IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 November 2023, accepted 28 November 2023, date of publication 5 December 2023,
date of current version 13 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3339647

RESEARCH ARTICLE

Multi-Class fNIRS Classification Using an Ensemble of GNN-Based Models

MINSEOK SEO, EUGENE JEONG, AND KYUNG-SOO KIM, (Member, IEEE)
Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, Yuseong-gu, Daejeon 34141, Republic of Korea

Corresponding author: Kyung-Soo Kim (kyungsookim@kaist.ac.kr)

This work was supported by the Agency for Defense Development by the Korean Government under Grant UD210018ID.

ABSTRACT Functional near-infrared spectroscopy (fNIRS) is a neuroimaging technique used to estimate brain activity by measuring local hemodynamic changes. Due to its high spatial resolution, fNIRS is being actively researched as a control signal in the field of brain-computer interface (BCI). Extraction of effective features and accurate classification of signals have always been the focus of research. Previous studies have often converted fNIRS data into images based on the relative positions of the measurement channels and utilized convolutional neural networks (CNN) for classification. However, image representation cannot fully express the non-Euclidean characteristics of the brain signal. In this paper, we propose an approach for single-trial, multi-class fNIRS classification using a graph representation and a graph neural network (GNN). Specifically, a class-specific graph was constructed for each class to incorporate both positional and task-dependent functional connectivity (FC) information. The GNN-based models were then trained on each of the obtained class-specific graphs to have specificity for the corresponding class. Finally, the stacking ensemble learning with a gating network was introduced to weight the models for the final prediction. The proposed method was evaluated on a public dataset consisting of three types of overt movements. The results were compared with baseline models based on support vector machine (SVM) and CNN, using different image conversion methods. The best-performing baseline model achieved an average ternary classification accuracy of 68.71%, whereas the proposed model achieved a classification accuracy of 72.31% for the single model, and 75.47% for the ensemble model.

INDEX TERMS Brain-computer interface, ensemble learning, functional connectivity, functional near-infrared spectroscopy, graph neural network.

I. INTRODUCTION

Functional near-infrared spectroscopy (fNIRS) is a non-invasive neuroimaging technique for estimating brain activity by measuring oxygenation and hemodynamic changes [1]. The change in concentration of oxygenated (HbO) and reduced (HbR) hemoglobin is calculated from the measured attenuation of near-infrared light. Fluctuations in HbO and HbR reflect brain activity due to neurovascular coupling. With its ability to reveal cortical activity in the natural environment [2], fNIRS was initially used to monitor infants [3] or patients with psychiatric disorders [4]. Furthermore, due to its portability and ease of use compared to other neuroimaging techniques such as electrocorticography (ECoG) [5], positron emission tomography (PET) [6], magnetoencephalography (MEG) [7], and functional magnetic resonance imaging (fMRI) [8], fNIRS has gained attention in the field of brain-computer interfaces (BCI) along with electroencephalography (EEG) [9]. BCI aims to establish an interface that estimates the user's attention through measuring brain activity and translates it into an external device control signal [10]. Originally developed to assist individuals with severe motor disabilities [11], [12], [13], BCI has expanded its application to various domains such as machine control [14], [15], mental state monitoring [16], [17], and entertainment [18], [19]. For these applications, the multi-class classification problem of the brain signal is crucial.

fNIRS provides complementary information to EEG by measuring different aspects of brain activity. EEG has superior temporal resolution as it measures the electrical

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Kafiul Islam.

activity of the brain, but lacks spatial specificity in localizing the source of the signals. On the other hand, fNIRS has better spatial resolution as it measures changes in local HbO and HbR levels, but has lower temporal resolution and carries a time delay of the brain hemodynamic response [20]. Thus, it can be said that proper extraction of the spatial feature is the key to classifying fNIRS signals.

Early research in fNIRS signal classification focused on finding robust statistical features that can be used to characterize the measured signal. Commonly used features include mean and peak values of HbO and HbR concentrations, as well as variance, slope, skewness, and kurtosis [21]. Each feature was evaluated with the performance of machine-learning-based models, mainly linear discriminant analysis (LDA) and support vector machine (SVM). As useful hand-crafted features were established and interest in deep learning increased, the choice of classifier and feature extraction method shifted to deep-learning-based approaches. Several methods were employed, including convolutional neural network (CNN), long short-term memory (LSTM), recurrent neural network (RNN), and multi-layer perceptron (MLP). Among these methods, CNN was used most frequently [22] due to its ability to capture and extract spatial features from images [23]. In order to apply convolutional layers, fNIRS signals must first be converted into image data.

A straightforward method of converting the fNIRS signal into an image is by concatenating the measured time series [24], [25], or the extracted temporal features [26] along the channel dimension. In some studies, single-channel time series were first converted to a virtual image using Gramian angular fields (GAF) and concatenated [27], [28]. With these methods, the resulting images do not represent the spatial characteristics of the measurement channels.

To represent the relative position of each channel, the fNIRS signal can be rearranged into three-dimensional tensors based on the sensor location [29], [30], [31]. However, this method assumes that the distance between channels implies connectivity, which may not reflect the internal neural connection. Since functional networks in the brain have a non-Euclidean character [32], mapping fNIRS channels to image pixels may lead to unwanted connections between adjacent channels or neglect the possible connections between distant channels.

By considering each measurement channel as a node and connectivity as an edge, a graph may better represent the underlying brain network compared to an image. Representing the brain network as a graph is a well-known concept in the field of connectome research [33], [34]. Resting-state [35] and task-dependent [36] functional connectivity (FC) have been used to identify brain connectivity. Once represented as a graph, the brain network can be classified using a graph neural network (GNN). Li et al. proposed BrainGNN [37] to classify seven task states of the brain network using the graph representation of the fMRI signal. Pearson correlation and partial correlation of averaged fMRI data for each subject and task were used to calculate the FC.

However, the application of FC in a single-trial BCI scheme remains relatively questionable. Demir et al. [38] represented the EEG signal as a graph by connecting each electrode pair, k-nearest neighbors, and electrodes with a distance below threshold to classify error-related potentials (ErrP) and rapid serial visual presentation (RSVP) datasets. Zhong et al. [39] used correlation, coherence, and distance thresholding to initialize the connections between EEG channels for the emotion recognition task. Among the tried methods, distance thresholding performed the best. This may be a consequence of the inter-trial variability of FC and the low spatial resolution of EEG. To the best of our knowledge, there only exists a single research on applying GNN to fNIRS. Qiao Yu et al. [40] were able to distinguish patients with depression using a complete graph. Subjects performed a fixed task, and the average FC of all trials was used as the edge weight.

In this paper, we propose a step-by-step approach for using graph representation to classify fNIRS signals in a single-trial, multi-class scheme. Inspired by previous studies, we define graphs for each class using averaged task-dependent FC instead of defining a graph per trial. Additionally, we apply position-based pruning and add far-channel connections. Therefore, multiple graph representations are obtained for each fNIRS signal. We then propose a spatial module based on GNN to extract spatial features from the graphs. The spatial module provides an initial prediction from the given graph through graph convolution [41]. We train spatial modules using different graph representations with the expectation that they will specialize in distinguishing a certain class. Finally, these obtained class-specific models are treated as base models to train a meta-model through ensemble learning. The final prediction is made by taking a weighted average of the predictions from each class-specific model. The weight assigned to each model was determined using a gating network [42] and stacked-ensemble learning [43]. To validate the effectiveness of the proposed GNN-based ensemble model, experiments were conducted using open-access data. To summarize, the main contributions of this paper are as follows:

- We propose a graph representation method of fNIRS data using the relative position and task-dependent FC between channels.
- We propose a GNN-based model that can extract and classify spatial features from graph-converted fNIRS data.
- We propose an ensemble method using a gating network to combine predictions made by models trained with different class-specific graphs.
- We conduct experiments on public data [44] consisting of three motor execution tasks. Experimental results show that the proposed models outperform SVM and CNN-based baseline models.

The rest of the paper is organized as follows: In Section II, the dataset and pre-processing steps applied to the experiments are explained. In Section III, preliminaries of GNN and FC are outlined. In Section IV, the proposed graph representation method and modules are introduced. In Section V, the setup and results of conducted experiments are presented and discussed. Finally, Section VI concludes the paper.

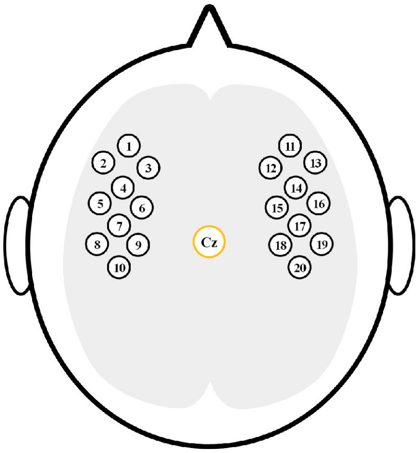


FIGURE 1. **Location of the measurement channels. Channels 1-10 are located around C3 (channel 9) and channels 11-20 are located around C4 (channel 18) [44]. The location of Cz is shown for convenience.**

II. MATERIALS

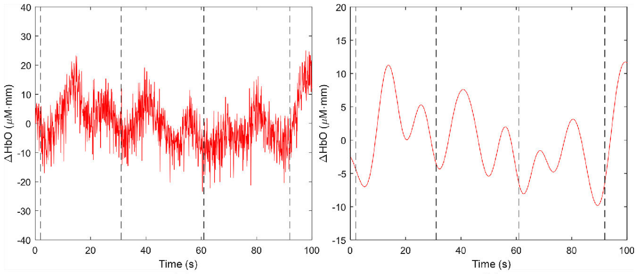
A. DATASET

A publicly available dataset [44] for the three-class classification of fNIRS signals was used. Data were collected from 30 subjects, consisting of 17 males and 13 females with an average age of 23.4 years. Concentration changes of HbO (Δ HbO) and HbR (ΔHbR) were measured while subjects performed one of three types of motor execution tasks. The tasks consisted of right-hand finger-tapping (RHT), left-hand finger-tapping (LHT), and foot-tapping (FT). Each subject performed 25 trials per task, resulting in a total of 2250 measured trials. Each trial consisted of a 2-second introduction period, a 10-second task period, and an intertrial rest period ranging from 17 to 19 seconds. Eight light sources and eight detectors were placed around the C3 and C4 regions, configuring a total of 20 measurement channels. The locations of the measurement areas were labeled and shown in Fig. 1. The data were measured at a sampling rate of 13.3 Hz.

