.05, respectively. Additionally, the standard deviation of EEG_GENet's accuracy is higher than Shallow ConvNet, SCCNet, two kinds of CNN + LSTM, EEG_TCNet, and NG-CRAM, but is lower than EEGNet.

Table 3 presents the comparison results of the four methods on HGD. Because other methods do not provide k-scores, we use only the classification accuracy for comparison. The mean accuracy of EEG_GENet is only 0.22 % lower than that of GCNs_Net. Specifically, EEG_GENet obtains the highest classification accuracy for subjects 1, 2, 4, 6, 7, 10, and 12. In addition, EEGNet only achieves 94.33 % mean accuracy. However, when EEGNet is embedded in 1-order graph information, i.e., EEG_GENet, the mean accuracy increases from 94.33 % to 96.02 % (increase by 1.69 %), and the standard deviation decreases from 6.64 to 4.61 % (decrease by 2.03 %). We also report the p-value and statistics corresponding to the one-tailed Wilcoxon signed-rank test of mean accuracy; the p-value is 0.063(42.5). Approximately half of the sensors lost meaningful signals for subject 14 in the test set, which significantly affects evaluation. Therefore, to verify the proposed method more comprehensively, this paper also evaluates the test set that excludes subject 14. This evaluation shows that the mean accuracy of EEG_GENet exceeds GCNs-Net by approximately 0.5 %, and EEG_GENet achieves the lowest standard deviations among the seven networks.

4.2. Feature-level graph embedding vs data-level graph embedding

This subsection we show why we need the feature-level graph embedding rather than the data-level embedding. Data-level embedding can be obtained by exchanging stage 1 and stage 2 in Fig. 1. The features extracted from stage 1 are replicated so that the number of trainable parameters is as high as EEG_GENet, and this strategy eliminates the influence of the

Table 2 – Comparison with the state-of-the-art methods on the BCICIV-2a dataset.

Subject		1	2	3	4	5	6	7	8	9	Mean	Std	p-value (statistics)
Shallow ConvNet*	acc%	84.38	62.15	90.28	73.96	70.49	56.94	**91.67**	86.11	81.94	76.48	12.38	/
	k-scores	0.79	0.49	0.87	0.65	0.6	0.43	**0.88**	0.81	0.75	0.7	0.16	/
SCCNet	acc%	78.82	59.38	89.93	68.40	67.71	64.93	80.56	80.56	76.74	74.11	9.60	/
	k-scores	/	/	/	/	/	/	/	/	/	/	/	/
MCNN	acc%	**90.21**	63.40	89.35	71.16	62.82	47.66	90.86	83.72	82.32	75.72	15.24	/
	k-scores	/	/	/	/	/	/	/	/	/	/	/	/
CNN + LSTM (serial)	acc%	87.84	**73.95**	83.79	72.65	**76.66**	**67.29**	73.37	79.60	77.42	76.95	**6.18**	/
	k-scores	**0.84**	**0.65**	0.78	0.64	0.65	**0.56**	0.56	0.73	0.70	0.68	**0.09**	/
CNN + LSTM (parallel)*	acc%	81.25	64.93	89.58	71.18	74.31	67.01	77.43	85.07	84.72	77.28	8.57	/
	k-scores	0.75	0.53	0.86	0.62	**0.66**	**0.56**	0.7	0.8	0.8	0.7	0.11	/
EEG_TCNet	acc%	85.77	65.02	94.51	64.91	75.36	61.40	87.36	83.76	78.03	77.35	11.58	/
	k-scores	0.81	0.53	0.93	0.53	0.67	0.49	0.83	0.78	0.71	0.7	0.15	/
NG-CRAM	acc%	85.07	63.54	94.79	71.18	70.49	62.50	85.76	83.33	77.08	77.08	10.99	/
	k-scores	0.80	0.51	0.93	0.62	0.61	0.50	0.80	0.77	0.69	0.69	0.14	/
EEGNet	acc%	86.11	61.81	91.32	62.50	68.06	59.03	88.19	82.99	88.54	76.51	13.34	/
	k-scores	0.81	0.49	0.88	0.50	0.57	0.45	0.84	0.77	0.85	0.68	0.18	/
EEG_GENet	acc%	82.64	64.58	**95.14**	**74.31**	72.92	57.99	90.97	**87.50**	**90.27**	**79.59**	12.85	0.027 (39.0)
	k-scores	0.76	0.53	**0.94**	**0.66**	0.64	0.44	**0.88**	**0.83**	**0.87**	**0.73**	0.18	/

Both SCCNet and EEG_TCNet have two versions: fixed and variable. Fixed version means that all hyperparameters and networks' parameters keep consistent for all subjects, while variable version means that those parameters could be changed for different subjects to get optimal results. For fair comparison, this paper uses the fixed version. The p-value and statistics are obtained via the one-tailed Wilcoxon signed-rank test of mean accuracy, whose null hypothesis is that the performance of EEG_GENet is not better than the baseline EEGNet on BCICIV-2a dataset. The best results are marked in bold. * Reproduced.

Table 3 – Comparison with the state-of-the-art networks on HGD.

Subject	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Mean1/ Mean2	Std1/ Std2	P-value (statistics)
CDAN	95.00	96.30	96.30	98.10	**99.40**	95.00	93.70	**98.80**	97.50	92.50	92.50	95.60	95.60	87.50	95.30/ 95.98	3.09/ 2.21	/
Shallow ConvNet*	95.62	97.50	99.38	97.50	98.13	98.13	**94.34**	98.13	93.12	91.88	**98.75**	96.88	96.23	79.38	95.36/ 96.58	5.09/ 2.27	/
MSFBCNN*	95.00	96.88	**100**	98.75	96.88	**98.13**	91.19	95.00	96.88	92.50	97.50	98.12	96.86	83.75	95.53/ 96.43	4.31/ 2.46	/
CNN + LSTM (parallel)*	94.38	92.50	95.63	97.50	93.75	90.00	91.19	93.13	91.88	95.63	83.75	96.25	94.97	71.88	91.60/ 93.12	6.62/ 3.54	/
NG-CRAM*	95.00	94.38	99.38	97.50	93.75	95.62	94.34	95.00	94.38	93.75	83.75	97.5	94.97	76.88	93.29/ 94.56	5.88/ **3.66**	/
GCNs-Net	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**	**90.43**	**96.24/** 96.67	**2.60/** 2.07	/
EEGNet*	**96.88**	95.63	99.38	**99.38**	96.26	92.50	**94.34**	93.13	**100**	96.25	83.75	97.50	94.97	80.63	94.33/ 95.38	5.64/ 4.20	/
EEG_GENet	**96.88**	**98.75**	98.75	**99.38**	96.25	**98.13**	**94.34**	93.75	98.13	**96.88**	98.13	**98.75**	94.97	81.25	96.02/ **97.16**	4.61/ **1.84**	0.063 (42.5)

Mean1 and std1 are obtained via all subjects, while calculation of mean2 and std2 exclude subject 14. The p-value and statistics are obtained via the one-tailed Wilcoxon signed-rank test of mean accuracy, whose null hypothesis is that the performance of EEG_GENet is not better than the baseline EEGNet on HGD. The best results are marked in bold. * Reproduced.

