Subject-Independent Motor Imagery EEG Classification Based on Graph Convolutional Network

Juho Lee, Jin Woo Choi and Sungho Jo

November 10, 2021

Subject-Independent Motor Imagery EEG Classification Based on Graph Convolutional Network

Juho Lee, Jin Woo Choi, and Sungho Jo(✉)

School of Computing, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34141, South Korea
{leejh1021, rayoakmont, shjo}@kaist.ac.kr

Abstract. Electroencephalogram (EEG) motor imagery (MI) has attracted much attention in brain-computer interfaces (BCIs) as it directly encodes human intentions. However, the variability of EEG-based brain signals between individuals requires current BCI systems to undergo calibration procedures before its usage. In this paper, we propose a model that targets minimizing such procedures by improving inter-subject classification performance. The purpose of our proposed method is to extract features using previously studied convolution-based deep learning structures while utilizing a graph structure to analyze inter-subject relationships with multiple subjects. By utilizing not only features but also the relationship between subject-specific features, it becomes possible to make predictions focusing on subjects with high similarity. Therefore, even new users not seen during the training process are predicted relatively efficiently. To validate our method, we evaluated our model with the public dataset BCI Competition IV IIa. The results in our study suggest that our proposed method improved the cross-subject classification accuracy by combining it with the previous deep learning model and induced a balanced prediction for the classes. Our study has shown the potential to develop MI-based BCI applications that do not require user calibration by training the model with pre-existing datasets.

Keywords: Brain-computer interfaces (BCIs) · graph convolutional network (GCN) · deep learning (DL) · electroencephalography (EEG) · motor imagery (MI) · subject-independent

1 Introduction

Brain-computer interface (BCI) systems provide a communication path that can interact with external devices by classifying the user's brain neural activity patterns [1–3]. One of the non-invasive ways to record brain activity is through electroencephalogram (EEG), which utilizes multiple electrodes placed on a specific scalp area. EEG signals have been widely used in BCI applications as they can record brain activity relatively easily and inexpensively compared to other neural acquisition techniques [4–8].

So far, EEG-based BCI applications have been developed using neurophysiological patterns, including steady-state visual evoked potential (SSVEP), event-related potential (ERP), movement-related cortical potentials (MRCPs), and motor imagery (MI) [9–12]. Among these BCI studies, MI, which classifies EEG signals based on the imagination of body movement, has recently attracted much attention. MI EEG signal can be elicited in a specific pattern when a subject imagines performing a specific movement, such as moving of hand or foot. Several studies have researched MI-based EEG patterns in advance, as it is a signal that both healthy and disabled people can control without external stimulation.

Recently, with the success of machine learning and deep learning in computer vision [13] and speech recognition [14], many studies have been conducted to apply them to the EEG classification task. For instance, Ang et al. and Pfurtscheller et al. applied machine learning techniques using power spectral density (PSD) measures from MI EEG to classify participants' intentions [15, 16]. However, machine learning has a limitation in that it relies on the handcrafted features of human experts. Therefore, to overcome the limitations of machine learning, deep learning-based methods have been proposed, and significant improvements have been made. In particular, convolutional neural network (CNN) has been widely applied to MI classification tasks because they effectively extract temporal and spatial features from EEG signals [17, 18].

In general, human EEG signals change according to the difference in each individual's mental state, resulting in large variability on inter-subject relations [19, 20]. Existing machine learning and deep learning methods have been successful in MI classification, but many have poor performance when there is insufficient data on new users. Therefore, most MI task methods, including the studies mentioned above, require the calibration process to be performed using sufficient data from new users. However, the calibration process is time-consuming and inconvenient [21]. This is a significant obstacle in the practical application of the BCI system, and research to eliminate or minimize the calibration process is necessary.

In this paper, we focus on improving subject-independent classification accuracy to take a step closer to BCI systems that do not require calibration from the user. Thus, the proposed model in our study performs the relation-based prediction by defining the relationship between multiple subjects through a graph structure. This approach enables efficient prediction even for new users by focusing on subjects with high similarity. The main contributions are as follows.

- In order to improve the inter-subject classification performance, we developed a method applicable to the existing deep learning-based MI classification model.
- We applied the proposed method to the previously studied MI classification model and analyzed the performance with and without our method.