B. DATA PRE-PROCESSING

Throughout the study, the Δ HbO time series is used for classification. To reduce physiological artifacts and drift components, a third-order Butterworth filter with a passband of 0.01-0.1 Hz was applied. The data were then segmented into epochs containing measurements up to 15 s after the task onset and labeled. The baseline for each epoch was corrected based on the average value of the introduction period.

An example of a raw signal and its filtered signal measured from channel 5 is shown in Fig. 2. This channel was chosen for its strong activity. The plots show that periodic noise such as heartbeat and respiration has been effectively removed.



(a) (b)

FIGURE 2. **(a) Raw and (b) band-pass filtered ΔHbO data measured from a single channel (channel 5). Dashed line indicates task onset period.**

Example signals for each class after baseline correction and segmentation are shown in Fig. 3. It can be observed that each class is hardly distinguishable in a single trial. However, when signals are averaged over all trials, the RHT signal shows high activity.

For some experiments, hand-crafted temporal features were used. Following the literature [44], the Δ HbO within the time windows of 0-5 s, 5-10 s, and 10-15 s were averaged across each channel and used as the temporal feature matrix $\boldsymbol{F}_T \in \mathbb{R}^{20\times 3}$.

III. PRELIMINARIES

In this section, preliminary knowledge about convolutional graph neural networks and functional connectivity is introduced. In addition, the terminology used in the paper is described.

A. CONVOLUTIONAL GRAPH NEURAL NETWORK

An undirected graph can be expressed as $G = (V, E)$, where V is the set of nodes, and E is the set of edges. Let $v_i \in V$ to denote a node, n to denote the number of nodes, and $\varepsilon_{ij} \in E$ to denote an edge between v_i and v_j. Then the adjacency matrix $\boldsymbol{A}$ is defined as an $n \times n$ matrix with A_{ij} equal to 1 if $\varepsilon_{ij} \in E$ and 0 if $\varepsilon_{ij} \notin E$. In the case of a weighted graph, elements A_{ij} are taken to be the weight of ε_{ij}, denoted as e_{ij}. Each node may carry a feature vector, which can be defined as node attributes $\boldsymbol{F} \in \mathbb{R}^{n\times f}$, where f is the length of the feature vector.

The main idea of convolutional GNN is to generate a node embedding by aggregating features from its neighbors. In this study, GCN [41] was utilized. The GCN uses a normalized adjacency matrix to update the node embedding with low numerical instability. Layer-wise propagation rule of GCN is as follows:

$$\boldsymbol{H}^{(l+1)} = \sigma\left(\tilde{\boldsymbol{D}}^{-1/2}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-1/2}\boldsymbol{H}^{(l)}\boldsymbol{\Theta}^{(l)}\right), \tag{1}$$

where $\boldsymbol{H}^{(l)}$ and $\boldsymbol{\Theta}^{(l)}$ denote the graph embedding and weight matrices of layer l, respectively. $\boldsymbol{H}^{(0)}$ corresponds to the input node feature matrix $\boldsymbol{F}$. $\sigma(\cdot)$ denotes the activation function, and $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ is the adjacency matrix with added self-loops. $\tilde{\boldsymbol{D}}$ denotes the degree matrix of $\tilde{\boldsymbol{A}}$, which is a diagonal matrix defined as

$$\tilde{\boldsymbol{D}}_{ii} = \sum\nolimits_j \tilde{\boldsymbol{A}}_{ij}. \tag{2}$$

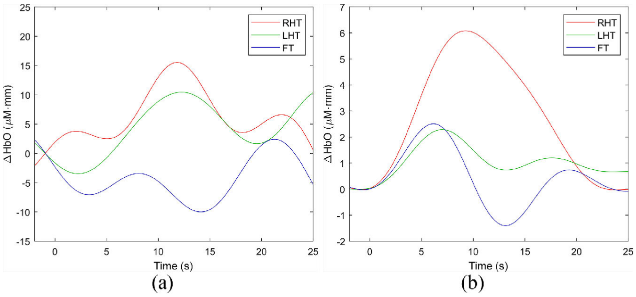


FIGURE 3. Example signals for each class (RHT, LHT, FT) after baseline correction and segmentation. (a) Signal from a single trial. (b) Signal averaged over all trials.

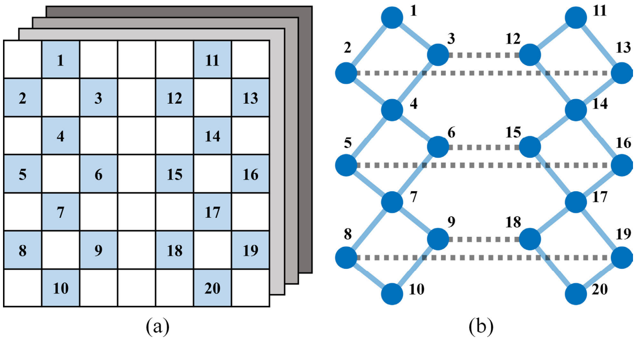


FIGURE 4. Illustration of (a) an image and (b) a graph constructed based on the relative position of the measurement channels. Each channel is labeled with its channel number. The dotted lines indicate additional global connections.

In the case of an undirected and weighted graph, a node-wise formulation can be expressed as follows:

$$x_i^{(l+1)}=\mathbf{\Theta}^{\top}\sum_{j\in N(i)\cup\{i\}}\frac{e_{ij}}{\sqrt{\hat{d}_i\,\hat{d}_j}}x_j^{(l)}, \tag{3}$$

where $x_i^{(l)}$ denotes the feature vector of node i of layer l, $N(i)$ denotes the set of neighboring nodes of node i, and $\hat{d}$ is defined as

$$\hat{d}_i=1+\sum\nolimits_{j\in N(i)}\left|e_{ij}\right|. \tag{4}$$

Absolute values of edge weight e_{ij} were used to consider possible negative weights.

B. FUNCTIONAL CONNECTIVITY

Functional connectivity is defined as the temporal correlation between neurophysiological measurements obtained from distinct brain areas [45]. Coherence, phase locking value, phase lag index, and Pearson correlation are commonly used as indicators of FC. In this paper, Pearson's correlation coefficient, often referred to as correlation, is used due to its ability to evaluate the temporal similarity between time series. The correlation between two time series of Δ HbO measured in channel i and j, denoted as $C^i=\{C_1^i\ldots C_T^i\}$ and $C^j=\{C_1^j\ldots C_T^j\}$, can be expressed as follows:

$$\rho\left(C^i,C^j\right)=\frac{\sum_k^T\left(C_k^i-\overline{C}^i\right)\left(C_k^j-\overline{C}^j\right)}{\sqrt{\sum_k^T\left(C_k^i-\overline{C}^i\right)^2\left(C_k^j-\overline{C}^j\right)^2}}, \tag{5}$$

where $\bar{C}$ denotes the mean of C. The FC of a given time window can be represented as a correlation matrix $\boldsymbol{\rho}=[\rho_{ij}]$, where $\rho_{ij}=\rho\left(C^i,C^j\right)$.

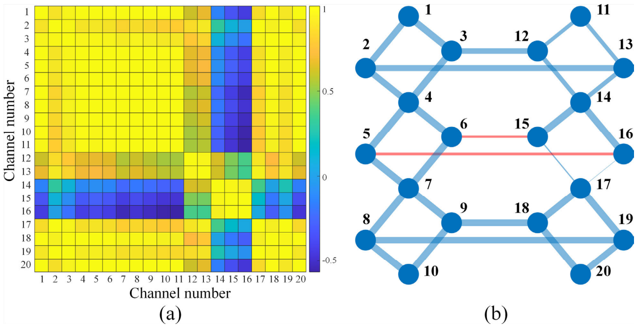


FIGURE 5. (a) A task-dependent correlation matrix and (b) a class-specific graph of the LHT task. The width of the edges indicates the edge weight, which is the correlation between two channels. Edges with negative weights are colored red.

C. TERMINOLOGY

For clarity, we explain some of the terminology used in this paper. *Task-dependent FC* refers to the functional connectivity of the brain while performing a particular task (e.g., RHT, LHT, and FT). When considering a multi-class classification problem, each performed task is referred to as a class. Thus, the term *class-specific* is used to describe an adjacency matrix, a graph, and a model dedicated to a particular class.

IV. METHODS

A. GRAPH CONSTRUCTION

To represent the fNIRS data as a graph, an adjacency matrix was defined in terms of relative position and task-dependent FC. Each node represents a measurement channel, and the edge weight indicates the connectivity between channels.

1) POSITION-BASED ADJACENCY MATRIX

As mentioned earlier, studies [38], [39] have shown that physical distance can be used to express connectivity between channels. Analogous to the CNN approach, connecting nearby channels can be beneficial in capturing local spatial information. Thus, the adjacency matrix $\boldsymbol{A}_{\text{POS}}$ was defined by connecting nodes according to the sensor location depicted in Fig. 1. To take advantage of the information originated from the hemispheric asymmetry [46] and possible far-channel connectivity, the global connections between symmetric channels were also introduced. The weight of all edges was set to 1. The resulting graph is illustrated in Fig. 4 (b).

2) CLASS-SPECIFIC ADJACENCY MATRIX

Next, the class-specific adjacency matrices $\boldsymbol{A}_{\mathrm{RHT}}$, $\boldsymbol{A}_{\mathrm{LHT}}$, and $\boldsymbol{A}_{\mathrm{FT}}$ were defined based on the task-dependent FC of the RHT, LHT, and FT, respectively. To mitigate the influence of inter-trial variability, the task-dependent FCs were calculated using averaged time series. For each motor task, the ΔHbO signals were averaged across all trials to obtain task-dependent ΔHbO time series for each channel. To find a representative FC, a sliding-window scheme was employed [47]. The window size was set to 10 s, which corresponds to the task duration. For each time window, a correlation matrix was computed. The correlation matrix with the highest variance was selected to maximize the difference between active and inactive channels.

TABLE 1. The architecture of the spatial model.

Layer		Channel	Output Size
Input	Adjacency	-	20×20
	Feature	-	$20 \times f$
Graph convolution		$f^{(1)}$	$20 \times f^{(1)}$
Graph convolution		$f^{(2)}$	$20 \times f^{(2)}$
Flatten		-	$1 \times 20f^{(2)}$
Fully-connected		$10f^{(2)}$	$1 \times 10f^{(2)}$
Fully-connected		3	1×3

f denotes the length of the input feature vector, and $f^{(i)}$ denotes the length of the output feature vector of the i_{th} graph convolution layer.

To avoid the effect of post-task changes, the center of the time windows was shifted within the task period. As a result, three task-dependent FCs were obtained and referred to as $\boldsymbol{\rho}_{\mathrm{RHT}}$, $\boldsymbol{\rho}_{\mathrm{LHT}}$, and $\boldsymbol{\rho}_{\mathrm{FT}}$.