Table 4 – Evaluation results from two different graph embedding networks on the BCICIV-2a dataset.

Subject		1	2	3	4	5	6	7	8	9	Mean	Std
EEGNet	acc%	86.61	61.82	91.32	62.50	68.06	59.03	88.19	82.99	88.54	76.51	13.34
	k-scores	0.81	0.49	0.88	0.5	0.57	0.45	0.84	0.77	0.85	0.68	**0.18**
Data_level	acc%	**89.58**	62.84	92.44	61.11	67.01	53.47	87.15	86.46	89.93	76.88	15.53
	k-scores	**0.86**	0.50	0.93	0.48	0.56	0.38	0.83	0.82	0.86	0.69	0.21
Duplicate	acc%	85.76	63.19	94.10	57.64	70.49	**61.81**	85.76	85.42	88.19	76.93	13.60
	k-scores	0.81	0.51	0.92	0.44	0.61	**0.49**	0.81	0.81	0.84	0.69	**0.18**
EEG_GENet	acc%	82.64	**64.58**	**95.14**	**74.31**	**72.92**	57.99	**90.97**	**87.50**	**90.27**	**79.59**	**12.85**
	k-scores	0.76	**0.53**	**0.94**	**0.66**	**0.64**	0.44	**0.88**	**0.83**	**0.87**	**0.73**	**0.18**

EEGNet is the baseline method. The data_level represents the graph embedding carried out for raw EEG signals. Duplicate represents that after the first convolution, the features are duplicated k times, as the blank control experiment. EEG_GENet uses the feature-level graph embedding.

number of trainable parameters. The overall comparison results are shown in Tables 4 and 5. As seen on the BCICIV-2a dataset, the mean accuracy of the data_level and duplicate methods reaches 76.9 %, which is only 0.4 % higher than the baseline EEGNet. On the other hand, the k-scores of the data_-level and duplicate methods are more elevated than baseline by only 0.01. In particular, the data_level method achieves the most significant standard deviation of accuracy and k-scores. In the results on HGD, except for EEG_GENet, the other methods achieve nearly the same mean accuracy and k-scores. These results are consistent with that of the BCICIV-2a dataset. In contrast, EEG_GENet achieves the highest mean accuracy and k-scores, as well as the lowerest standard deviation, on the both datasets.

4.3. *The influence of adjacency matrix initialization methods and whether to learn on EEG_GENet*

This paper proposes two methods to initialize the adjacency matrix: one is based on the spatial position of the EEG electrodes, and the other is based on Pearson's correlation matrix. We also carry out experiments with and without adjacency matrix learning. The random method is used to be the blank control experiment. Fig. 4 shows the comparison results on BCICIV-2a, HGD, and HGD without subject 14. In the results on the BCICIV-2a dataset, the SPEE_un and SPEE achieve 78.3 % mean accuracy, while the PCM_un and PCM achieve 78.9 % and 79.6 % mean accuracy, respectively. These results all exceed EEGNet in terms of classification accuracy. In the results on HGD, the mean accuracy of SPEE_un, SPEE, PCM_un, and PCM reaches 95.6 %, 95.4 %, 95.4 %, and 96.0 %, respectively, which also exceed the mean accuracy of EEGNet. Moreover, there are two outliers in the EEGNet box and the PCM_un method, while the box plot for other methods has only one outlier. Subject 14 is always the outlier. When subject 14 is excluded, Fig. 5(c) shows that SPEE_un achieves the highest mean accuracy, i.e., 97.4 %. In addition, Fig. 4(a)-(c) show that the adjacency matrix initialized randomly works worse than the other methods. Compared with EEGNet, the accuracy of the random method is reduced by 0.6 % on BCICIV-2a and 11.9 % on HGD.

4.4. *The influence of order k on EEG_GENet*

Fig. 5 shows the bar plot of the evaluation results from EEG_GENet with different k on the BCICIV-2a dataset. Note that when k is 3, the mean accuracy reaches just 77.6 %, which is 2.0 % lower than the best result. In particular, subjects 2, 5, 6, 7, and 8 obtain the worst accuracy when k is 3. When k is 1 and 2, EEG_GENet obtains the same mean accuracy, and the best k for different subjects is 1 or 2. Similar results can also be seen in Fig. 6, which shows the evaluation results on HGD. We can see that when k is 1 and 2, EEG_GENet achieves similar results, i.e., 96.0 % and 95.7 %, respectively, with a difference of only 0.3 %. Except for subject 1, whose accuracy reaches the highest at 97.2 % when k is 1, the best accuracy for other subjects is when k is 1 or 2. However, when k is 3, the mean accuracy drops to 83.7 %, which is 12.3 % lower than the best result. Except for subject 1, the accuracy of other subjects also drops significantly. For example, the accuracy of subject 11 drops by 32.3 %, which is a retrograde step. Based on the above results, it can be concluded that when k is 1 or 2, the graph embedding significantly improves the classification accuracy of EEG_GENet. In contrast, the graph embedding with a larger order destroys EEG_GENet's performance.

