EEG-Based Motor Imagery Classification with Deep Multi-Task Learning

Yaguang Song[1,2]
[1]State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
Beijing, China
songyaguang2016@ia.ac.cn

Danli Wang*[,1,2]
[1]State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
Beijing, China
danli.wang@ia.ac.cn

Kang Yue[1,2]
[1]State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
Beijing, China
einhep@gmail.com

Nan Zheng[1,2,3]
[1]State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences,
Beijing, China
[3]University of California, Berkeley,
Berkeley, USA
nan.zheng@ia.ac.cn

Zuo-Jun Max Shen
Department of Industrial Engineering and Operations Research and Department of Civil and Environmental Engineering, University of California, Berkeley,
Berkeley, USA
maxshen@berkeley.edu

***Abstract*—In the past decade, Electroencephalogram (EEG) has been applied in many fields, such as Motor Imagery (MI) and Emotion Recognition. Traditionally, for classification tasks based on EEG, researchers would extract features from raw signals manually which is often time consuming and requires adequate domain knowledge. Besides that, features manually extracted and selected may not generalize well due to the limitation of human. Convolutional Neural Networks (CNNs) plays an important role in the wave of deep learning and achieve amazing results in many areas. One of the most attractive features of deep learning for EEG-based tasks is the end-to-end learning. Features are learned from raw signals automatically and the feature extractor and classifier are optimized simultaneously. There are some researchers applying deep learning methods to EEG analysis and achieving promising performances. However, supervised deep learning methods often require large-scale annotated dataset, which is almost impossible to acquire in EEG-based tasks. This problem limits the further improvements of deep learning models for classification based on EEG. In this paper, we propose a novel deep learning method *DMTL-BCI* based on Multi-Task Learning framework for EEG-based classification tasks. The proposed model consists of three modules, the representation module, the reconstruction module and the classification module. Our model is proposed to improve the classification performance with limited EEG data. Experimental results on benchmark dataset, BCI Competition IV dataset 2a, show that our proposed method outperforms the state-of-the-art method by 3.0%, which demonstrates the effectiveness of our model.**

***Keywords*—EEG, Deep Learning, Multi-Task Learning**

I. Introduction

A Brain Computer Interface (BCI) can be defined as a system that uses the brain as a control center to communicate with external devices [1]. It does not depend on the normal output pathway of the brain (i.e. peripheral nerves and muscles) and translates the brain activity of a user into commands [2]. BCI technology has received much attention globally because of its significant meaning. It can be utilized to help disabled people as a rehabilitation device [3], such as the Motor Imagery (MI) task [4]. MI refers to imagination of moving the left , right hands or other body parts without actual movement [5]. It has been investigated in many BCI studies [5]. For healthy users, BCI systems will greatly enrich people's entertainment as a new interaction method [6]. Therefore, BCI is continuing a hot topic that is worthy of further study.

BCI systems control the external devices mainly by measuring and analyzing the Electroencephalogram (EEG) signals of users [7]. The reason is that the collection of EEG signals is non-invasive to the human body and the measurement method is relatively mature. EEG signals record integration of spontaneous electrical activities of a large number of brain cells from scalp, which is closely related with mental and physical states [8]. There are differences in terms of magnitude or frequency between different activities which can be utilized for analysis [9].

A complete brain computer interface system is divided into the following parts [10]: acquisition of EEG signals, EEG records pre-processing, feature extraction and selection from EEG signals for subsequent tasks, classification based on the extracted features and specific command execution. In this case, the classifier is a very important part which is responsible for transforming the features into commands of users. It determines whether the system is effective. Therefore, the performance of classification plays a decisive role in practical BCI systems [11].

* Corresponding author

978-1-7281-2009-6/$31.00 ©2019 IEEE paper N-19781.pdf

The mainstream methods for EEG-based classification tasks such as MI recognition mainly focus on the feature extraction and selection part. The commonly used strategy is that they extract hand-designed features from the time domain, frequency domain and time-frequency domain of EEG signals. However, the traditional feature extraction methods require researchers to have adequate domain knowledge, and the whole process is quite complicated. At the same time, due to the limitation of human, extracted features may not be generalized well on some tasks [12].

Deep learning has developed rapidly in recent years and achieved remarkable performances in many fields, such as computer vision [13] and speech recognition [14]. One of the most attractive characteristics of deep learning for EEG based tasks is the end-to-end learning [15]. It can directly learn from the raw EEG signals, also, the feature extractor and classifier are jointly optimized, which avoids complicated manual extraction process. Some researchers have applied deep learning methods to EEG-based classification tasks and achieve promising results [10]. Although the proposed methods have certain innovations, they are still subject to some limitations. The performance of EEG-based classification are not further improved [15].

These limitations are mainly due to deep learning itself. The biggest problem of deep learning is that it requires large-scale annotated data for supervised learning [16]. Large-scale annotated EEG datasets are almost impossible to acquire because of the high cost of data acquisition and annotation [11]. In this case, limited annotated samples are not enough for training shallow models, let alone deep ones. The two-stage training method can be adopted to alleviate the problem of insufficient data and improve the performance of classification [10]. However, it still can't meet the requirements of some tasks. For example, methods based on two-stage training sometimes requires a large amount of unlabeled data. Also the model can't achieve the best performance because the whole pipeline can't be optimized end-to-end.