The correlation matrix can be directly used as an adjacency matrix. However, it has been reported that dense graphs are prone to information loss caused by over-smoothing [48] when applied to GNN. In addition, Achard et al. [49] suggested that a sparse brain functional network increases the efficiency of the network topology, with a low-cost threshold of sparsity of around 0.1. To offer sparsity, thresholding is often applied to leave connections with high connectivity [37]. However, in the case of fMRI, it is known that a high correlation may be observed among regions with no practical cerebral blood flow. Physiological noise, such as cardiac and respiratory signals, can also lead to high correlations [50]. Since fNIRS data contain signals from the skin, the impact of physiological artifacts is more significant. Thus, thresholding may result in prioritizing connections between inactive channels.

To offer sparsity while preserving the connections between active channels, the Hadamard product between the correlation matrix and $\boldsymbol{A}_{\mathrm{POS}}$ was performed as follows:

$$\boldsymbol{A}_{RHT} = \boldsymbol{A}_{POS} \circ \boldsymbol{\rho}_{RHT}, \tag{6}$$

where $\circ$ denotes the Hadamard product operator. Similar operations were performed to obtain $\boldsymbol{A}_{\mathrm{LHT}}$ and $\boldsymbol{A}_{\mathrm{FT}}$ using $\boldsymbol{\rho}_{\mathrm{LHT}}$ and $\boldsymbol{\rho}_{\mathrm{FT}}$. In this way, connections between active channels can be preserved while incorporating the positional information. The $\boldsymbol{\rho}_{\mathrm{LHT}}$ and $\boldsymbol{A}_{\mathrm{LHT}}$ obtained from the entire dataset are shown in Fig. 5 as an example.

TABLE 2. The architecture of the temporal module.

Layer	Kernel size	Stride	Channel	Output Size
Input	-	-	-	$20 \times 32 \times 1$
1D-convolution	8	4	4	$20 \times 7 \times 4$
1D-convolution	4	3	8	$20 \times 2 \times 8$
1D-convolution	2	2	16	$20 \times 1 \times 16$

B. SPATIAL MODULE

The spatial module was designed to extract spatial features from the graph representations. Inspired by the effectiveness of CNN in the Euclidean domain, the spatial module uses GCN to capture local spatial information from the graph-structured fNIRS data. Then, the readout function is applied to the obtained graph embedding to perform classification. In general, various pooling methods are used as the readout function to generate a fixed-size representation from the graph embedding regardless of the number of nodes [51]. However, since fNIRS data have a fixed number of nodes, a flatten layer can be used as the readout function, similar to CNN. The flattened representation can then be fed into a fully connected layer for classification.

It has been reported that increasing the number of GCN layers can degrade the performance of a model due to the convergence of node features [52]. Generally, it is known that using two to three layers achieves optimal performance. In this study, two layers were used in the spatial module. The overall architecture of the spatial module is summarized in Table 1.

The spatial module takes adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{20\times20}$ and feature matrix $\boldsymbol{F} \in \mathbb{R}^{20\times f}$ as an input. The number of features f was varied in different experiments. When a hand-crafted temporal feature was used, f was set to 3. When a temporal feature was extracted with an additional temporal module, f was set to 16. Features were extracted through two GCN layers with output feature lengths of $f^{(1)}$ and $f^{(2)}$. The flatten layer followed by two fully connected layers was employed to classify the extracted features. The values of $f^{(1)}$ and $f^{(2)}$ were adjusted for different input features. The rectified linear unit (ReLU) was used for activation, and batch normalization was applied after each convolution layer.

C. TEMPORAL MODULE

As mentioned earlier, GCN effectively extracts spatial features from graph-structured data by aggregating local node features. However, it lacks the ability to capture temporal information because the aggregation function of GCN does not consider the locality and order of elements within each node feature. To overcome this limitation, a temporal module was designed to extract temporal features from the time series. The temporal module uses 1D convolution to extract temporal features without integrating signals from different channels.

The module consists of three 1D convolutional layers with ReLU activation, as shown in Table 2. An input feature is a downsampled time series of Δ HbO, denoted as $\boldsymbol{X}_{\Delta \mathrm{HbO}} \in \mathbb{R}^{20\times 32}$. To ensure compatibility with the spatial module, the temporal module should have an output size of $20\times f$. To match the dimension, the features were downsampled to $20 \times 1 \times 16$ by adjusting the kernel size and stride, and then squeezed to a final size of 20×16. The temporal module is placed before the spatial module, and both modules were trained simultaneously to extract relevant features for classification.

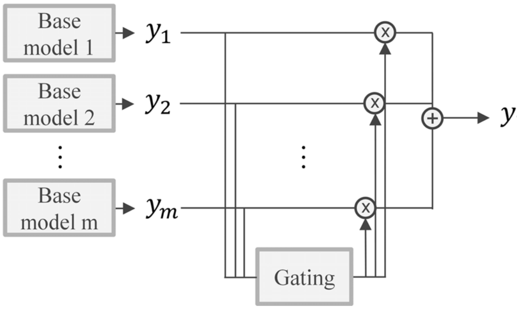


FIGURE 6. A schematic diagram of a gating network proposed for the fusion module. The gating model takes predictions from the base models as an input to obtain the weighting. y_i: prediction made by the i-th base model, y_T: weighted sum of predictions.

D. FUSION MODULE

The model trained with the class-specific graph is expected to distinguish the corresponding class better than others. Inspired by stacking ensemble learning and a mixture of experts [53], a fusion module is proposed to utilize the prediction of each base model to improve the overall performance.

The fusion module uses a gating network to obtain weights for each base model and outputs the final prediction based on these weights. Letting m denote the number of models, the gating network aims to learn a weight vector $W \in \mathbb{R}^m$ that minimizes the loss of the final prediction. The final prediction is given by a weighted sum of the predictions made by the base models, which can be expressed as

$$y_T = \sum\nolimits_i^m w_i y_i, \tag{7}$$

where y_T is the final prediction, y_i is the prediction made by the i-th base model, and $w_i \in W$ is the scalar weight assigned to each base model. A schematic diagram of a gating network is shown in Fig. 6. The weight vector is obtained through two fully connected layers with 64 hidden dimensions as follows:

$$W = \sigma_{\mathrm{soft}}(\mathrm{FL}^{1\times m}(\mathrm{FL}^{1\times 64}([y_1, y_2, \ldots, y_m]))), \tag{8}$$

where $\mathrm{FL}^{1\times m}(\cdot)$ denotes fully connected layer with an output size of $1\times m$, $\sigma_{\mathrm{soft}}(\cdot)$ denotes softmax function, and $[\cdot]$ denotes concatenation.

E. OVERALL MODEL

The overall architecture of the proposed GNN-based ensemble model is depicted in Fig. 7. The ensemble model consists of three main elements: pre-constructed adjacency matrices, base models, and a fusion module. From the training set, graphs representing relative position and task-dependent FC between measurement channels are constructed. Base models, consisting of a temporal module and a spatial module, are individually trained using the fNIRS time series data accompanied by the obtained graph. The fusion module is trained with predictions from the base models to output a weighted sum for the final prediction. For a given dataset, the training and testing process of the overall model is explained in the following.

1) TRAINING PHASE

The first step of the training phase is to obtain adjacency matrices. From predefined $\boldsymbol{A}_{\mathrm{POS}}$ and labeled training data, the average task-dependent FCs and corresponding class-specific adjacency matrices were acquired. Then, each data was converted into four different graphs regardless of its ground truth label, denoted as $G(\boldsymbol{A}_S, \boldsymbol{X}_{\Delta \mathrm{HbO}})$, where $G(\boldsymbol{A}, \boldsymbol{F})$ denotes a graph with adjacency $\boldsymbol{A}$ and node attributes $\boldsymbol{F}$, and $S \in \{\mathrm{POS}, \mathrm{RHT}, \mathrm{LHT}, \mathrm{FT}\}$. Using each of the graph representations, position-based model M_{POS} and class-specific models M_{RHT}, M_{LHT}, and M_{FT} were trained. Finally, the fusion module was trained with obtained base models.

2) TEST PHASE

In the test phase, the adjacency matrices constructed during the training phase are used. As in the training phase, each data is converted to four different graphs. Then, predictions from each model are obtained as follows:

$$y_s = M_s(G(\boldsymbol{A}_s, \boldsymbol{X}_{\Delta \mathrm{HbO}})). \tag{9}$$

Predictions are then fused to a final prediction according to (7).

V. EXPERIMENTS

A. EXPERIMENTAL SETUP

The first part of the experiments was conducted using the hand-crafted temporal features $\boldsymbol{F}_T$ described in Section II to compare the efficiency of extracting spatial features. The spatial module was used as a standalone GNN-based classifier. Using four adjacency matrices $\boldsymbol{A}_{\mathrm{POS}}$, $\boldsymbol{A}_{\mathrm{RHT}}$, $\boldsymbol{A}_{\mathrm{LHT}}$, $\boldsymbol{A}_{\mathrm{FT}}$, and a feature matrix $\boldsymbol{F}_T$ as training data, four models were

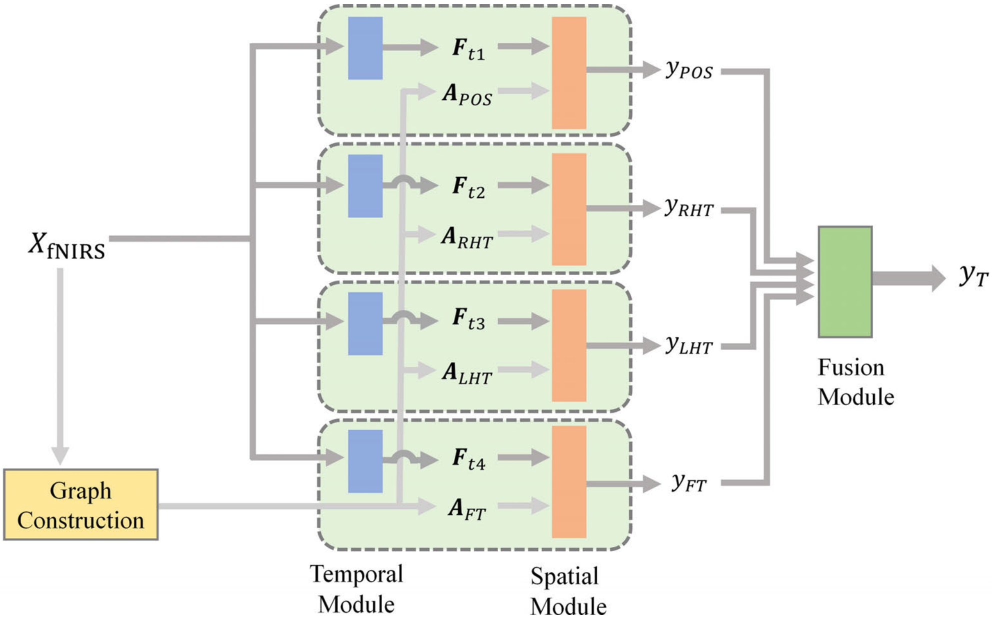


FIGURE 7. The overall architecture of the proposed model. X_{fNIRS}: fNIRS data (time series of ΔHbO), F_{ti}: temporal features extracted from i_{th} temporal module, A_{POS}, A_{RHT}, A_{LHT}, A_{FT}: position-based and RHT, LHT, FT-specific adjacency matrix, y_{POS}, y_{RHT}, y_{LHT}, y_{FT}: prediction made by position-based and RHT, LHT, FT-specific models, y_T: final prediction made by the fusion module.

trained (i.e., $\mathrm{GNN_{POS}}$, $\mathrm{GNN_{RHT}}$, $\mathrm{GNN_{LHT}}$, and $\mathrm{GNN_{FT}}$). The trained models served as a base model for training a gated meta-model $\mathrm{GNN_{Gated}}$ through the fusion module. To evaluate the effectiveness of the fusion module, an average ensemble model $\mathrm{GNN_{Avg}}$ was tested. The predictions from each base model were weighted equally to obtain the final prediction of the average ensemble model.