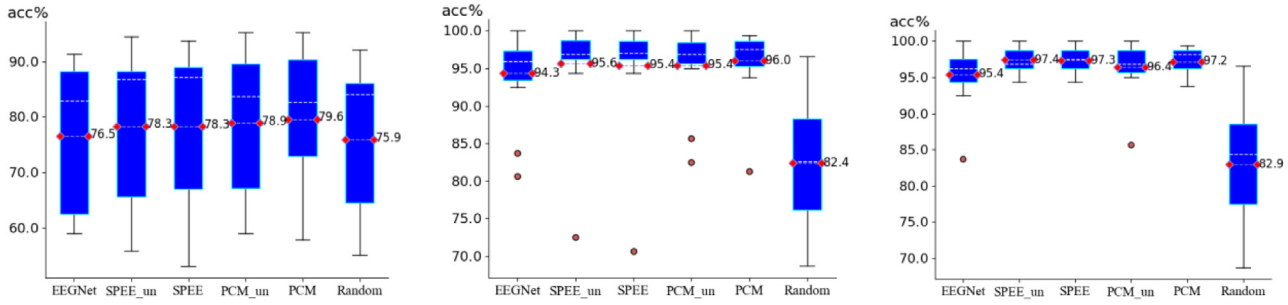
4.5. *Other networks (Shallow ConvNet, Deep ConvNet, and MSFBCNN) based on our feature-level graph embedding method*

The proposed feature-level graph embedding method can also improve other networks, e.g., Shallow ConvNet, Deep ConvNet [34], and MSFBCNN [56], all using a temporal_conv before they convolve all the channels together like EEGNet. Table 6 shows the comparison results of the two networks with and without our method on BCICIV-2a. For Shallow ConvNet_GE, its mean accuracy is 78.01 %, which is higher than Shallow ConvNet by 1.53 %. Specifically, except for subjects 4 and 7, the accuracy is improved on other subjects. Regarding k-scores, Shallow ConvNet based on our method achieves 0.71, also higher than Shallow ConvNet without our method. Deep ConvNet achieves an average accuracy of 77.77 % and an average k-score of 0.71. When our method is applied to Deep ConvNet, its accuracy is improved by 2.63 % to 80.4 %, and the k-score is improved by 0.03 to 0.74. Except for subjects 5 and 6, the accuracy and k-score are improved on other subjects. MSFBCNN_GE achieves 81.41 % accuracy 0.75 k-score. It is an increase in accuracy by 3.86 % and in k-score by 0.05 compared to MSFBCNN. Moreover, except for subject 6, the classification results are all improved on other subjects. In particular, the accuracies of subjects 2 and 7 increase by more

Table 5 – Evaluation results from two different graph embedding networks on HGD.

Subject		1	2	3	4	5	6	7	8	9	10	11	12	13	14	Mean	Std
EEGNet	acc%	96.88	95.63	99.38	**99.38**	**96.26**	92.5	94.34	93.13	**100**	96.25	83.75	97.5	**94.97**	80.63	94.33	5.64
	k-scores	0.96	0.96	0.99	**0.99**	**0.96**	0.93	**0.94**	0.93	**1.00**	0.95	0.84	0.98	**0.95**	**0.81**	0.92	0.08
Data_level	acc%	**98.75**	95.63	**100.00**	**99.38**	95.00	**98.75**	**94.97**	**95.63**	97.50	95.63	78.75	**98.75**	94.34	76.88	94.28	7.23
	k-scores	**0.98**	0.94	**1.00**	**0.99**	0.93	**0.98**	0.93	**0.94**	0.96	0.94	0.72	**0.98**	0.92	0.69	0.92	0.10
Duplicate	acc%	96.88	95.63	99.38	**99.38**	**96.26**	98.13	94.34	95.00	99.38	95.63	83.75	**98.75**	93.71	70.63	94.06	7.83
	k-scores	0.96	0.94	0.99	**0.99**	0.95	**0.98**	0.92	0.93	0.99	0.94	0.78	**0.98**	0.92	0.61	0.92	0.10
EEG_GENet	acc%	96.88	**98.75**	98.75	**99.38**	**96.26**	98.13	94.34	93.75	98.13	**96.88**	**98.13**	**98.75**	**94.97**	**81.25**	**96.02**	**4.60**
	k-scores	0.96	**0.98**	0.98	**0.99**	0.95	**0.98**	0.92	0.92	0.98	**0.96**	**0.98**	**0.98**	0.93	0.75	**0.95**	**0.06**

EEGNet is the baseline method. The data_level represents the graph embedding carried out for raw EEG signals. Duplicate represents that after the first convolution, the features are duplicated k times, as the blank control experiment. EEG_GENet uses the feature-level graph embedding.



(a) BCICIV-2a (b) HGD (c) HGD (excluding subject 14)

Fig. 4 – Box plot of the evaluation results of the adjacency matrix initialized based on SPEE and PCM. EEGNet is the baseline method. The suffix _un means that the adjacency matrix is untrainable. Random means the adjacency matrix is initialized randomly and trainable as the blank control experiment. The red-dotted lines and white dotted lines represent the mean accuracy and median, respectively. The small dots are outliers.

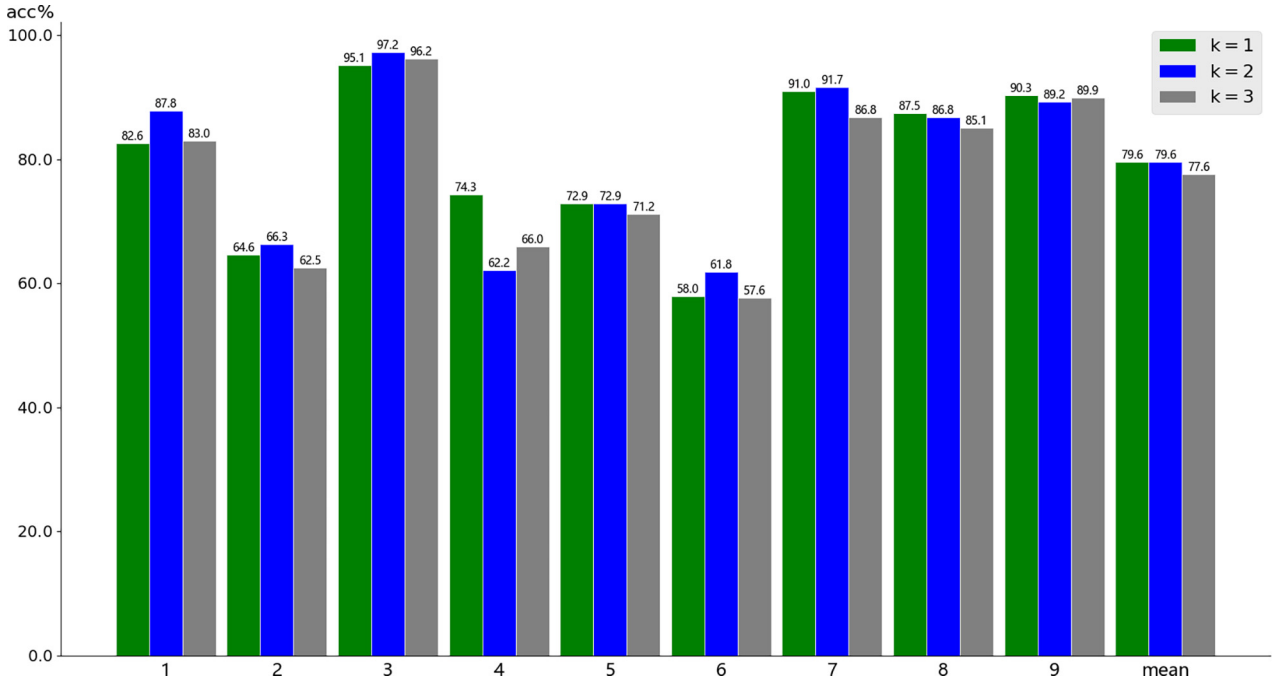


Fig. 5 – Bar plot of the evaluation results of EEG_GENet with different on the BCICIV-2a dataset.

than 10 %, and k-scores increase by more than 0.14. These are huge improvements. Consequently, All networks based on our method have smaller accuracy and k-sorce standard deviations.