2 Related work

Most of the existing methods for the MI EEG task have been developed based on the intra-subject situation requiring calibration time [22, 23]. In previous studies, researchers have employed techniques based on machine learning using handcrafted features. As one of several methods, common spatial pattern (CSP) is a method for maximizing the variance difference between different classes [24, 25]. In particular, filter bank CSP (FBCSP), an algorithm based on CSP method, is one of the widely used methods in MI-based BCI [26]. These features are fed to classifiers such as support vector machine (SVM) [27] and Random Forest [28] to generate predictions.

Existing machine learning methods classified EEG MI data with good performance. However, such methods have limitations on achieving higher accuracy as it relies heavily on handcrafted features created by human experts. Deep learning methods such as CNN overcome these limitations through automated feature extraction. For example, Schirrmeister et al. [17] outperformed the existing FBCSP [26] algorithm by using CNN.

MI classification methods mentioned above are intra-subject-based methods that require a calibration process. However, the calibration process is time-consuming and inconvenient to acquire data [21]. Recently, several methods have been proposed to reduce the calibration time in the MI-based BCI system. For example, Tang et al. [29] suggested a method using the adversarial domain adaptation technology and reduced the inconvenience of the calibration process by using only unlabeled data of new users. An et al. [30] applied the attention module to the few-shot relation network structure [31] and reduced the calibration time by using only a few seconds of data from new users.

Wang et al. [32] tried to solve the lack of data in the target domain in computer vision by utilizing multi-source domain adaptation. We applied this study to improve inter-subject classification performance in BCI applications. We defined the relationship between each EEG feature of subjects as a graph structure and utilized the relationship information for prediction using a graph convolution network [33]. In addition, it was implemented as a model suitable for classifying EEG signals by designing a structure to extract temporal and spatial characteristics of EEG signals effectively.

3 Method

3.1 Definition and notation

Before a detailed description of the proposed model, we first describe the definitions and notations.

Data of a single subject is defined with $\{(X_i, y_i), i = 1, 2, ..., l\}$, where l denotes the number of motor imagery EEG trials, $X_i \in \mathbb{R}^{E \times T}$ denotes EEG signals from a single trial with dimensions E, the number of electrodes, by T, the number of time points sampled, y corresponds to a label defined by $y \in \{1, 2, ..., C\}$, and C is the number of classes.

3.2 Network architecture

Fig. 1 shows the overall architecture of the proposed model. The description of the architecture is divided into Feature extraction, Prototype update, Graph construction, Prediction, and Model inference. In the feature extraction part, the raw EEG signals of subjects from S_1 to S_n are passed through the feature extractor and are converted into fixed-dimensional feature embeddings. The extracted features are used in two different ways: as updating elements for prototypes representing features per class for each subject, and as queries used for classification. While prototypes are used to learn similarities between classes and subjects using graph structure, the feature embeddings are also concatenated as a query sample and are predicted through the graph convolutional network to classify into labels. As the query is predicted by considering the relationships between all subjects and classes, subject-independent prediction is taken in consideration within the model.

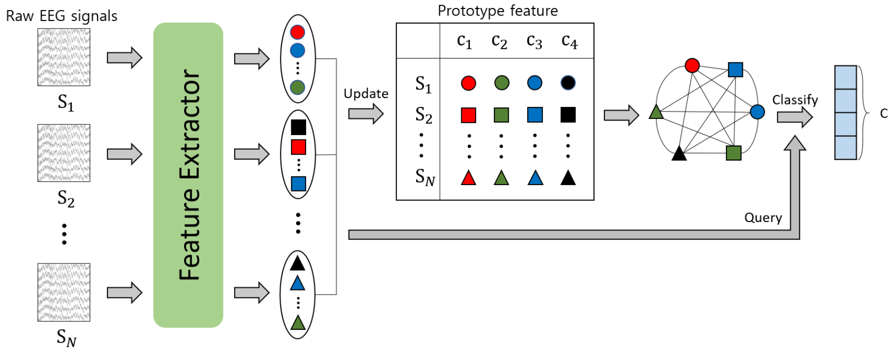

Fig. 1. An overview of the proposed model. From left to right, raw EEG signals, feature extractor, extracted features, prototype features, prototype graph, and prediction probability are shown. $S_1, S_2, \ldots, S_n$ denote subjects, and c_1, c_2, c_3, c_4 denote classes.