SVM with a linear kernel and CNN-based model were selected as the baseline. The CNN-based baseline, referred to as $\mathrm{BL_1}$, takes image-converted $\boldsymbol{F}_T$ as input. The feature vector of each channel was relocated as shown in Fig. 4(a), resulting in a 3-channel image of size $7 \times 7 \times 3$ (width × height × number of channels). $\mathrm{BL_1}$ consists of two 2D convolutional layers and two fully connected layers as listed in Table 10 given in the Appendix. After each convolution, ReLU activation and batch normalization were applied.

The output feature length of the spatial model is selected through grid search from the sets $f^{(1)} \in \{8, 16, 32, 64, 128\}$ and $f^{(2)} \in \{1, 2, 4, 8, 16\}$, resulting in $(f^{(1)}, f^{(2)}) = (64, 2)$. The models were trained for 150 epochs with a batch size of 32. The Adam [54] optimizer with a learning rate of 0.01, $(\beta_1, \beta_2) = (0.9, 0.999)$, and a weight decay of 0.0001 was used. The fusion module is trained with a batch size of 16 and zero weight decay. The softmax cross-entropy loss was used for training.

In the second part of the experiments, a time series $\boldsymbol{X}_{\Delta\mathrm{HbO}}$ was used to further improve the performance. The temporal module was placed before the spatial module to extract and feed temporal features. Using four adjacency matrices $\boldsymbol{A}_{\mathrm{POS}}$, $\boldsymbol{A}_{\mathrm{RHT}}$, $\boldsymbol{A}_{\mathrm{LHT}}$, $\boldsymbol{A}_{\mathrm{FT}}$, and a time series $\boldsymbol{X}_{\Delta\mathrm{HbO}}$ as training data, four models were trained (i.e., $\mathrm{T-GNN_{POS}}$, $\mathrm{T-GNN_{RHT}}$, $\mathrm{T-GNN_{LHT}}$, and $\mathrm{T-GNN_{FT}}$). As in the previous experiment, the trained models were used as the base model for training gated meta-model $\mathrm{T-GNN_{Gated}}$ and average ensemble model $\mathrm{T-GNN_{Avg}}$.

Two CNN-based models, $\mathrm{BL_2}$ and $\mathrm{BL_3}$, were used as the baseline. The $\mathrm{BL_2}$ is a single-modal classification model presented in [55], which consists of six 1D convolutional layers and two fully connected layers. Since it was designed for binary classification of 36-channel time series, the layer dimensions were adjusted to fit the data as listed in Table 11 given in the Appendix. $\boldsymbol{X}_{\Delta\mathrm{HbO}}$ was resampled to a size of 20×30 (number of channels × number of time steps). $\mathrm{BL_3}$ is a modified version of FSNet presented in [31]. The $\mathrm{BL_3}$ consists of three 3D convolutional layers followed by a pooling layer and a fully connected layer. The input size was adjusted, and the temporal attention pooling layer was replaced by a max pooling layer as shown in Table 12 given in the Appendix. Using an image conversion method similar to $\mathrm{BL_1}$, $\boldsymbol{X}_{\Delta\mathrm{HbO}}$ was converted to an image sequence of size $8 \times 8 \times 30$ (width × height × time). Zero-padding was applied to the width and height dimensions. For all models, ReLU activation and batch normalization were applied after each convolution. In addition, a dropout rate of 0.5 was applied to the first fully connected layer for regularization.

The number of hidden channels of the spatial model was selected as $(f^{(1)}, f^{(2)}) = (32, 2)$. The $\mathrm{BL_2}$, $\mathrm{BL_3}$, and $\mathrm{T-GNN}$ models were trained for 300 epochs with a batch size of 32. The Adam optimizer with a learning rate of 0.001, $(\beta_1, \beta_2) = (0.9, 0.999)$, and zero weight decay was used. The training condition of a fusion module was the same as in the first experiment. The softmax cross-entropy loss was used.

Each model was trained in a subject-independent manner. A 5-fold cross-validation was conducted to evaluate the average classification accuracy. Since the base models were fitted to the training set, unused data were needed to train the fusion module. To maintain the size of the test set, the data was divided into three subsets with a ratio of 6:2:2. Each subset was designated as a training set for the base model, a training set for the fusion module, and a test set, respectively. Task-specific adjacency matrices were constructed from the training set of the base model. For each fold, all base models were trained using the same training set to avoid data leakage during the ensemble learning process.

B. EXPERIMENTAL RESULTS

Table 3 shows the mean classification accuracy and standard deviation of the proposed and baseline models. Mean accuracy was obtained by averaging the validation accuracy of each fold of 5-fold cross-validation. The additional classification evaluation metrics of the class-specific models are listed in Table 4 - 6.

When the hand-crafted temporal feature $\boldsymbol{F}_T$ was used for training, SVM and BL$_1$ achieved classification performance of 68.66% and 67.91%, respectively. With the same feature, the proposed GNN-based models performed better regardless of the adjacency matrix used, despite the smaller training set.

The position-based model, GNN$_{\mathrm{POS}}$, obtained an average accuracy of 70.62%, which is higher than the accuracy of the class-specific models, GNN$_{\mathrm{RHT}}$, GNN$_{\mathrm{LHT}}$, and GNN$_{\mathrm{FT}}$. For ensemble learning, averaging did not improve the performance. On the other hand, the GNN$_{\mathrm{Gated}}$ achieved an average accuracy of 72.93%, a significant improvement over the base models (paired t-test, $p < 0.01$).

When the time series $\boldsymbol{X}_{\Delta\mathrm{HbO}}$ was used for training, the baseline models BL$_2$ and BL$_3$ achieved an average accuracy of 68.71% and 68.62%, respectively. The proposed GNN-based models with a temporal module outperformed BL$_2$ and BL$_3$, regardless of the adjacency matrix used. The LHT-specific model $\mathrm{T-GNN_{LHT}}$ performed best among the models trained without a fusion module, with an average accuracy of 72.31%, surpassing the performance of the base models trained with $\boldsymbol{F}_T$. The averaged ensemble model performed marginally better than the base models, whereas the $\mathrm{T-GNN_{Gated}}$ significantly outperformed $\mathrm{T-GNN_{LHT}}$, the best-performing base model (paired t-test, $p < 0.01$).

The characteristics of the position-based and task-specific GNN models were further investigated using the classification metrics, namely precision, recall, and F1-score. These metrics were calculated from the combined prediction result of each fold. It can be observed that the recall and F1-score of the class-specific models were highest in the corresponding classes, whereas the position-based models showed balanced performance with high precision. The confusion matrices of the combined prediction results for each model are shown in Table 13-15 given in the Appendix.

TABLE 3. Classification accuracy (Mean ± standard deviation) of proposed and baseline models.

Model	Training data	Mean Accuracy (%)
SVM	$\boldsymbol{F}_T$	68.66±1.26
BL$_1$	$\boldsymbol{F}_T$ (image)	67.91±1.41
GNN$_{\mathrm{POS}}$	$\boldsymbol{A}_{\mathrm{POS}}, \boldsymbol{F}_T$	70.62±0.84
GNN$_{\mathrm{RHT}}$	$\boldsymbol{A}_{\mathrm{RHT}}, \boldsymbol{F}_T$	69.38±1.48
GNN$_{\mathrm{LHT}}$	$\boldsymbol{A}_{\mathrm{LHT}}, \boldsymbol{F}_T$	69.47±0.23
GNN$_{\mathrm{FT}}$	$\boldsymbol{A}_{\mathrm{FT}}, \boldsymbol{F}_T$	70.02±0.51
GNN$_{\mathrm{Avg}}$	$\boldsymbol{A}_{\mathrm{POS}}, \boldsymbol{A}_{\mathrm{RHT}}, \boldsymbol{A}_{\mathrm{LHT}}, \boldsymbol{A}_{\mathrm{FT}}, \boldsymbol{F}_T$	69.82±0.29
GNN$_{\mathrm{Gated}}$	$\boldsymbol{A}_{\mathrm{POS}}, \boldsymbol{A}_{\mathrm{RHT}}, \boldsymbol{A}_{\mathrm{LHT}}, \boldsymbol{A}_{\mathrm{FT}}, \boldsymbol{F}_T$	**72.93±1.29**
BL$_2$	$\boldsymbol{X}_{\Delta\mathrm{HbO}}$	68.71±2.61
BL$_3$	$\boldsymbol{X}_{\Delta\mathrm{HbO}}$ (image sequence)	68.62±3.34
T-GNN$_{\mathrm{POS}}$	$\boldsymbol{A}_{\mathrm{POS}}, \boldsymbol{X}_{\Delta\mathrm{HbO}}$	72.18±1.59
T-GNN$_{\mathrm{RHT}}$	$\boldsymbol{A}_{\mathrm{RHT}}, \boldsymbol{X}_{\Delta\mathrm{HbO}}$	71.56±2.09
T-GNN$_{\mathrm{LHT}}$	$\boldsymbol{A}_{\mathrm{LHT}}, \boldsymbol{X}_{\Delta\mathrm{HbO}}$	72.31±0.21
T-GNN$_{\mathrm{FT}}$	$\boldsymbol{A}_{\mathrm{FT}}, \boldsymbol{X}_{\Delta\mathrm{HbO}}$	71.38±1.84
T-GNN$_{\mathrm{Avg}}$	$\boldsymbol{A}_{\mathrm{POS}}, \boldsymbol{A}_{\mathrm{RHT}}, \boldsymbol{A}_{\mathrm{LHT}}, \boldsymbol{A}_{\mathrm{FT}}, \boldsymbol{X}_{\Delta\mathrm{HbO}}$	72.71±1.97
T-GNN$_{\mathrm{Gated}}$	$\boldsymbol{A}_{\mathrm{POS}}, \boldsymbol{A}_{\mathrm{RHT}}, \boldsymbol{A}_{\mathrm{LHT}}, \boldsymbol{A}_{\mathrm{FT}}, \boldsymbol{X}_{\Delta\mathrm{HbO}}$	**75.47±2.36**

$\boldsymbol{F}_T$: hand-crafted feature, $\boldsymbol{A}_{\mathrm{POS}}, \boldsymbol{A}_{\mathrm{RHT}}, \boldsymbol{A}_{\mathrm{LHT}}, \boldsymbol{A}_{\mathrm{FT}}$: position-based and RHT, LHT, FT-specific adjacency matrix, $\boldsymbol{X}_{\Delta\mathrm{HbO}}$: ΔHbO time series.