5. Discussion

This study uses the proposed feature-level graph embedding method to integrate graph topology information into feature maps and enhance feature representation.

5.1. *Role of feature-level graph embedding*

The feature-level graph embedding is the most significant contribution of this paper. Based on Tables 4 and 5, it is corroborated that EEGNet with the data-level graph embedding hardly improves the accuracy on the BCICIV-2a dataset. On HGD, it gets lower accuracy than the EEGNet baseline. In addition, the standard deviations of accuracy and k-scores are even higher on the both datasets. Those results indicate that the traditional data-level graph embedding method has a limited effect on enhancing practical information. A decrease in stability maybe since the SNR of the original EEG signals is very low, and the graph embedding at the data level introduces more noise into the signals. Our proposed method, however, doesn't have such problems. Because the previous network layers effectively filter out the noise and produce cleaner signals, which are beneficial for transferring information between channels to enhance the practical information. The blank control experiments in Tables 4 and 5 show

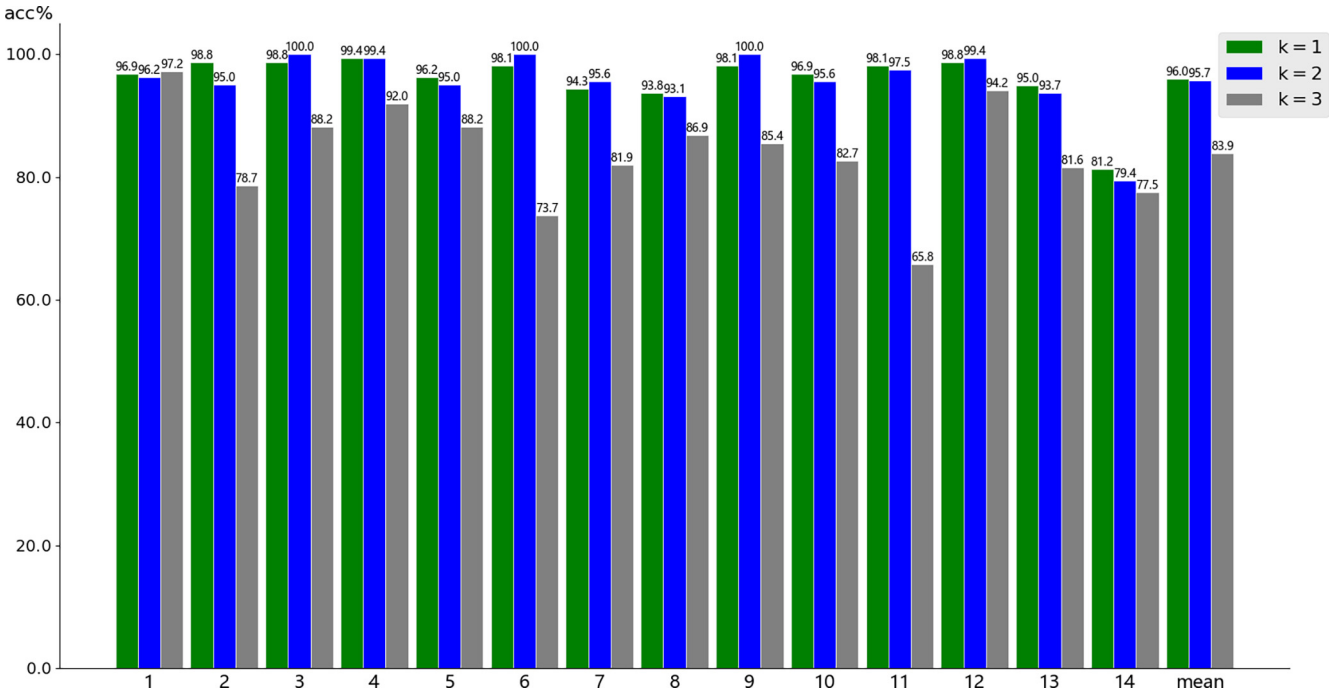


Fig. 6 – Bar plot of the EEG_GENet evaluation results with different on HGD.

no significant difference between the duplicate and the baseline. In other word, EEG_GENet improves the network by embedding the graph information instead of increasing the number of trainable parameters.

5.2. *The universality of feature-level graph embedding*

Most EEG classification networks are designed to follow convolution along time series before spatio consolidation. Thus, our method can be implanted between them. According to Tables 2, 3, and 6, we see that EEGNet, Shallow ConvNet, Deep ConvNet, and MSFBCNN with our proposed method outperform the baselines in terms of accuracy and k-scores. It demonstrates that our method improves the network's ability to decode EEG signals and improve the classification performance. The standard deviations of accuracy and k-scores also decrease, proving that the graph embedding enhances the stability and robustness when classifying different subjects. These baselines have temporal convolution layers that effectively extract shallow features and filter out noise to output cleaner and more useful signals, which are beneficial for transferring information between channels. Moreover, Deep ConvNet is slightly better than MSFBCNN according to the results on the BCICIV-2a dataset, but the accuracy and k-score of MSFBCNN_GE are higher than Deep ConvNet_GE. The performance improvement of EEG_GETNet and MSFBCNN_GE to the baselines EEGNet and MSFBCNN is significant (p-value < 0.05), but it is not the case for Shallow ConvNet_GE and Deep ConvNet_GE (p-value > 0.05). Consequently, these baselines benefit from the feature-level graph embedding strategy, while the degree differs. It is noteworthy that on the BCICIV-2a dataset, EEG_GENet outperforms the baseline (p-value < 0.05), showing that the performance improvement of the proposed model to the baseline is significant. The result in Table 3 indicates that the baseline progress by the feature_level graph embedding on HGD is not significant (p-value > 0.05). The reason may be that the accuracy of the baseline on HGD has already been high, so the benefit imported by the feature_level graph embedding is limited. In a word, the proposed feature-level graph embedding method is adaptive to many kinds of networks.