Feature extraction The raw EEG signal is first band-pass filtered from 4 to 38 Hz, the range of motor imagery frequency, and then cropped to fixed window size. A feature extractor $f(\cdot)$ serves to convert EEG signals into features and is designed by modifying the representative MI classifier model. The feature extractor can be flexibly replaced, and in this paper, we designed the feature extractor by modifying the architectures of Shallownet [17], and EEGnet [18], which were used as comparison groups. In the case of Shallownet, features are extracted using temporal and spatial convolution and square and logarithm activation functions. EEGnet extract features using depth-wise separable convolution and Elu activation function. The detailed structure is in Table 1 and Table 2.

Prototype update For every mini-batch, sampled EEG feature embeddings are used to estimate prototypes representing each subject and class label. The estimated prototype $\hat{p}_{S_n,c}$ is measured as the average value of the corresponding features (Subject S_n, class c) for each mini-batch. The prototype features $p_{S_n,c}$ are updated while accommodating some of the estimated prototypes as follows:

$$p_{S_n,c} = \beta p_{S_n,c} + (1-\beta)\hat{p}_{S_n,c} \qquad n = 1,2,...,N \tag{1}$$

β is an exponential decay rate that determines the update rate, and N is the number of train subjects.

Table 1. Details of the Shallownet feature extractor. Parameters contain the kernel size, the number of the feature map, and the type of layer.

Layers	Output size	Parameters
Input	$1 \times 22 \times 1000$	-
Temporal conv	$40 \times 22 \times 976$	(1×22), 40, conv
Spatial conv	$40 \times 1 \times 976$	(22×1), 40, conv
Batch norm	$40 \times 1 \times 976$	-
Square activation	$40 \times 1 \times 976$	-
Average pooling	$40 \times 1 \times 61$	(1×75), stride 15, avg pool
Logarithm activation	$40 \times 1 \times 61$	-
Dropout	$40 \times 1 \times 61$	rate = 0.5
Global average pooling	$40 \times 1 \times 1$	(1×61), avg pool

Table 2. Details of the EEGnet feature extractor. Parameters contain the kernel size, the number of the feature map, and the type of layer.

Layers	Output size	Parameters
Input	$1 \times 22 \times 1000$	-
Conv	$8 \times 22 \times 1000$	(1×125), 8, conv, pad=same
Batch norm	$8 \times 22 \times 1000$	-
Depth-wise conv	$16 \times 1 \times 1000$	(22×1), 16, depth-wise conv
Batch norm	$16 \times 1 \times 1000$	-
Elu activation	$16 \times 1 \times 1000$	-
Max pooling	$16 \times 1 \times 250$	(1×4), max pool
Dropout	$16 \times 1 \times 250$	rate = 0.5
Separable conv	$16 \times 1 \times 250$	(1×16), 16, separable conv, pad=same
Batch norm	$16 \times 1 \times 250$	-
Max pooling	$16 \times 1 \times 31$	(1×8), max pool
Dropout	$16 \times 1 \times 31$	rate = 0.5
Flatten	496	-

Graph construction One of our important purposes is to make good predictions on a sampled query sample. In order to use the relationship between each subject and class for query sample prediction, prototype features were utilized to construct a graph G_p. To be specific, the graph is composed of a prototype feature matrix $M \in \mathbb{R}^{NC \times d}$ and a prototype adjacency matrix $A \in \mathbb{R}^{NC \times NC}$, where C is the number of classes and d is the size of a single feature. Each feature of the feature matrix constitutes as a vertex of the graph, and the adjacency matrix represents all edges between the vertices. The prototype feature matrix M is produced by concatenating the prototype features $p_{S_n,c}$ calculated in Eq. 1.

The prototype adjacency matrix represents the relationship between each vertex. As the distance between vertices increases, the element value of A should decrease. Specifically, all elements of A are computed as Gaussian kernel values of two pairs of vertex features:

$$A_{i,j} = exp\left(-\frac{\left\|M_i - M_j\right\|_2^2}{2\sigma^2}\right) \tag{2}$$

σ is a parameter that controls the sparsity of A and is set to 0.005 in this paper.

Prediction In this step, predictions are performed on the query samples and prototype features for each batch B. To begin with, an expanded graph $\bar{G}$ is constructed by adding query samples to the prototype graph G_p. The expanded feature matrix $\bar{M} \in \mathbb{R}^{(NC+|B|)\times d}$ of $\bar{G}$ is created by concatenating the features of query samples into the prototype feature matrix of G_p.