TABLE 4. Classification metrics for class-specific models without temporal module.

Model	Class	Precision	Recall	F1-score
GNN$_{\mathrm{POS}}$	RHT	0.7597	0.7333	0.7463
	LHT	**0.7504**	0.6973	0.7229
	FT	**0.6224**	0.6880	0.6536
GNN$_{\mathrm{RHT}}$	RHT	0.7547	**0.7547**	**0.7547**
	LHT	0.7117	0.6880	0.6997
	FT	0.6181	0.6387	0.6282
GNN$_{\mathrm{LHT}}$	RHT	0.7245	0.7293	0.7269
	LHT	0.7391	**0.7253**	**0.7322**
	FT	0.6219	0.6293	0.6256
GNN$_{\mathrm{FT}}$	RHT	**0.7660**	0.7160	0.7402
	LHT	0.7438	0.6773	0.7090
	FT	0.6178	**0.7133**	**0.6621**

The highest metrics for each class are highlighted in bold.

Regarding the task-dependent FCs obtained from the test set, the median of the time windows used to obtain $\boldsymbol{\rho}_{\mathrm{RHT}}$, $\boldsymbol{\rho}_{\mathrm{LHT}}$, and $\boldsymbol{\rho}_{\mathrm{FT}}$ were 9.0 s, 9.5 s, and 5.9 s after task onset, respectively. It is noteworthy that the correlations of global channel pairs (5,16) and (6,15) were negative for RHT and LHT, and positive for FT.

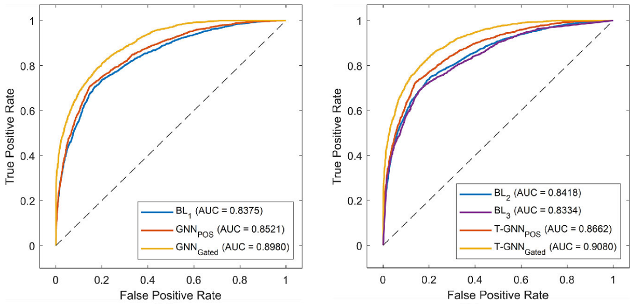
TABLE 5. Classification metrics for class-specific models with temporal module.

Model	Class	Precision	Recall	F1-score
T-GNN$_{\mathrm{POS}}$	RHT	**0.7745**	0.7053	0.7383
	LHT	0.7611	**0.7520**	0.7565
	FT	**0.6429**	0.7080	0.6739
T-GNN$_{\mathrm{RHT}}$	RHT	0.7580	**0.7600**	**0.7590**
	LHT	0.7490	0.7240	0.7363
	FT	0.6429	0.6627	0.6527
T-GNN$_{\mathrm{LHT}}$	RHT	0.7629	0.7080	0.7344
	LHT	**0.7833**	**0.7520**	**0.7673**
	FT	0.6379	0.7093	0.6717
T-GNN$_{\mathrm{FT}}$	RHT	0.7637	0.6893	0.7246
	LHT	0.7726	0.7067	0.7382
	FT	0.6302	**0.7453**	**0.6830**

The highest metrics for each class are highlighted in bold.

TABLE 6. Classification metrics for ensemble models.

Model	Class	Precision	Recall	F1-score
GNN$_{\mathrm{Gated}}$	RHT	0.7966	0.7413	0.7680
	LHT	0.8148	0.7627	0.7879
	FT	0.6706	0.7600	0.7125
T-GNN$_{\mathrm{Gated}}$	RHT	0.7721	0.7680	0.7701
	LHT	0.7797	0.7080	0.7421
	FT	0.6501	0.7133	0.6802



(a) (b)

FIGURE 8. The average ROC curves and AUC of trained models. (a) BL$_1$, GNN$_{\mathrm{POS}}$, and GNN$_{\mathrm{Gated}}$ are compared. (b)BL$_2$, BL$_3$, T − GNN$_{\mathrm{POS}}$, and T − GNN$_{\mathrm{Gated}}$ are compared.

C. DISCUSSION

1) PERFORMANCE OF PROPOSED MODULES

The SVM trained with $\boldsymbol{F_T}$ achieved a classification accuracy well above the chance level, indicating that $\boldsymbol{F_T}$ is a valid temporal feature for the given dataset. Using graph convolution, the spatial module successfully extracted spatial features from the given input. Since GNN$_{\mathrm{POS}}$ and BL$_1$ were trained with common features, the performance advantage of GNN$_{\mathrm{POS}}$ implies that the graph can better represent the fNIRS data compared to the image. Note that $\boldsymbol{A}_{\mathrm{POS}}$ is obtained from the relative positions of the channels, similar to the image conversion method employed by BL$_1$.

The higher performance of T − GNNs over models trained with $\boldsymbol{F_T}$ implies that the temporal module effectively extracted the temporal features. The implementation of the temporal model not only increased the classification accuracy of each base model but also the effect of ensemble learning. Compared to the best-performing base model, the increase in average accuracy was 2.31%p for GNN$_{\mathrm{Gated}}$ and 3.16%p for T − GNN$_{\mathrm{Gated}}$. This may be due to the flexibility of the temporal module, which can focus on different parts of the time series as opposed to the predefined $\boldsymbol{F_T}$. If each base module extracts features from different parts of the signal, the meta-model can generate more generalized predictions.

The performance of the models was also compared using receiver operating characteristic (ROC) curves. ROC curves, which are typically used for binary classifiers, can be extended to multi-class classifiers by treating the classifier as multiple one-versus-rest (OvR) binary classifiers. The ROC curve of each class can then be micro-averaged to obtain an average ROC curve. The average ROC curves of the trained models are shown in Fig. 8. Since a larger area under the ROC curve (AUC) indicates a better classifier performance, it can be seen that GNN$_{\mathrm{POS}}$ performed marginally better than BL$_1$, while GNN$_{\mathrm{Gated}}$ performed significantly better. Similarly, T − GNN$_{\mathrm{POS}}$ performed better than BL$_2$ and BL$_3$, and T − GNN$_{\mathrm{Gated}}$ performed significantly better than others.

TABLE 7. Contingency table for McNemar test to compare GNN$_{\mathrm{POS}}$ and GNN$_{\mathrm{Gated}}$.

		GNN$_{POS}$	
		Correct	Incorrect-
GNN$_{\mathrm{Gated}}$	Correct	1503	139
	Incorrect-	86	522

TABLE 8. Contingency table for McNemar test to compare T − GNN$_{\mathrm{POS}}$ and T − GNN$_{\mathrm{Gated}}$.

		T-GNN$_{\mathrm{POS}}$	
		Correct	Incorrect-
T-GNN$_{\mathrm{Gated}}$	Correct	1549	149
	Incorrect-	75	477

Since the test folds between the models were kept the same during the cross-validation, the McNemar test can be performed. By comparing the prediction result of each data, the two-by-two contingency tables shown in Table 7 and 8

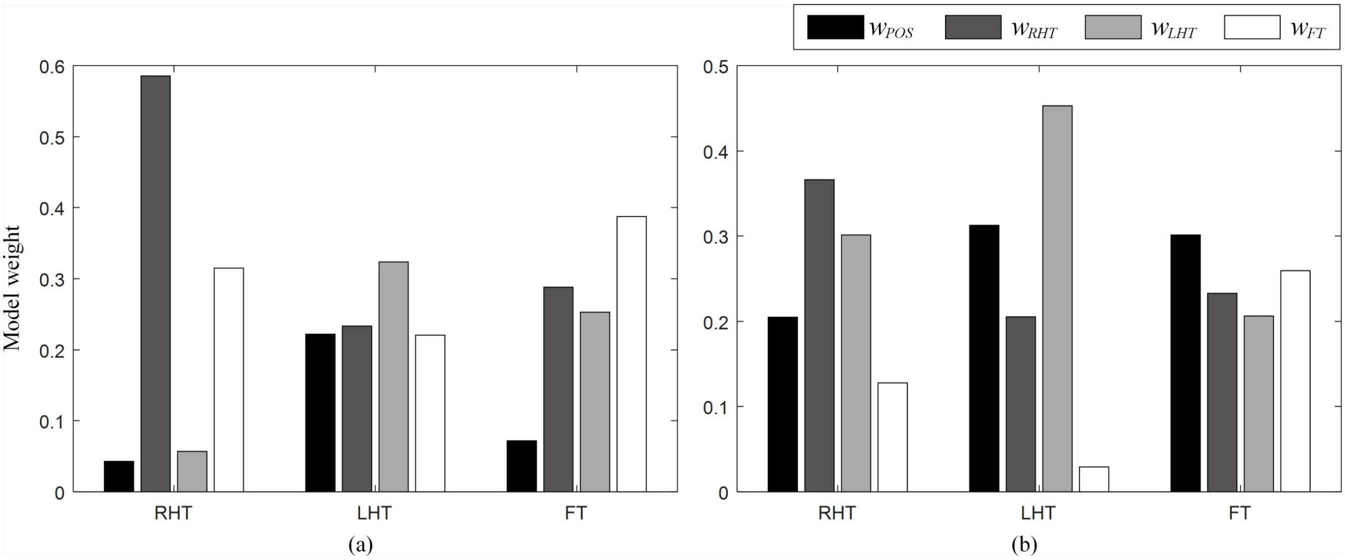


FIGURE 9. Average model weights used by the fusion module to predict each class using (a) the spatial module and (b) the spatial and temporal modules together. w_{POS}, w_{RHT}, w_{LHT}, and w_{FT} denote the weight applied to the position-based model, RHT-specific model, LHT-specific model, and FT-specific model, respectively.

were constructed to compare the ensemble model and the single model.