5.3. *Analysis of comparison*

Generally, EEG_GENet achieves state-of-the-art performance and is superior to the other comparison networks without the feature-level graph embedding. Specifically, on the BCICIV-2a dataset, the accuracy of EEG_GENet is higher than that of Shallow ConvNet, SCCNet, MCNN, CNN + LSTM (serial), CNN + LSTM (parallel), NG-GRAM, and EEG_TCNet; On HGD, the accuracy of EEG_GENet is higher than that of CDAN, Shallow ConvNet, MSFBCNN, CNN + LSTM (parallel), NG-GRAM. Likewise, Shallow ConvNet_GE, Deep ConvNet_GE, and MSFBCNN_GE all have better classification performance than other networks without feature-level graph embedding graph methods. An interesting phenomenon is that the classification performance of the four baselines (EEGNet, Shallow ConvNet, MSFBCNN) is not superior to other comparative networks. After embedding graph information, they gain an edge regarding accuracy and k-score. Thus, the feature-level graph embedding method explicitly represents EEG channels' spatial distribution to help the following network layers extract powerful features. NG-CRAM and GCNs_Net also use graph embedding. However, the graph embedding in NG_CRAM is data-level. As discussed earlier, the data-level graph embedding method has a limited effect on networks because of

Table 6 – The results of other networks based on our feature-level graph embedding method on the BCICIV-2a dataset.

Subject		1	2	3	4	5	6	7	8	9	Mean	Std	P-value (statistics)
Shollow ConvNet*	acc%	84.38	62.15	90.28	**73.96**	70.49	56.94	**91.67**	86.11	81.94	76.48	12.38	/
	k-scores	0.79	0.49	0.87	**0.65**	0.60	0.43	**0.88**	0.81	0.75	0.70	**0.16**	/
Shollow ConvNet _GE	acc%	**84.72**	**64.24**	**91.67**	71.18	**72.22**	**57.64**	90.63	**87.15**	**82.64**	**78.01**	**12.14**	0.180 (31.5)
	k-scores	0.79	**0.52**	**0.89**	0.62	**0.63**	**0.46**	**0.88**	**0.83**	**0.77**	**0.71**	**0.16**	/
Deep ConvNet*	acc%	77.08	55.90	90.97	74.31	**78.82**	68.40	90.97	81.25	82.29	77.77	10.95	/
	k-scores	0.76	0.39	0.79	0.71	**0.69**	0.55	0.89	0.78	**0.79**	0.71	0.16	/
Deep ConvNet_GE	acc%	**83.33**	**60.76**	**88.54**	**84.38**	77.08	68.40	**92.36**	**85.07**	**83.68**	**80.40**	**10.04**	0.062 (29.0)
	k-scores	**0.78**	**0.48**	**0.85**	**0.79**	**0.69**	**0.58**	**0.90**	**0.80**	0.78	**0.74**	**0.13**	/
MSFBCNN*	acc%	84.03	61.11	93.4	76.39	**73.61**	60.07	82.29	86.11	80.90	77.55	11.14	/
	k-scores	0.79	0.48	0.91	0.69	**0.65**	0.47	0.76	0.81	0.75	0.70	0.15	/
MSFBCNN_GE	acc%	**87.85**	**71.53**	**94.10**	**78.13**	72.22	**64.24**	**93.40**	**87.16**	**84.03**	**81.41**	**10.45**	0.010 (42.0)
	k-scores	**0.84**	**0.62**	**0.92**	**0.71**	0.63	**0.52**	**0.91**	**0.83**	**0.79**	**0.75**	**0.14**	/

The suffix _GE means that the feature-level graph embedding is applied to the networks. p-value and statistics are obtained via the one-tailed Wilcoxon signed-rank test of mean accuracy, whose null hypothesis is that the baselines based on our method don't have the better performance than the baselines on BCICIV-2a dataset. The best results are marked in bold. * Reproduced.

low-SNR signals. Hence, NG_CRAM is worse than EEG_GENet in terms of accuracy, k-scores, and standard deviation. GCNs_Net has no convolution layer and can't take advantage of the powerful feature extraction capability of CNNs. Combined with Tables 2 and 6, we report a statistical significance test, as seen in Table 7. The four baselines based on our method all outperform SCCNet significantly (p-value < 0.05). It is worthwhile to explain that the accuracy of EEG_GeNet is lower than Deep ConvNet_GE, while EEG_GeNet exceeds MCNN significantly, but Deep ConvNet_GE doesn't. The reason is that the test results of some subjects in Deep ConvNet_Ge are still lower than the results from MCNN. MSFBCNN_GE outperforms all the state-of-the-art networks significantly except for CNN + LSTM (serial), achieving the best results. The reason may be that parallel multiscale filter banks in MSFBCNN_GE produce multiscale signals, which is more suitable for feature-level graph embedding to enhance practical information.

To further comprehensively discuss the advantages of our method, Table 8 shows five metrics (Acc, ROC-AUC, #parameter, MACCs, information density) of our methods and the start-of-the-art networks that can be reproduced on the BCICIV2a dataset. EEG_GENet, Shallow ConvNet_GE, Deep ConvNet_GE. MSFBCNN_GE have the top four ROC-AUC, signifying that the superior performance of our method again. NG-CRAM has the most training parameters and MACCs, requiring huge computing memory and time, which go against the real-time performance of portable devices. Also, it is observed that the feature-level graph embedding method inevitably increases the number of parameters and MACCs since the graph embedding requires additional channels and computational effort, but the sacrifice is worthwhile for better classification performance. Especially, the information density of Deep ConvNet_GE increases, signifying that the feature-level graph embedding method can also increase the information density. In general, EEG_GENet achieves an excellent balance between complexity and classification performance. On the one hand the depthwise separable convolutions significantly reduce the number of parameters and MACCs. On the other hand, the feature-level graph embeddings improve feature representation.

5.4. *Analysis of ablation experiments*

Adjacency matrix initialization and "whether to learn" influence EEG_GENet. As shown in Fig. 4, EEG_GENet achieves better performance than the baseline method regardless of whether the adjacency matrix is based on SPEE or PCM and whether the adjacency matrix is trainable or not, again demonstrating the advantage of the graph embedding. When the adjacency matrix is initialized based on SPEE, EEG_GENet achieves nearly the same result regardless of whether the adjacency matrix is trainable. Thus, training a SPEE-initialized adjacency matrix is unnecessary. Furthermore, according to the blank control experiments, it makes sense to add prior knowledge to the initialization of the adjacency matrix. The initialization method based on PCM is better than SPEE on the BCICIV-2a dataset, while they exhibit similar performance on HGD. The difference between the two datasets may be due to the different EEG electrodes. HGD has more EEG electrodes, which provide more comprehensive physical topology information; in such cases, the SPEE method is more helpful in supplementing features from the network.