To obtain the expanded adjacency matrix $\bar{A} \in \mathbb{R}^{(NC+|B|)\times(NC+|B|)}$ of the $\bar{G}$, adjacency matrix A' between the prototypes and query samples is measured. A' is computed using a Gaussian kernel as before when calculating A:

$$A'_{i,j} = exp\left(-\frac{\left\|M_i - f(q_j)\right\|_2^2}{2\sigma^2}\right) \tag{3}$$

Finally, the $\bar{A}$ is calculated by connecting A and A' with the identity matrix I. The reason for using I is that the similarity between a query and another query is uncertain. $\bar{A}$ is obtained as:

$$\bar{A} = \begin{bmatrix} A & A' \\ A'^T & I \end{bmatrix} \tag{4}$$

After the expanded graph $\bar{G}$ is completed, it is fed to graph convolutional network to generate predicted values. Specifically, the graph convolutional network receives the expanded feature matrix and expanded adjacency matrix of $\bar{G}$ and outputs the classification prediction probability. The architecture of a graph convolutional network is shown in Table 3.

Table 3. Details of the graph convolutional network. Parameters contain the output feature size and the type of layer.

Layers	Type of input	Output size	Parameters
Input	feature, adj matrix	$(NC+\|B\|)\times d$, $(NC+\|B\|)\times(NC+\|B\|)$	-
Graph conv	feature, adj matrix	$(NC+\|B\|)\times d$	d, GCN
Relu activation	gcn feature	$(NC+\|B\|)\times d$	-
Dropout	gcn feature	$(NC+\|B\|)\times d$	rate = 0.5
Classifier	gcn feature, adj matrix	$(NC+\|B\|)\times C$	C, GCN

Model inference During the testing process, no more prototype updates are performed, and only the feature extraction and query sample prediction phase are used. Specifically, after the training process is over, the feature extractor, the graph convolutional network, the prototype feature matrix M, and the prototype adjacency matrix A are stored prior to testing. As a testing procedure, the Prediction phase is performed using the test data as query data. The process includes creating an expanded feature matrix and an expanded adjacency matrix as described in the Prediction phase, and going through graph convolutional network to output the classification probability of test samples.

3.3 Model optimization

During training, the proposed model is optimized by four losses. The total loss function can be expressed as:

$$L = \lambda_1 L_{global} + \lambda_2 L_{local} + L_{proto} + L_{query} \tag{5}$$

where λ_1 and λ_2 are trade-off parameters between L_{global} and L_{local}.

L_{global} serves to reinforce the invariance of features, and the higher the λ_1, the stronger the influence of L_{global}. L_{local} plays a role in improving the feature compactness of each class, and the higher the λ_2, the stronger the influence of L_{local}. L_{proto} and L_{query} are classification losses that allow features to be distinguished by category.

Global relation alignment loss In the training process, the relative position of feature embeddings should be constrained, excluding the effort to extract subject-independent features. This loss aims to align the features of subjects at the global level. By setting the similarity of all element pair combinations of the adjacency matrix A to loss, the similarity between all subjects and classes is maintained consistently.

$$L_{global} = \frac{1}{N^4} \sum_{i,j,m,n=1}^{N} \left\| A_{i,j} - A_{m,n} \right\|_F \tag{6}$$

where $|\cdot|_F$ denotes the Frobenius norm.

Local relation alignment loss The L_{local} aims to align the features of subjects at the local level. It is calculated as the average of the $l2$ distance between all query samples and their corresponding prototype features. By minimizing the $l2$ distance, the compressibility of the feature is improved.

Prototype classification loss The L_{proto} is a loss to increase the distinguishability of prototype features. A cross-entropy loss was employed to calculate the difference between the label and the predicted probability. L_{proto} is calculated as the average of the cross-entropy loss of the true value and the predicted probability in the prototype features.

Query classification loss The L_{query} is a loss to increase the distinguishability of query features. Like L_{proto}, it is calculated as the average of the cross-entropy loss of the true value and the predicted probability in the query samples.

4 Experiments and Results

4.1 Datasets

To evaluate the proposed method, one of the public datasets of the MI task, BCI Competition IV IIa [34], was used. The dataset contains nine subjects, where each subject performed imagination of four different body movements (left hand, right hand, both feet, and tongue). The EEG signal was measured using a 22 Ag/AgCl electrode with a sampling frequency of 250 Hz. The dataset was divided into two sessions, and each session was measured on a different day. A single session included 72 trials per each MI task. Overall, there are a total of 288 trials of EEG data per session.