From the table, the null hypothesis that the predictive performance of the two models is equal can be tested. Using the mid-p McNemar test, the hypothesis can be rejected if the p-value is less than the significance level of 0.05. Since the resulting p-value is 4.99×10^{-4} for Table 7 and 6.43×10^{-7} for Table 8, the null hypothesis can be rejected for both cases. Therefore, the predictive performance of $\mathrm{GNN_{Gated}}$ and $\mathrm{T-GNN_{Gated}}$ is significantly better compared to $\mathrm{GNN_{POS}}$ and $\mathrm{T-GNN_{POS}}$, respectively.

In terms of average classification accuracy, there was no noticeable difference between the class-specific models and the position-based model. However, as mentioned earlier, class-specific models achieved higher F1-score and recall for the corresponding class compared to the position-based model. Thus, if the binary classification of discriminating specific class is considered, class-specific models can perform as a better OvR classifier compared to the position-based model.

Fig. 9 shows the average weights used by the fusion module to predict each class. In the case of $\mathrm{GNN_{Gated}}$, the most weighted model for the data predicted as RHT, LHT, and FT were $\mathrm{GNN_{RHT}}$, $\mathrm{GNN_{LHT}}$, and $\mathrm{GNN_{FT}}$, respectively. In the case of $\mathrm{T-GNN_{Gated}}$, the most weighted model for the data predicted as RHT, LHT, and FT were $\mathrm{T-GNN_{RHT}}$, $\mathrm{T-GNN_{LHT}}$, and $\mathrm{T-GNN_{POS}}$, respectively. The weight of $\mathrm{T-GNN_{FT}}$ was the second highest for data classified as FT. From observations, it can be said that the fusion module is able to assign a higher weight to a class-specific model with correct prediction, or possibly a higher confidence.

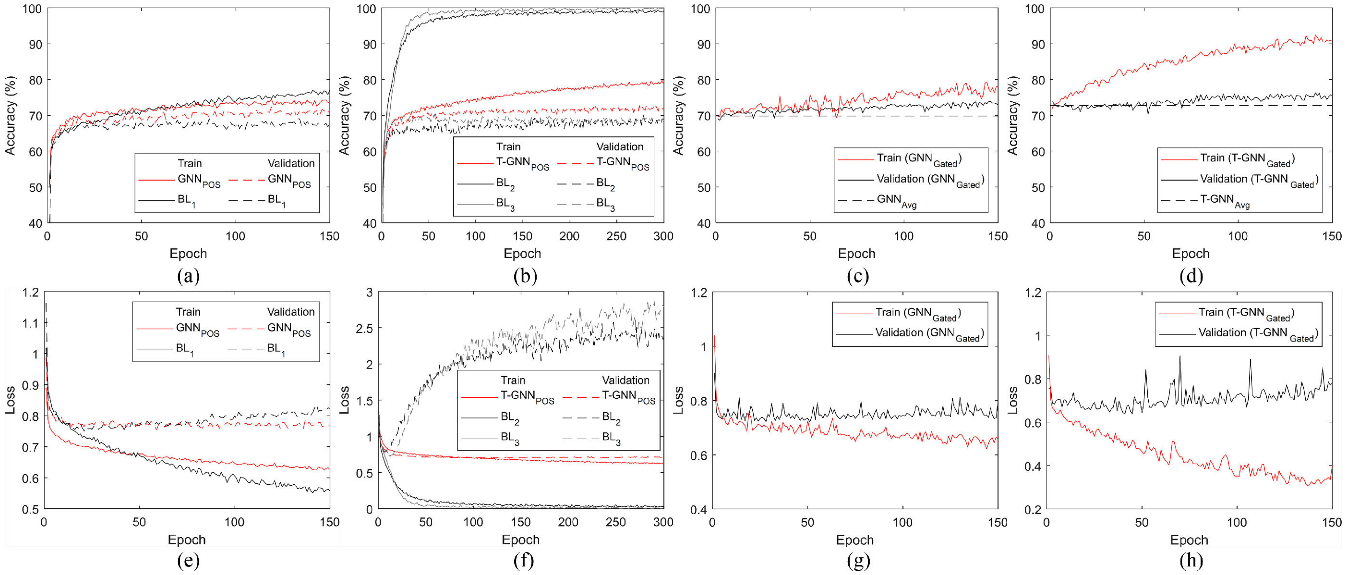
Overall, the proposed GNN-based ensemble model outperformed all baseline models based on SVM and CNNs with different data conversion methods.

While the proposed model overcomes the limitation of position-based image representation by using a graph representation, there are some high-performance models with different approaches. The reported mean accuracy of the high-performance models is shown in Table 9. The fNIRS-T [56] is a model based on a Transformer that takes time series as an input data. Other models utilize computer vision (CV) after encoding the signals into a virtual image for classification [28]. The CV models used were ViT [57], EarlyConvViT [58], MLP-Mixer [59], and ResMLP [60]. Since the Transformer and Vision-MLPs are known to have a large number of parameters, the model size is also listed. The mean accuracy of the 5 × 5-fold cross-validation is selected to compare the performance under similar conditions.

TABLE 9. Reported accuracy of 5-Fold cross validation for high-performance models [28].

Model	Input data	Model size (# Params)	Mean Accuracy (%)
$\mathrm{T\text{-}GNN_{Gated}}$ (proposed)	Graph	9.7 K	75.47±2.36
fNIRS-T	Time series	3.5 M	75.49±2.07
ViT	Virtual image	2.4 M	73.17±1.66
EarlyConvViT	Virtual image	2.7 M	72.87±1.26
MLP-Mixer	Virtual image	1.4 M	74.14±1.39
ResMLP	Virtual image	1.2 M	74.36±1.90

The performance of $\mathrm{T-GNN_{Gated}}$ was comparable to the state-of-the-art models, despite its compact size. This



(a) (b) (c) (d)

(e) (f) (g) (h)

FIGURE 10. The learning curves of trained models. The top row shows the training and validation accuracy curves of (a) $\mathrm{BL_1}$ and $\mathrm{GNN_{POS}}$, (b) $\mathrm{BL_2}$, $\mathrm{BL_3}$, and $\mathrm{T-GNN_{POS}}$, (c) $\mathrm{GNN_{Gated}}$ and $\mathrm{GNN_{Avg}}$, and (d) $\mathrm{T-GNN_{Gated}}$ and $\mathrm{T-GNN_{Avg}}$. The bottom row shows the training and validation loss curves of (e) $\mathrm{BL_1}$ and $\mathrm{GNN_{POS}}$, (f) $\mathrm{BL_2}$, $\mathrm{BL_3}$, and $\mathrm{T-GNN_{POS}}$, (g) $\mathrm{GNN_{Gated}}$, and (h) $\mathrm{T-GNN_{Gated}}$.

parameter efficiency is due to the nature of GCN, where all nodes share weights and the number of layers is limited. It should be noted that there are differences in the modality used and the sampling method. For example, fNIRS-T uses Δ HbO and ΔHbR time series of 1.5-19.2 s after task onset. In this study, a Δ HbO time series of 0-15 s after task onset is used.

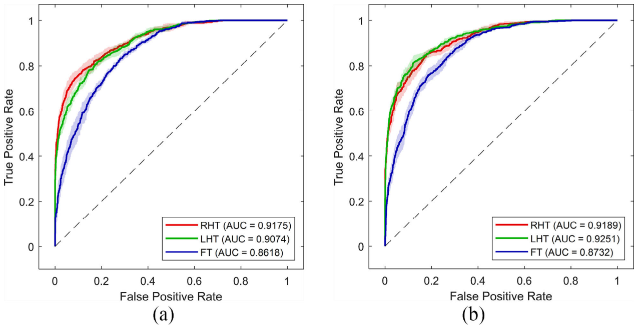
2) LEARNING CURVES

The accuracy and loss curves of the trained models are shown in Fig. 10. The curves obtained from each fold of the 5-fold cross-validation were averaged. Since the class-specific models and the position-based model produced similar results, the curves of the $\mathrm{GNN_{POS}}$ and $\mathrm{T-GNN_{POS}}$ are presented as representative. Overall, the GNN-based model showed higher validation accuracy and lower validation loss compared to the baseline models.

CNN-based models tended to converge faster, but the increase in training accuracy did not lead to better validation performance, which implies insufficient regularization. Overfitting was more evident for $\mathrm{BL_2}$ and $\mathrm{BL_3}$, as model depth increased. As training progressed, the model became overconfident leading to a rapid increase in validation loss. Although validation accuracy was not significantly affected, overconfidence can make prediction scores less informative. Since the ensemble model relies on a gating network based on prediction scores, maximum epochs of 150 and 300 were set to prevent overfitting. It can be seen that the validation loss of $\mathrm{GNN_{POS}}$ and $\mathrm{T-GNN_{POS}}$ stopped decreasing around the maximum epoch, whereas the training accuracy continued to increase. Note that the training and validation accuracy of ensemble models starts at about 70% because the base models are already trained.

3) TASK-DEPENDENT FUNCTIONAL CONNECTIVITY

Haggard et al. [61] reported that the primary motor cortex (M1) becomes active during voluntary movement. Also, the contralateral hemisphere becomes more active compared to the ipsilateral hemisphere, creating an asymmetry. This asymmetry explains the negative correlation of channel pairs (5, 16) and (6, 15) in $\boldsymbol{\rho}_{\mathrm{RHT}}$ and $\boldsymbol{\rho}_{\mathrm{LHT}}$. Considering the sensor position shown in Fig. 1, it can be deduced that M1 corresponds to the measurement channels of $\{4, 5, 6, 14, 15, 16\}$. Furthermore, the brain functional map of the primary motor cortex [62] shows that the area responsible for foot movement is closer to Cz compared to the area responsible for hand movement. This explains why no significant hemispheric asymmetries were observed in $\boldsymbol{\rho}_{\mathrm{FT}}$ since no sensors were placed around Cz. Thus, obtained task-dependent FCs agree well with previous studies on brain activation areas. In addition, if a threshold was applied, the characteristic asymmetry and connections between active channels could be lost.



(a) (b)

FIGURE 11. The ROC curves and AUC for each class of (a) $\mathrm{GNN_{Gated}}$ and (b) $\mathrm{T-GNN_{Gated}}$. The shaded area indicates the 95% confidence intervals for all folds.

TABLE 10. The architecture of the baseline model BL_1.