Fig. 7 visualizes the adjacency matrix to observe its changes intuitively. The adjacency matrix initialized based on the SPEE changes slightly after learning from each pair of heat maps. In contrast, the adjacency matrix initialized based on the PCM changes considerably after learning. We also report the mean-square error (MSE) between the initialized adjacency matrix and the after-training adjacency matrix for measuring the degree of change of the adjacency matrix after training all subjects, as shown in Table 9. The MSE values are consistent with Fig. 7. The reason may be that the SPEE method has more practical topology information for many subjects, and there is no risk of over-learning. In comparison, the PCM method is data-driven and contains more individual information and considerable noise. Thus, the weights in the adjacency matrix must be adjusted continuously.

Based on the results in Figs. 4 and 5, it can be concluded that when k is 1 or 2, the graph embedding strategy significantly improves the classification accuracy of EEG_GENet. In contrast, the graph embedding with a larger order destroys EEG_GENet's performance due to possible over-smoothing or over-fitting (The larger the order, the more parameters need to be trained).

Table 7 – p-value (statistics).

	SCCNet	MCNN	CNN + LSTM (serial)	CNN + LSTM* (parallel)	EEG_TCNet	NG-CRAM*
EEG_GENet	**0.037(38.0)**	**0.027(39.0)**	0.248(29.0)	0.125(33.5)	0.150(32.9)	0.082(35.0)
Shollow ConvNet _GE	**0.049(37.0)**	0.064(36.0)	0.455(24.0)	0.312(21.5)	0.285(28.0)	0.163(25.0)
Deep ConvNet_GE	**0.006(43)**	0.125(33.0)	0.102(34.0)	0.104(27.0)	0.180(31.0)	0.082(35.5)
MSFBCNN_GE	**0.004(44)**	**0.006(43.0)**	0.064(36.0)	**0.049(37.5)**	**0.020(40.0)**	**0.004(44.0)**

The first column contains four baselines based on feature-level graph embedding method. The first row contains the start-of-the art networks. p-value and statistics are obtained via the one-tailed Wilcoxon signed-rank test of mean accuracy, whose null hypothesis is that the performance of baselines based on feature-level graph embeding method is not better than the normal networks. The significant results (p-value < 0.05) are marked in bold.

Table 8 – Five metrics of our method and the start-of-the-art networks on the BCICIV2a dataset.

	Acc	ROC-AUC	#Parameter	MACCs	Information dentisty
CNN + LSTM (parallel)*[54]	77.28 ± 8.57	84.85 ± 5.70	290.92 K	54.87 M	2.66
NG-CRAM*[48]	77.08 ± 10.99	84.45 ± 8.65	734.43 K	1078.16 M	1.05
EEGNet*[36]	76.51 ± 13.34	84.34 ± 8.90	2.55 K	13.20 M	290.91
Shallow ConvNet*[34]	76.48 ± 12.38	84.03 ± 8.25	47.36 K	64.09 M	16.15
Deep ConvNet*[34]	77.77 ± 10.95	85.18 ± 7.30	284.48 K	38.10 M	2.73
MSFBCNN*[56]	77.55 ± 11.14	85.03 ± 7.40	158.56 K	315.25 M	4.89
EEG_GENet	79.59 ± 12.85	86.41 ± 7.19	3.68 K	13.38 M	216.28
Shallow ConvNet_GE	78.01 ± 12.04	85.60 ± 8.09	82.80 K	103.83 M	9.42
Deep ConvNet_GE	80.40 ± 10.04	86.93 ± 6.70	298.46 K	53.38 M	2.78
MSFBCNN_GE	81.41 ± 10.45	87.43 ± 7.94	299.92 K	481.70 M	2.71

EEG_GENet, Shallow ConvNet_GE, Deep ConvNet_GE, MSFBCNN_GE are applied with our feature-level graph embedding method, while the other networks not. * Reproduced.

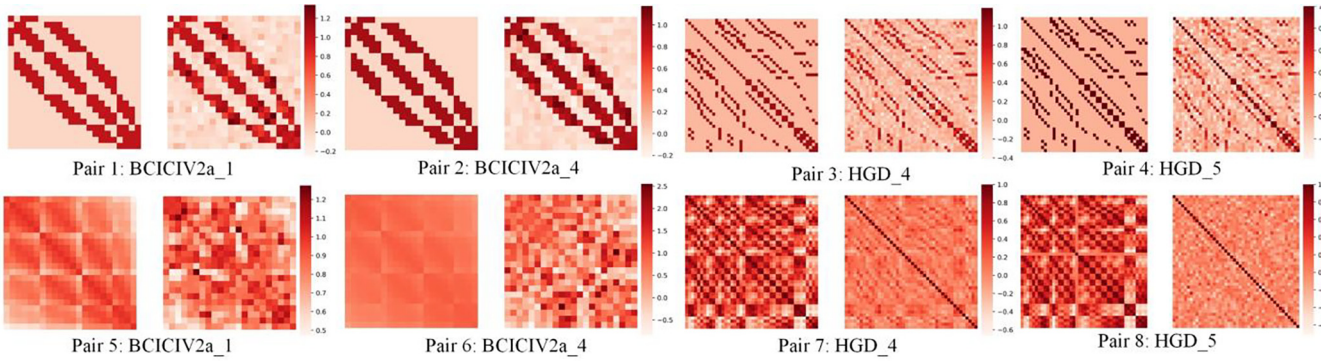


Fig. 7 – Heat map of the adjacency matrix. The adjacency matrix in the first row is initialized based on SPEE, while in the second row, it is initialized based on PCM. For each pair, the left plot is created before training, while the right plot is created after training. Subject 1 and subject 4 in BCICIV2a dataset and subject 4 and sujbect 5 in HGD are randomly selected to study.

Table 9 – MSE of the adjacency matrix before and after training.

Dataset	PCM	SPEE	The ratio of PCM to SPEE
BCICIV-2a	0.020	0.008	2.50
HGD	0.025	0.014	1.79

Moreover, the graph embedding with large orders causes the information in each node to be equal, which is harmful to EEG_GENet when extracting distinctive features.