4.2 Experiment settings

To demonstrate the advantages of the proposed model, Shallownet [17], and EEGnet [18] which are two popular models for MI classification, were used. Shallownet is a CNN-based deep learning model designed to extract discriminative oscillatory EEG features. EEGnet is a compressed CNN framework designed to be applied to various BCI paradigms, including MI. For a fair comparison, both models were implemented with the optimal set of hyperparameters recommended in the original paper, and conditions such as data preprocessing process and GPU device were set the same.

The experiment is performed with a leave-one(subject)-out cross-validation (LOSO-CV) protocol. For example in [34], when the test set is the data of $subject_1$, the train and validation set consists of the remaining subjects ($subject_2$ to $subject_9$) excluding $subject_1$.

The proposed method was implemented using the Pytorch framework, and the training process was implemented using the NVIDIA GeForce GTX 1080 Ti with 11GB memory. The loss function was optimized using the Adam optimizer for Shallownet and EEGnet with learning rates of 2×10^{-4}, 1×10^{-3} respectively. Batch size was set to 32, λ_1 was set to 20, λ_2 was set to 0.01. The EEG data was cropped with 4 s sliding window, 0.5 s stride before being fed to the model. Moreover, the early stopping technique [35] was used to prevent overfitting. So, if there was more than the minimum epoch and the validation loss did not decrease for 20 epochs, training was terminated early. The minimum epochs of Shallownet and EEGnet were set to 200 and 800, respectively.

4.3 Experimental Results

Classification performance For Shallownet and EEGnet, we evaluated the version with and without our method. Accuracy and F1-score were used to evaluate the performance of the All considered algorithm. We reported the experimental results of ‘Shallownet without our method’ and ‘Shallownet with our method’ in Table 4, and the experimental results of ‘EEGnet without our method’ and ‘EEGnet with our method’ were reported in Table 5. As seen on the tables, performance improvements were observed in both models when using along with our method.

Table 4. Comparison of classification performance (average accuracy ± standard deviation and average F1-score ± standard deviation) in % with and without our method in Shallownet

Method	Avg accuracy ± Std	Avg F1-score ± Std
Shallownet	42.86 ± 12.24	37.96 ± 15.13
Shallownet w/ Ours	**44.33 ± 15.07**	**39.95 ± 17.34**

Table 5. Comparison of classification performance (average accuracy ± standard deviation and average F1-score ± standard deviation) in % with and without our method in EEGnet

Method	Avg accuracy ± Std	Avg F1-score ± Std
EEGnet	42.59 ± 16.30	38.61 ± 17.42
EEGnet w/ Ours	**44.40 ± 16.02**	**40.83 ± 18.82**

Fig. 2 shows the accuracy with and without our model as a scatter plot. The diagonal dash line corresponding to $y = x$ indicates a boundary line with the same accuracy regardless of applying our model. Based on the diagonal dash line, the upper dots correspond to the subject with improved performance by combining our model, and the lower dots correspond to the subject with reduced performance. As seen on the figure, accuracy was enhanced from five and six subjects out of nine total for Shallownet and EEGnet, respectively, when using along with our method.

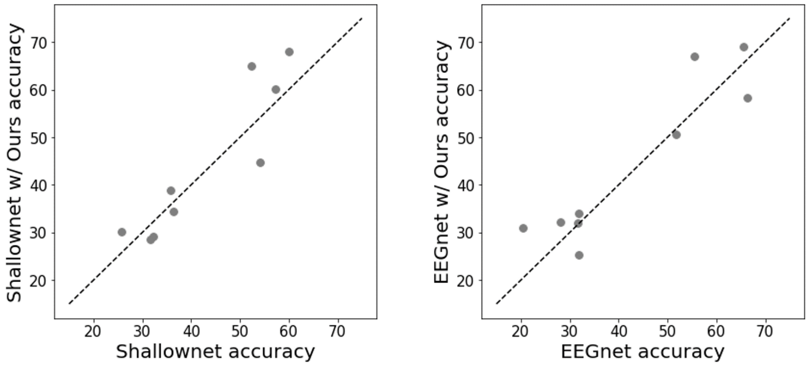

Fig. 2. Scatter plots for the baseline model and the model combining our method with the baseline. The left plot is the comparison result when combined with Shallownet, and the right plot is the comparison result when combined with EEGnet.

Efficiency analysis Fig. 3 shows the confusion matrix with and without our model using Shallownet and EEGnet. The values on the diagonal of the confusion matrix are correctly predicted samples in the MI classification task. As seen from the figure, the diagonal values of the confusion matrix were balanced when our method was applied to the model, indicating the balance of classification accuracy between classes. As for Shallownet, relatively great improvements were observed for accurately classifying feet class. And in EEGnet, by using our model, the prediction biased to the feet class was changed in a balanced way, and the accuracy related to the right hand and tongue, the lowest two classes, was improved.