Layer	Kernel size	Stride	Channel	Output Size
Input	-	-	-	$7 \times 7 \times 3$
2D-convolution	[3 3]	[1 1]	64	$7 \times 7 \times 64$
2D-convolution	[3 3]	[1 1]	2	$7 \times 7 \times 2$
Fully-connected	-	-	49	1×49
Fully-connected	-	-	3	1×3

TABLE 11. The architecture of the baseline model BL_2.

Layer	Kernel size	Stride	Channel	Output Size
Input ($C \times T$)	-	-	-	20×30
1D-convolution	5	2	40	40×13
1D-convolution	3	1	40	40×11
1D-convolution	3	1	40	40×90
1D-convolution	5	1	80	80×5
1D-convolution	3	1	80	80×3
1D-convolution	3	1	80	80×1
Fully-connected	-	-	40	40×1
Fully-connected	-	-	3	3×1

4) ERROR ANALYSIS

The observation made on task-dependent FC can be extended to an error analysis of the proposed model. As mentioned above, no sensors were placed around Cz, where the neural activity signal from FT could be effectively measured. From Table 4 - 6, it can be seen that the F1-score of the FT class is the lowest in all cases, indicating that the models suffered the most from misclassification of FT. This can also be verified by the ROC curves for each class. As shown in Fig. 11., the AUC of FT was significantly lower than the AUC of RHT or LHT for $\text{GNN}_{\text{Gated}}$ and $\text{T} - \text{GNN}_{\text{Gated}}$.

Another potential source of error is the so-called ''BCI illiteracy'', which states that approximately 20% of untrained subjects cannot produce a signal reliable enough for BCI control [63]. Although the motor task is known to produce a robust signal compared to mental arithmetic or motor imagery tasks, the reported subject-dependent [44] or leave-one-subject-out [56] validation result indicates that there is a large difference in classification accuracy between subjects. These unreliable signals not only degrade the validation accuracy but also result in noisy labels, which can interfere with the training process.

TABLE 12. The architecture of the baseline model BL_3.

Layer	Kernel size	Stride	Channel	Output Size
Input	-	-	-	$8 \times 8 \times 30 \times 1$
3D-convolution	[2 2 9]	[1 1 2]	16	$8 \times 8 \times 15 \times 16$
3D-convolution	[2 2 3]	[2 2 2]	32	$4 \times 4 \times 8 \times 32$
3D-convolution	[2 2 3]	[2 2 2]	64	$2 \times 2 \times 3 \times 64$
Maxpool	[1 1 3]	[1 1 1]	-	$2 \times 2 \times 1 \times 64$
Fully-connected	-	-	64	1×64
Fully-connected	-	-	3	1×3

TABLE 13. Confusion matrices of base models without temporal module.

Model			Predicted label		
			RHT	LHT	FT
GNN_{POS}	True label	RHT	550	56	144
		LHT	58	523	169
		FT	116	118	516
GNN_{RHT}		RHT	566	61	123
		LHT	61	516	173
		FT	123	148	479
GNN_{LHT}		RHT	547	60	143
		LHT	62	544	144
		FT	146	132	472
GNN_{FT}		RHT	537	64	149
		LHT	60	508	182
		FT	104	111	535

5) LIMITATIONS

Although GNN-based models trained with the proposed class-specific graph showed strong performance as a standalone classifier and as a base model for ensemble learning, heuristic steps were involved in the graph construction. A more comprehensive investigation of the graph construction method can further improve the performance. The proposed spatial and temporal modules were designed to work separately to evaluate the GNN's ability to extract spatial features. However, in terms of model performance, using a spatio-temporal graph neural network to extract spatial and temporal features simultaneously may be helpful. Finally, the method of utilizing ΔHbR as an additional modality can be explored.

VI. CONCLUSION

In this paper, we propose a graph representation and classification method for fNIRS signals. Although functional

TABLE 14. Confusion matrices of base models with temporal module.

Model			Predicted label		
			RHT	LHT	FT
T-GNN$_{\mathrm{POS}}$	True label	RHT	529	63	158
		LHT	49	564	137
		FT	105	114	531
T-GNN$_{\mathrm{RHT}}$		RHT	570	53	127
		LHT	58	543	149
		FT	124	129	497
T-GNN$_{\mathrm{LHT}}$		RHT	531	51	168
		LHT	52	564	134
		FT	113	105	532
T-GNN$_{\mathrm{FT}}$		RHT	517	64	169
		LHT	61	530	159
		FT	99	92	559

TABLE 15. Confusion matrices of ensemble models.

Model			Predicted label		
			RHT	LHT	FT
GNN$_{\mathrm{Gated}}$	True label	RHT	576	47	127
		LHT	58	531	161
		FT	112	103	535
T-GNN$_{\mathrm{Gated}}$		RHT	556	47	147
		LHT	45	572	133
		FT	97	83	570

connectivity is known to contain valuable information, applying FC to a single-trial, multi-class BCI scheme has been challenging due to its variability across trials and classes. To address this limitation, adjacency matrices were defined for each class, rather than for each trial. The positional information and the averaged task-dependent FC were used as the connectivity measure to define a characteristic adjacency matrix. The data were converted to graphs using the obtained adjacency matrices, which were then used to train class-specific models. Finally, a fusion module was employed to weight each model for the final prediction.

It is worth noting that, unlike previous studies in the field of fMRI or connectome, task-dependent FC was not used as a feature for distinguishing structural changes in the brain network. Instead, it served more as a soft mask that guides the model to extract features relevant to the corresponding classes, thus helping the ensemble model to generate a more generalized prediction.

The proposed models were evaluated on a public dataset and significantly outperformed baseline models based on SVM and CNN. In the future, we plan to improve the model performance by exploring better graph construction schemes, integrating spatial and temporal models, and using ΔHbR as an additional modality.

APPENDIX A
ARCHITECTURE OF THE BASELINE MODELS

See Tables 10–12.

APPENDIX B
CONFUSION MATRICES OF PROPOSED MODELS

See Tables 13–15.

REFERENCES

[1] Y. Hoshi and M. Tamura, "Detection of dynamic changes in cerebral oxygenation coupled to neuronal function during mental work in man," *Neurosci. Lett.*, vol. 150, no. 1, pp. 5–8, Feb. 1993.

[2] M. Ferrari and V. Quaresima, "A brief review on the history of human functional near-infrared spectroscopy (fNIRS) development and fields of application," *NeuroImage*, vol. 63, no. 2, pp. 921–935, Nov. 2012.

[3] R. N. Aslin, "Questioning the questions that have been asked about the infant brain using near-infrared spectroscopy," *Cognit. Neuropsychol.*, vol. 29, nos. 1–2, pp. 7–33, Mar. 2012.

[4] L. H. Ernst, S. Schneider, A.-C. Ehlis, and A. J. Fallgatter, "Functional near infrared spectroscopy in psychiatry: A critical review," *J. Near Infr. Spectrosc.*, vol. 20, no. 1, pp. 93–105, Feb. 2012.

[5] E. C. Leuthardt, K. J. Miller, G. Schalk, R. P. N. Rao, and J. G. Ojemann, "Electrocorticography-based brain computer interface—The Seattle experience," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 14, no. 2, pp. 194–198, Jun. 2006.

[6] D. L. Bailey, M. N. Maisey, D. W. Townsend, and P. E. Valk, *Positron Emission Tomography*, vol. 2. Cham, Switzerland: Springer, 2005.

[7] M. Hämäläinen, R. Hari, R. J. Ilmoniemi, J. Knuutila, and O. V. Lounasmaa, "Magnetoencephalography—Theory, instrumentation, and applications to noninvasive studies of the working human brain," *Rev. Mod. Phys.*, vol. 65, no. 2, pp. 413–497, Apr. 1993.

[8] S. Ogawa, D. W. Tank, R. Menon, J. M. Ellermann, S. G. Kim, H. Merkle, and K. Ugurbil, "Intrinsic signal changes accompanying sensory stimulation: Functional brain mapping with magnetic resonance imaging," *Proc. Nat. Acad. Sci. USA*, vol. 89, no. 13, pp. 5951–5955, Jul. 1992.

[9] E. Niedermeyer and F. H. L. da Silva, *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. Philadelphia, PA, USA: Lippincott Williams & Wilkins, 2005.

[10] J. R. Wolpaw, "Brain–computer interfaces (BCIs) for communication and control," in *Proc. 9th Int. ACM SIGACCESS Conf. Comput. Accessibility*, Oct. 2007, pp. 1–2.

[11] J. J. Vidal, "Toward direct brain–computer communication," *Annu. Rev. Biophys. Bioeng.*, vol. 2, no. 1, pp. 157–180, Jun. 1973.

[12] N. Birbaumer, N. Ghanayim, T. Hinterberger, I. Iversen, B. Kotchoubey, A. Kübler, J. Perelmouter, E. Taub, and H. Flor, "A spelling device for the paralysed," *Nature*, vol. 398, no. 6725, pp. 297–298, Mar. 1999.

[13] G. R. McMillan, G. Calhoun, M. S. Middendorf, J. H. Schnurer, D. F. Ingle, and V. T. Nasman, "Direct brain interface utilizing self-regulation of steady-state visual evoked response (SSVER)," in *Proc. RESNA*, Vancouver, BC, Canada, 1995, pp. 693–695.

[14] K. LaFleur, K. Cassady, A. Doud, K. Shades, E. Rogin, and B. He, "Quadcopter control in three-dimensional space using a noninvasive motor imagery-based brain-computer interface," *J. Neural Eng.*, vol. 10, no. 4, Aug. 2013, Art. no. 046003.

[15] R. Spataro, R. Sorbello, S. Tramonte, G. Tumminello, M. Giardina, A. Chella, and V. La Bella, "Reaching and grasping a glass of water by locked-in ALS patients through a BCI-controlled humanoid robot," *J. Neurolog. Sci.*, vol. 357, pp. e48–e49, Oct. 2015.

[16] L. A. Farwell, D. C. Richardson, G. M. Richardson, and J. J. Furedy, "Brain fingerprinting classification concealed information test detects U.S. navy military medical information with P300," *Frontiers Neurosci.*, vol. 8, p. 410, Dec. 2014.

[17] C.-T. Lin, Y.-C. Chen, T.-Y. Huang, T.-T. Chiu, L.-W. Ko, S.-F. Liang, H.-Y. Hsieh, S.-H. Hsu, and J.-R. Duann, "Development of wireless brain computer interface with embedded multitask scheduling and its application on real-time driver's drowsiness detection and warning," *IEEE Trans. Biomed. Eng.*, vol. 55, no. 5, pp. 1582–1591, May 2008.