6. Conclusion

This paper proposes a feature-level graph embedding EEG signals method for motor imagery classification. Specifically, time-domain features are obtained by the first convolution. Then, the adjacent matrix, which acts as a graph filter, is used to perform graph convolution where the time-domain features embed the topology information. The results on two public motor imagery EEG datasets (BCICIV 2a and HGD) show that our proposed method achieves superior performance to the state-of-the-art methods. The feature-level graph embedding method achieves significantly higher classification accuracy than data-level graph embedding methods. Although the initialization method based on SPEE is slightly inferior to the method based on PCM, the former does not have to train to enable the network to obtain advanced classification performance, while the latter needs to train to maximize the network's ability to decode the EEG signals. Other experiments show that the small graph filter order is the best choice for graph embedding [46]. We also apply the proposed method to other popular networks and obtain similar results as EEGNet. In short, our method can be used as an addable component of networks to improve their classification accuracy and even get state-of-the-art performance. Future work will focus on the graph embedding of EEG signals of different frequency bands because the relationship between the electrodes from signals of varying frequency bands still differs and requires further attention.

CRediT authorship contribution statement

Huiyang Wang: Data curation, Writing – original draft, Methodology. **Hua Yu:** Conceptualization, Methodology. **Haixian Wang:** Investigation, Conceptualization, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the editor and the anonymous reviewers for their constructive suggestions. This work was supported by the National Natural Science Foundation of China under Grant 62176054 and the University Synergy Innovation Program of Anhui Province under Grant GXXT-2020-015.

REFERENCES

[1] Cantillo-Negrete J, Carino-Escobar RI, Carrillo-Mora P, Barraza-Madrigal JA, Arias-Carrión O. Robotic orthosis compared to virtual hand for brain-computer interface feedback. Biocybern Biomed Eng 2019;39(2):263–72.
[2] Zapała D, Zabielska-Mendyk E, Augustynowicz P, Cudo A, Jaśkiewicz M, Szewczyk M, et al. The effects of handedness on sensorimotor rhythm desynchronization and motor-imagery BCI control. Sci Reports 2020;10(1):1–1.
[3] Tong J, Zhu D. Multi-phase cycle coding for SSVEP based brain-computer interfaces. Biomed Eng Online 2015;14(1):1–3.
[4] Ajami S, Mahnam A, Abootalebi V. Development of a practical high frequency brain-computer interface based on steady-state visual evoked potentials using a single channel of EEG. Biocybern Biomed Eng 2018;38(1):106–14.
[5] Meng J, Streitz T, Gulachek N, Suma D, He B. Three-dimensional brain-computer interface control through simultaneous overt spatial attentional and motor imagery tasks. IEEE Trans Biomed Eng 2018;65(11):2417–27.
[6] Santamaria L, James C. Using brain connectivity metrics from synchrostates to perform motor imagery classification in EEG-based BCI systems. Healthcare Technol Lett 2018;5 (3):88–93.
[7] Yu Y, Zhou Z, Liu Y, Jiang J, Yin E, Zhang N, et al. Self-paced operation of a wheelchair based on a hybrid brain-computer interface combining motor imagery and P300 potential. IEEE Trans Neural Syst Rehabil Eng 2017;25(12):2516–26.
[8] Frolov AA, Mokienko O, Lyukmanov R, Biryukova E, Kotov S, Turbina L, et al. Post-stroke rehabilitation training with a motor-imagery-based brain-computer interface (BCI)-controlled hand exoskeleton: A randomized controlled multicenter trial. Front Neurosci 2017;11:400.
[9] Chinbat O, Lin JS. Prosthetic arm control by human brain. In: 2018 IEEE International Symposium on Computer, Consumer and Control (IS3C). p. 54-57.
[10] Xu B, Li W, He X, Wei Z, Zhang D, Wu C, et al. Motor imagery based continuous teleoperation robot control with tactile feedback. Electronics 2020;9(1):174.
[11] Xu F, Miao Y, Sun Y, Guo D, Xu J, Wang Y, et al. A transfer learning framework based on motor imagery rehabilitation for stroke. Sci Rep 2021;11(1):1–9.
[12] Lu RR, Zheng MX, Li J, Gao TH, Hua XY, Liu G, et al. Motor imagery based brain-computer interface control of continuous passive motion for wrist extension recovery in chronic stroke patients. Neurosci Lett 2020;718 134727.
[13] Cuomo G, Maglianella V, Ghanbari Ghooshchy S, Zoccolotti P, Martelli M, Paolucci S, et al. Motor imagery and gait control in Parkinson's disease: techniques and new perspectives in neurorehabilitation. Expert Rev Neurotherap 2022 (just-accepted).
[14] Zeng H, Song A, Yan R, Qin H. EOG artifact correction from EEG recording using stationary subspace analysis and empirical mode decomposition. Sensors 2013;13 (11):14839–59.
[15] Sharma M, Goyal D, Achuth PV, Acharya UR. An accurate sleep stages classification system using a new class of optimally time-frequency localized three-band wavelet filter bank. Comput Biol Med 2018;98:58–75.
[16] Alzahab NA, Alimam H, Alnahhas MS, Alarja A, Marmar Z. Determining the optimal feature for two classes motor-imagery brain-computer interface (L/R-MI-BCI) systems in different binary classifiers. Int J Mech Mech Eng 2019;19 (1):132–50.
[17] Hamedi M, Salleh SH, Noor AM. Electroencephalographic motor imagery brain connectivity analysis for BCI: a review. Neural Comput 2016;28(6):999–1041.
[18] Zhang Y, Liu B, Ji X, Huang D. Classification of EEG signals based on autoregressive model and wavelet packet decomposition. Neural Process Lett 2017;45(2):365–78.
[19] Aggarwal S, Chugh N. Signal processing techniques for motor imagery brain computer interface: A review. Array 2019;1 100003.
[20] Yang J, Ma Z, Shen T. Multi-time and multi-band CSP motor imagery EEG feature classification algorithm. Appl Sci 2021;11(21):10294.
[21] Buccino AP, Keles HO, Omurtag A. Hybrid EEG-fNIRS asynchronous brain-computer interface for multiple motor tasks. PLoS ONE 2016;11(1):e0146610.
[22] Park SH, Lee D, Lee SG. Filter bank regularized common spatial pattern ensemble for small sample motor imagery classification. IEEE Trans Neural Syst Rehabil Eng 2017;26 (2):498–505.
[23] Ghanbar KD, Rezaii TY, Tinati MA, Farzamnia A. Correlation-based regularized common spatial patterns for classification of motor imagery EEG signals. In: 2019 IEEE 27th Iranian Conference on Electrical Engineering (ICEE). p. 1770-1774.
[24] Blinowska KJ, Lachert P, Kamiński M, Żygierewicz J, Liebert A. Causal coupling of low frequency oscillations during movement imagination-A multimodal study. In: 2021 International Conference on Bioengineering and Biomedical Signal and Image Processing (BIOMESIP). p. 107–111.
[25] Rozado D, Duenser A, Howell B. Improving the performance of an EEG-based motor imagery brain computer interface using task evoked changes in pupil diameter. PLoS ONE 2015;10(3):e0121262.
[26] Rodriguez-Ugarte MD, Iáñez E, Ortiz-Garcia M, Azorín JM. Effects of tDCS on real-time BCI detection of pedaling motor imagery. Sensors 2018;18(4):1136.
[27] Misawa T, Matsuda J, Hirobayashi S. A single-trial nulticlass classification of various motor imagery tasks for EEG-based brain-computer interface communication. Electron Commun Jpn 2017;100(1):18–26.
[28] Guan S, Zhao K, Yang S. Motor imagery EEG classification based on decision tree framework and Riemannian geometry. Comput Intelligence Neurosci 2019;1:13.
[29] Oikonomou VP, Georgiadis K, Liaros G, Nikolopoulos S, Kompatsiaris I. A comparison study on EEG signal processing techniques using motor imagery EEG data. In: 2017 IEEE 30th International Symposium on Computer-based Medical Systems (CBMS). p. 781–786.
[30] Alazrai R, Abuhijleh M, Alwanni H, Daoud MI. A deep learning framework for decoding motor imagery tasks of the same hand using eeg signals. IEEE Access 2019;7:109612–27.
[31] Ak A, Topuz V, Midi I. Motor imagery EEG signal classification using image processing technique over GoogLeNet deep