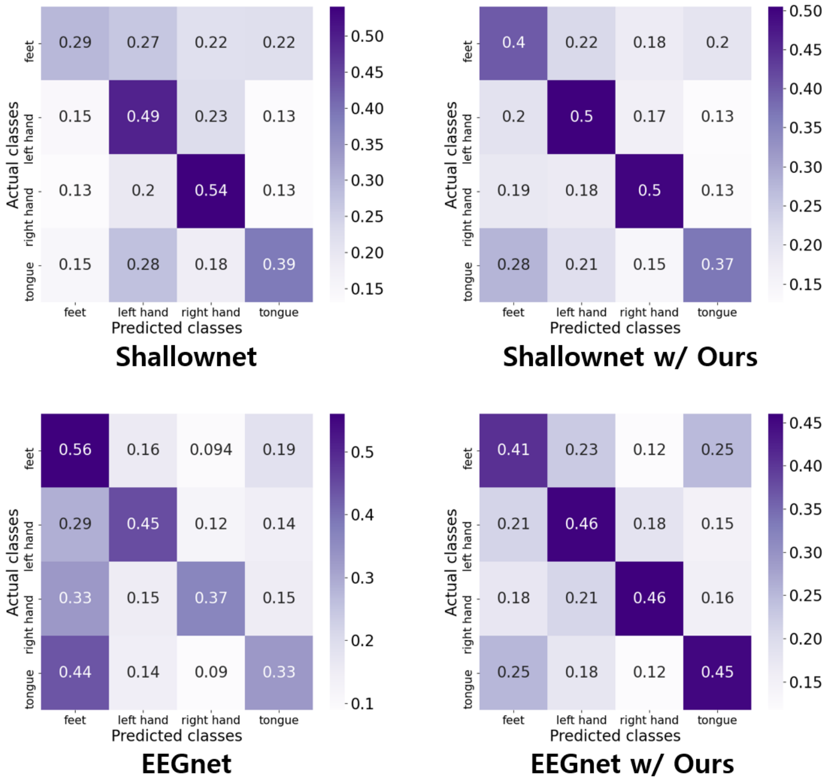

Fig. 3. Confusion matrices for the baseline model and the model combining our method with the baseline. The plots on the left are the results for the baseline model, and the plots on the right are the results for the model combining our method with the baseline. And the upper plots are the results for Shallownet, and the below plots are the results for EEGnet.

Feature visualization To further investigate how balancing accuracy between different classes were resulted, we employed t-SNE embeddings [36] to visualize the distribution of extracted features. As shown in Fig. 4, the distributions by class became clear in Shallownet and EEGnet when using along with our method for $subject_1$ and $subject_8$, respectively.

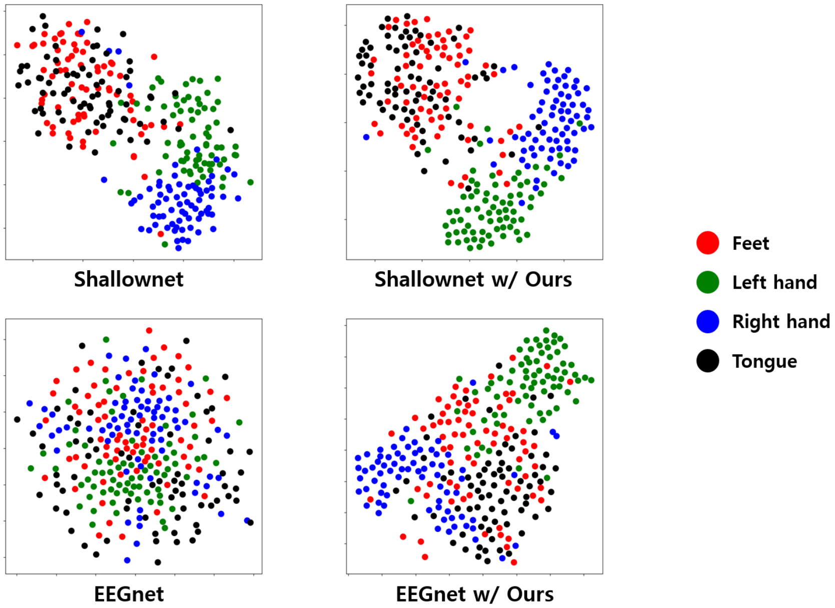

Fig. 4. Examples of t-SNE visualization for the baseline model and the model combining our method with the baseline. The upper plots are the results of $subject_1$ for Shallownet, and the plots below are the results of $subject_8$ for EEGnet.