[18] D. Rosenboom, "Active imaginative listening—A neuromusical critique," *Frontiers Neurosci.*, vol. 8, p. 251, Aug. 2014.

[19] A. Nijholt and M. Poel, "Multi-brain BCI: Characteristics and social interactions," in *Proc. Int. Conf. Augmented Cognition*, Toronto, ON, Canada, Jul. 2016, pp. 79–90.

[20] N. Naseer and K.-S. Hong, "FNIRS-based brain–computer interfaces: A review," *Frontiers Hum. Neurosci.*, vol. 9, p. 3, Jan. 2015.

[21] K.-S. Hong, M. J. Khan, and M. J. Hong, "Feature extraction and classification methods for hybrid fNIRS-EEG brain–computer interfaces," *Frontiers Hum. Neurosci.*, vol. 12, p. 246, Jun. 2018.

[22] C. Eastmond, A. Subedi, S. De, and X. Intes, "Deep learning in fNIRS: A review," *Neurophotonics*, vol. 9, no. 4, Jul. 2022, Art. no. 041411.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[24] J. Kwon and C.-H. Im, "Subject-independent functional near-infrared spectroscopy-based brain–computer interfaces based on convolutional neural networks," *Frontiers Hum. Neurosci.*, vol. 15, Mar. 2021, Art. no. 646915.

[25] T. Trakoolwilaiwan, B. Behboodi, J. Lee, K. Kim, and J.-W. Choi, "Convolutional neural network for high-accuracy functional near-infrared spectroscopy in a brain–computer interface: Three-class classification of rest, right-, and left-hand motor execution," *Neurophotonics*, vol. 5, no. 1, Sep. 2018, Art. no. 011008.

[26] Y. Gao, P. Yan, U. Kruger, L. Cavuoto, S. Schwaitzberg, S. De, and X. Intes, "Functional brain imaging reliably predicts bimanual motor skill performance in a standardized surgical task," *IEEE Trans. Biomed. Eng.*, vol. 68, no. 7, pp. 2058–2066, Jul. 2021.

[27] S. D. Wickramaratne and M. S. Mahmud, "A deep learning based ternary task classification system using Gramian angular summation field in fNIRS neuroimaging data," in *Proc. IEEE Int. Conf. E-Health Netw., Appl. Services (HEALTHCOM)*, Mar. 2021, pp. 1–4.

[28] Z. Wang, J. Zhang, Y. Xia, P. Chen, and B. Wang, "A general and scalable vision framework for functional near-infrared spectroscopy classification," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 30, pp. 1982–1991, 2022.

[29] D. Yang, R. Huang, S.-H. Yoo, M.-J. Shin, J. A. Yoon, Y.-I. Shin, and K.-S. Hong, "Detection of mild cognitive impairment using convolutional neural network: Temporal-feature maps of functional near-infrared spectroscopy," *Frontiers Aging Neurosci.*, vol. 12, p. 141, May 2020.

[30] M. Saadati, J. Nelson, and H. Ayaz, "Mental workload classification from spatial representation of FNIRS recordings using convolutional neural networks," in *Proc. IEEE 29th Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Oct. 2019, pp. 1–6.

[31] Y. Kwak, W.-J. Song, and S.-E. Kim, "FGANet: fNIRS-guided attention network for hybrid EEG-fNIRS brain–computer interfaces," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 30, pp. 329–339, 2022.

[32] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017.

[33] O. Sporns, "The human connectome: A complex network," *Schizophrenia Res.*, vol. 136, p. S28, Apr. 2012.

[34] F. V. Farahani, W. Karwowski, and N. R. Lighthall, "Application of graph theory for identifying connectivity patterns in human brain networks: A systematic review," *Frontiers Neurosci.*, vol. 13, p. 585, Jun. 2019.

[35] M. P. van den Heuvel and H. E. H. Pol, "Exploring the brain network: A review on resting-state fMRI functional connectivity," *Eur. Neuropsychopharmacol.*, vol. 20, no. 8, pp. 519–534, Aug. 2010.

[36] X. Di, Z. Fu, S. C. Chan, Y. S. Hung, B. B. Biswal, and Z. Zhang, "Task-related functional connectivity dynamics in a block-designed visual experiment," *Frontiers Human Neurosci.*, vol. 9, pp. 1–10, Sep. 2015.

[37] X. Li, Y. Zhou, N. Dvornek, M. Zhang, S. Gao, J. Zhuang, D. Scheinost, L. H. Staib, P. Ventola, and J. S. Duncan, "BrainGNN: Interpretable brain graph neural network for fMRI analysis," *Med. Image Anal.*, vol. 74, Dec. 2021, Art. no. 102233.

[38] A. Demir, T. Koike-Akino, Y. Wang, M. Haruna, and D. Erdogmus, "EEG-GNN: Graph neural networks for classification of electroencephalogram (EEG) signals," in *Proc. 43rd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Nov. 2021, pp. 1061–1067.

[39] P. Zhong, D. Wang, and C. Miao, "EEG-based emotion recognition using regularized graph neural networks," *IEEE Trans. Affect. Comput.*, vol. 13, no. 3, pp. 1290–1301, Jul. 2022.

[40] Q. Yu, R. Wang, J. Liu, L. Hu, M. Chen, and Z. Liu, "GNN-based depression recognition using spatio-temporal information: A fNIRS study," *IEEE J. Biomed. Health Informat.*, vol. 26, no. 10, pp. 4925–4935, Oct. 2022.

[41] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[42] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Comput.*, vol. 6, no. 2, pp. 181–214, Mar. 1994.

[43] D. H. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, Jan. 1992.

[44] S. Bak, J. Park, J. Shin, and J. Jeong, "Open-access fNIRS dataset for classification of unilateral Finger- and foot-tapping," *Electronics*, vol. 8, no. 12, p. 1486, Dec. 2019.

[45] K. J. Friston, C. D. Frith, P. F. Liddle, and R. S. J. Frackowiak, "Functional connectivity: The principal-component analysis of large (PET) data sets," *J. Cerebral Blood Flow Metabolism*, vol. 13, no. 1, pp. 5–14, Jan. 1993.

[46] A. W. Toga and P. M. Thompson, "Mapping brain asymmetry," *Nature Rev. Neurosci.*, vol. 4, no. 1, pp. 37–48, Jan. 2003.

[47] Ü. Sakoğlu, G. D. Pearlson, K. A. Kiehl, Y. M. Wang, A. M. Michael, and V. D. Calhoun, "A method for evaluating dynamic functional network connectivity and task-modulation: Application to schizophrenia," *Magn. Reson. Mater. Phys., Biol. Med.*, vol. 23, nos. 5–6, pp. 351–366, Feb. 2010.

[48] K. Oono and T. Suzuki, "Graph neural networks exponentially lose expressive power for node classification," 2019, *arXiv:1905.10947*.

[49] S. Achard and E. Bullmore, "Efficiency and cost of economical brain functional networks," *PLoS Comput. Biol.*, vol. 3, no. 2, p. e17, Feb. 2007.

[50] K. J. Friston, A. P. Holmes, K. J. Worsley, J. Poline, C. D. Frith, and R. S. J. Frackowiak, "Statistical parametric maps in functional imaging: A general linear approach," *Hum. Brain Mapping*, vol. 2, no. 4, pp. 189–210, Jan. 1994.

[51] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[52] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proc. AAAI*, Apr. 2018, vol. 32, no. 1, pp. 1–13.

[53] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," 2017, *arXiv:1701.06538*.

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

[55] Z. Sun, Z. Huang, F. Duan, and Y. Liu, "A novel multimodal approach for hybrid brain–computer interface," *IEEE Access*, vol. 8, pp. 89909–89918, 2020.

[56] Z. Wang, J. Zhang, X. Zhang, P. Chen, and B. Wang, "Transformer model for functional near-infrared spectroscopy classification," *IEEE J. Biomed. Health Informat.*, vol. 26, no. 6, pp. 2559–2569, Jun. 2022.

[57] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[58] T. Xiao, P. Dollar, M. Singh, E. Mintun, T. Darrell, and R. Girshick, "Early convolutions help transformers see better," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 30392–30400.

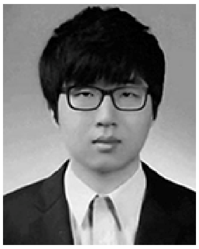
[59] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, M. Lucic, and A. Dosovitskiy, "MLP-Mixer: An all-MLP architecture for vision," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 24261–24272.

[60] H. Touvron, P. Bojanowski, M. Caron, M. Cord, A. El-Nouby, E. Grave, G. Izacard, A. Joulin, G. Synnaeve, J. Verbeek, and H. Jégou, "ResMLP: Feedforward networks for image classification with data-efficient training," 2021, *arXiv:2105.03404*.

[61] P. Haggard, "Human volition: Towards a neuroscience of will," *Nature Rev. Neurosci.*, vol. 9, no. 12, pp. 934–946, Dec. 2008.

[62] C. Weiss, C. Nettekoven, A. K. Rehme, V. Neuschmelting, A. Eisenbeis, R. Goldbrunner, and C. Grefkes, "Mapping the hand, foot and face representations in the primary motor cortex—Retest reliability of neuronavigated TMS versus functional MRI," *NeuroImage*, vol. 66, pp. 531–542, Feb. 2013.

[63] B. Z. Allison and C. Neuper, "Could anyone use a BCI?" in *Brain–Computer Interfaces: Applying our Minds to Human–Computer Interaction*, D. S. Tan and A. Nijholt, Eds. London, U.K.: Springer, 2010, pp. 35–54.



MINSEOK SEO received the B.S. and M.S. degrees in mechanical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree in mechanical engineering. His research interests include the processing and classification of brain activity signals using deep learning approaches.

EUGENE JEONG received the B.S. and M.S. degrees in mechanical engineering from KAIST, Daejeon, South Korea, in 2014 and 2017, respectively, where he is currently pursuing the Ph.D. degree in mechanical engineering. His research interests include the development of fNIRS and BCI systems for BCI and intention estimation from the acquired signals.

KYUNG-SOO KIM (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in mechanical engineering from KAIST, Daejeon, South Korea, in 1993, 1995, and 1999, respectively. He was a Chief Researcher with LG Electronics Inc., from 1999 to 2003, and the DVD Front-End Manager of STMicroelectronics Company Ltd., from 2003 to 2005. In 2005, he joined the Department of Mechanical Engineering, Korea Polytechnic University, Gyeonggi-do, South Korea, as a Faculty Member. Since 2007, he has been with the Department of Mechanical Engineering, KAIST. His research interests include control theory, sensor and actuator design, and robot manipulator design. He serves as an Associate Editor for *Automatica* and the *Journal of Mechanical Science and Technology*.