learning algorithm for controlling the robot manipulator. Biomed Signal Process Control 2022;72 103295.
[32] Ortiz-Echeverri CJ, Salazar-Colores S, Rodríguez-Reséndiz J, Gómez-Loenzo RA. A new approach for motor imagery classification based on sorted blind source separation, continuous wavelet transform, and convolutional neural network. Sensors 2019;19(20):4541.
[33] Chaudhary S, Taran S, Bajaj V, Sengur A. Convolutional neural network based approach towards motor imagery tasks EEG signals classification. IEEE Sens J 2019;19(12):4494–500.
[34] Schirrmeister RT, Springenberg JT, Fiederer LD, Glasstetter M, Eggensperger K, Tangermann M, et al. Deep learning with convolutional neural networks for EEG decoding and visualization. Hum Brain Mapp 2017;38(11):5391–420.
[35] Ang KK, Chin ZY, Zhang H, Guan C. Filter bank common spatial pattern (FBCSP) in brain-computer interface. In: 2008 IEEE International Joint Conference on Neural Networks (CNN).p. 2390–2397.
[36] Lawhern VJ, Solon AJ, Waytowich NR, Gordon SM, Hung CP, Lance BJ. EEGNet: A compact convolutional neural network for EEG-based brain-computer interfaces. J Neural Eng 2018;15(5) 056013.
[37] Tang X, Zhang X. Conditional adversarial domain adaptation neural network for motor imagery EEG decoding. Entropy 2020;22(1):96.
[38] Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: 2017 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). p. 4700-4708.
[39] Qiao W, Bi X. Deep spatial-temporal neural network for classification of EEG-based motor imagery. In: 2019 Proceedings of the 2019 International Conference on Artificial Intelligence and Computer Science (AICS). p. 265-272.
[40] Izzuddin TA, Safri NM, Othman MA. Compact convolutional neural network (CNN) based on sincnet for end-to-end motor imagery decoding and analysis. Biocybern Biomed Eng 2021;41(4):1629–45.
[41] Mattioli F, Porcaro C, Baldassarre G. A 1D CNN for high accuracy classification and transfer learning in motor imagery EEG-based brain-computer interface. J Neural Eng 2022;18(6) 066053.
[42] Bassett DS, Sporns O. Network neuroscience. Nat Neurosci 2017;20(3):353–64.
[43] Bruna J, Zaremba W, Szlam A, LeCun Y. Spectral networks and locally connected networks on graphs. arXiv:1312.6203, Dec.2013.
[44] Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. Adv Neural Inf Process Syst 2016;29:3844–52.
[45] Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. arXiv:1609.02907, Sep. 2016.
[46] Nt H, Maehara T. Revisiting graph neural networks: All we have is low-pass filters. arXiv:1905.09550, May. 2019.
[47] Lun X, Jia S, Hou Y, Shi Y, Li Y, Yang H, et al. GCNs-net: A graph convolutional neural network approach for decoding time-resolved eeg motor imagery signals. arXiv:2006.08924, Jun. 2020.
[48] Zhang D, Chen K, Jian D, Yao L. Motor imagery classification via temporal attention cues of graph embedded EEG signals. IEEE J Biomed Health Inf 2020;24(9):2570–9.
[49] Li Y, Zhong N, Taniar D, Zhang H. MutualGraphNet: A novel model for motor imagery classification. arXiv:2109.04361, Sep. 2021.
[50] Brunner C, Leeb R, Müller-Putz G, Schlögl A, Pfurtscheller G. "BCI Competition 2008-Graz data set A," Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces). Graz Univ Technol 2008;16:1–6.
[51] Wei CS, Koike-Akino T, Wang Y. Spatial component-wise convolutional network (SCCNet) for motor-imagery EEG classification. In: 2019 IEEE 9th International IEEE/EMBS Conference on Neural Engineering (NER). p. 328-331.
[52] Amin SU, Alsulaiman M, Muhammad G, Mekhtiche MA, Hossain MS. Deep Learning for EEG motor imagery classification based on multi-layer CNNs feature fusion. Future Generat Comp Syst 2019;101:542–54.
[53] Lu P, Gao N, Lu Z, Yang J, Bai O, Li Q. Combined CNN and LSTM for motor imagery classification. In: 2019 IEEE 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI). p. 1-6.
[54] Li H, Ding M, Zhang R, Xiu C. Motor imagery EEG classification algorithm based on CNN-LSTM feature fusion network. Biomed Signal Process Control 2022;72 103342.
[55] Ingolfsson TM, Hersche M, Wang X, Kobayashi N, Cavigelli L, Benini L. EEG-TCNet: An accurate temporal convolutional network for embedded motor-imagery brain-machine interfaces. In: 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC). p. 2958–2965.
[56] Wu H, Niu Y, Li F, Li Y, Fu B, Shi G, et al. A parallel multiscale filter bank convolutional neural networks for motor imagery EEG classification. Front Neurosci 2019;13:1275.