5 Discussion and Conclusion

This paper proposed a framework that can improve the prediction probability for new users not seen in the training process. The proposed method was applied to the existing deep learning models for MI classification to improve subject-independent classification accuracy. Thus, we used relationship information between multiple subjects for training and prediction through a graph convolutional network. The public dataset BCI Competition IV IIa was used to validate the inter-subject classification performance of our method. As a result, the average classification performance was enhanced by applying our method to two types of MI classification deep learning models.

Most previous deep learning-based studies have only used features of EEG signals for training. However, we performed efficient prediction on new users while utilizing not only EEG features but also relationship information between subject-specific features for learning. The relationship between subjects was defined using the graph structure, and this relationship was used for training and prediction through a graph convolutional network.

The effectiveness of our method is clearly shown through the confusion matrixes and t-SNE plots. The existing deep learning-based MI classification model used in the experiments showed biased results in inter-subject classification. For example, Shallownet output biased results to all classes except feet class, whereas EEGnet output biased results to feet class. However, after applying our method, all classes were predicted in a balanced way. The t-SNE plots are also relatively well aligned for each class after applying our method.

In this work, we have proven through experiments that we can make more efficient predictions in inter-subject tasks by using our method. In addition, the proposed method has the potential to be applied to various deep learning-based MI classification models. This study provides insight to develop BCI applications that work well in the real world without a calibration process for new users.

Acknowledgment. This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00432) and the Defense Challengeable Future Technology Program of Agency for Defense Development, Republic of Korea.

References

1. P. Sawangjai, S. Hompoonsup, P. Leelaarporn, S. Kongwudhikunakorn, and T. Wilaiprasitporn, “Consumer grade eeg measuring sensors as research tools: A review,” *IEEE Sensors Journal*, vol. 20, no. 8, pp. 3996–4024, 2019.
2. J. W. Choi, S. Huh, and S. Jo, “Improving performance in motor imagery bci-based control applications via virtually embodied feedback,” *Computers in Biology and Medicine*, vol. 127, p. 104079, 2020.
3. B. H. Kim, S. Jo, and S. Choi, “Alis: Learning affective causality behind daily activities from a wearable life-log system,” *IEEE Transactions on Cybernetics*, 2021.
4. N. Kaongoen, J. Choi, and S. Jo, “Speech-imagery-based brain–computer interface system using ear-eeg,” *Journal of neural engineering*, vol. 18, no. 1, p. 016023, 2021.
5. Z. Gao, X. Wang, Y. Yang, C. Mu, Q. Cai, W. Dang, and S. Zuo, “Eeg-based spatio–temporal convolutional neural network for driver fatigue evaluation,” *IEEE transactions on neural networks and learning systems*, vol. 30, no. 9, pp. 2755–2763, 2019.
6. L. S. Vidyaratne and K. M. Iftekharuddin, “Real-time epileptic seizure detection using eeg,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 11, pp. 2146–2156, 2017.
7. D. D. Chakladar, S. Dey, P. P. Roy, and M. Iwamura, “Eeg-based cognitive state assessment using deep ensemble model and filter bank common spatial pattern,” in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 4107–4114, IEEE, 2021.
8. D. D. Chakladar, S. Dey, P. P. Roy, and D. P. Dogra, “Eeg-based mental workload estimation using deep blstm-lstm network and evolutionary algorithm,” *Biomedical Signal Processing and Control*, vol. 60, p. 101989, 2020.
9. P. Autthasan, X. Du, J. Arnin, S. Lamyai, M. Perera, S. Itthipuripat, T. Yagi, P. Manoonpong, and T. Wilaiprasitporn, “A single-channel consumer-grade eeg device for brain–computer interface: Enhancing detection of ssvep and its amplitude modulation,” *IEEE Sensors Journal*, vol. 20, no. 6, pp. 3366–3378, 2019.
10. Y. Zou, V. Nathan, and R. Jafari, “Automatic identification of artifact-related independent components for artifact removal in eeg recordings,” *IEEE journal of biomedical and health informatics*, vol. 20, no. 1, pp. 73–81, 2014.
11. J.-H. Jeong, N.-S. Kwak, C. Guan, and S.-W. Lee, “Decoding movement-related cortical potentials based on subject-dependent and section-wise spectral filtering,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 3, pp. 687–698, 2020.

12. J. W. Choi, B. H. Kim, S. Huh, and S. Jo, "Observing actions through immersive virtual reality enhances motor imagery training," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 7, pp. 1614–1622, 2020.
13. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.
14. D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International conference on machine learning*, pp. 173–182, PMLR, 2016.
15. K. K. Ang and C. Guan, "Eeg-based strategies to detect motor imagery for control and rehabilitation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 4, pp. 392–401, 2016.
16. G. Pfurtscheller and C. Neuper, "Motor imagery and direct brain-computer communication," *Proceedings of the IEEE*, vol. 89, no. 7, pp. 1123–1134, 2001.
17. R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball, "Deep learning with convolutional neural networks for eeg decoding and visualization," *Human brain mapping*, vol. 38, no. 11, pp. 5391–5420, 2017.
18. V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces," *Journal of neural engineering*, vol. 15, no. 5, p. 056013, 2018.
19. B. H. Kim and S. Jo, "Deep physiological affect network for the recognition of human emotions," *IEEE Transactions on Affective Computing*, vol. 11, no. 2, pp. 230–243, 2018.
20. F. Lotte, "Signal processing approaches to minimize or suppress calibration time in oscillatory activity-based brain–computer interfaces," *Proceedings of the IEEE*, vol. 103, no. 6, pp. 871–890, 2015.
21. F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, "A review of classification algorithms for eeg-based brain–computer interfaces," *Journal of neural engineering*, vol. 4, no. 2, p. R1, 2007.
22. B. Blankertz, M. Kawanabe, R. Tomioka, F. U. Hohlefeld, V. V. Nikulin, and K.-R. Müller, "Invariant common spatial patterns: Alleviating nonstationarities in brain-computer interfacing.," in *NIPS*, pp. 113–120, 2007.
23. H. Wang and W. Zheng, "Local temporal common spatial patterns for robust single-trial eeg classification," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 16, no. 2, pp. 131–139, 2008.
24. H. Ramoser, J. Muller-Gerking, and G. Pfurtscheller, "Optimal spatial filtering of single trial eeg during imagined hand movement," *IEEE transactions on rehabilitation engineering*, vol. 8, no. 4, pp. 441–446, 2000.
25. B. Blankertz, G. Dornhege, M. Krauledat, K.-R. Müller, and G. Curio, "The non-invasive berlin brain–computer interface: fast acquisition of effective performance in untrained subjects," *NeuroImage*, vol. 37, no. 2, pp. 539–550, 2007.
26. K. K. Ang, Z. Y. Chin, C. Wang, C. Guan, and H. Zhang, "Filter bank common spatial pattern algorithm on bci competition iv datasets 2a and 2b," *Frontiers in neuroscience*, vol. 6, p. 39, 2012.
27. C. M. Bishop, "Pattern recognition," *Machine learning*, vol. 128, no. 9, 2006.
28. L. Fraiwan, K. Lweesy, N. Khasawneh, H. Wenz, and H. Dickhaus, "Automated sleep stage identification system based on time–frequency analysis of a single eeg channel and random forest classifier," *Computer methods and programs in biomedicine*, vol. 108, no. 1, pp. 10–19, 2012.

29. X. Tang and X. Zhang, "Conditional adversarial domain adaptation neural network for motor imagery eeg decoding," *Entropy*, vol. 22, no. 1, p. 96, 2020.
30. S. An, S. Kim, P. Chikontwe, and S. H. Park, "Few-shot relation learning with attention for eeg-based motor imagery classification," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10933–10938, IEEE, 2020.
31. F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1199–1208, 2018.
32. H. Wang, M. Xu, B. Ni, and W. Zhang, "Learning to combine: Knowledge aggregation for multi-source domain adaptation," in *European Conference on Computer Vision*, pp. 727–744, Springer, 2020.
33. T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.
34. C. Brunner, R. Leeb, G. Müller-Putz, A. Schlögl, and G. Pfurtscheller, "Bci competition 2008–graz data set a," *Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology*, vol. 16, pp. 1–6, 2008.
35. R. Caruana, S. Lawrence, and L. Giles, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping," *Advances in neural information processing systems*, pp. 402–408, 2001.
36. J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *International conference on machine learning*, pp. 647–655, PMLR, 2014.