To address the above challenge, a novel deep learning method based on Multi-Task Learning framework (MTL) [17] named *DMTL-BCI* is proposed for EEG-based classification. Our model consists of three modules, the representation module, the classification module and the reconstruction module. The representation module learns features directly from raw EEG signals. The obtained intermediate features are then sent to the classification module to make prediction. Similarly, the same features are sent to the reconstruction module to produce the reconstructed input. This is the multi-task learning framework we propose. The three modules are trained simultaneously and jointly optimized in an end-to-end manner. The features obtained by the representation module are called shared features, which work as a bridge to unite the two tasks [18]. Multi-task learning is a well-studied framework in machine learning and has been applied to many fields [19]. MTL is related to transfer learning. But tasks under MTL interact with each other which is different from transfer learning [20]. Through the interaction of two tasks, the shared intermediate features keep both the reconstruction and classification ability. This enhances the generalization ability of the model and improves the performance of classification with limited data.

The contributions of our method are as follows:

- A novel end-to-end deep learning model is proposed for EEG-based classification tasks. It can further improve the performance with limited annotated data.
- The proposed model is based on Multi-Task Learning. Three modules of our model are trained at the same time and jointly optimized. Because of the interaction of two tasks, the generalization ability of the model is enhanced.
- Experiments conducted on public datasets of motor imagery, BCI Competition IV dataset 2a [21], show that our method outperforms the FBCSP [22] and state of the art deep learning methods.

The remainder of the paper is organized as follows: Section II introduces the related works. Our method is detailed described in Section III . Section IV provides the experiments setup, results and discussions. Conclusions are given in Section V.

II. Related work

A lot of works have been proposed to improve the performance of EEG-based classification tasks. Müller-Gerking et al. proposed the Common Spatial Patterns (CSP) algorithm for Motor Imagery task [23]. CSP extracted features of EEG signals that distinguish two kinds of motion imaging through a set of spatial filters. It has been successfully applied to many applications of MI [24]. Ang et al. proposed the extension of original CSP which is named Filter bank common spatial pattern (FBCSP) [22]. FBCSP performed the best in the BCI competition IV dataset 2a [25]. Other methods such as independent component analysis (ICA) [26] were also utilized for feature extraction. For classification, traditional classifiers such as Support Vector Machine (SVM) were commonly adopted.

In recent years, deep learning methods for BCI classification have emerged. Lotte et al. provided a comprehensive review of classification methods for BCI tasks including recent deep learning based works [10]. Bozhkov et al. gave an overview of deep learning architectures for EEG-based tasks [27]. An et al. proposed a method utilizing a deep belief network (DBN) to perform feature extraction for MI classification [28]. Compared with DBN, Convolutional Neural Networks (CNNs) can learn the local features and patterns well [13]. Schirrmeister et al. explored various deep learning models for motor imagery and presented Shallow ConvNets, Deep ConvNets and others [15]. In particular, both the Shallow and Deep ConvNets outperformed FBCSP. Vernon et al. proposed EEGNet which first introduced the Depthwise and Separable convolutions to EEG-based classification tasks [12].

Some researchers extracted features by traditional methods and applied deep learning models to perform classification. Yang et al. used CSP for feature extraction and CNN for classification [29]. Semi-supervised learning was also adopted to alleviate the overfitting problem [10]. Li et al presented a self-training semi-supervised SVM algorithm for small training data [30]. Multi-task learning is relatively new to BCI tasks. Alamgir et al. proposed a multi-task method employing a parametric probabilistic approach for BCI classification [31]. To the best

the our knowledge, deep learning methods based on multi-task learning have not been introduce to BCI tasks.

III. Methods

EEG-based classification task is first formularized as follows. The definition and notations will be used in the remaining sections. Then a brief introduction of input format and our model *DMTL-BCI* is given. Afterwards, the representation learning module , the classification module and the reconstruction module are described separately. Then the multi-task framework for model training is given.

A. Problem Formulation

The recorded EEG signals are divided into several labeled segments which are also called trials [15]. Given an EEG input (a trial), the task is to predict the label correspondingly. The dataset can be denoted as $D^i = \{(X^1, y^1) \dots (X^{N_i}, y^{N_i})\}$, where N_i denotes the number of the trials for subject i. There are four categories including Hand(left), Hand(right), Feet and Tongue for each trial and can be denoted by 0-3 respectively. The EEG input of trial k, $1 \leq k \leq N_i$ is denoted as $X^k \in \mathbb{R}^{T*S}$, where T denotes the time steps recorded for each trial and S denotes the number of electrodes used in the experiment.

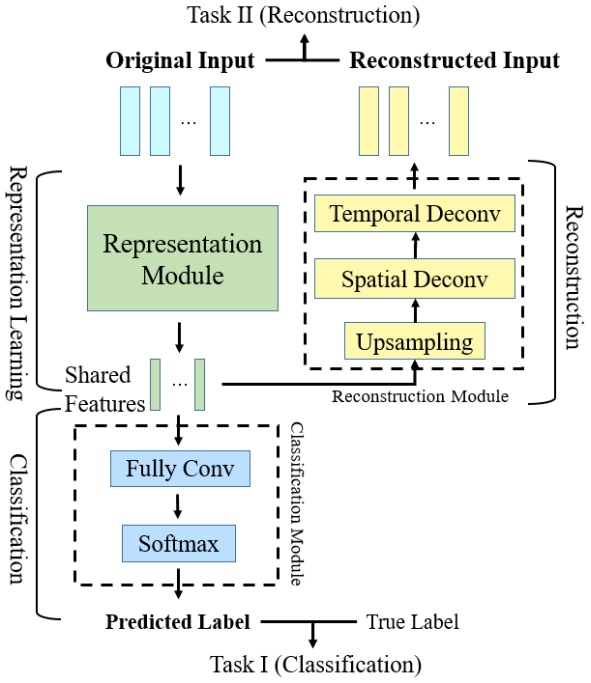

Fig. 1. Overall architechture of our model.

B. An Overview of Our Model

Before introducing the details of the model, it is necessary to determine the EEG input format $X^k \in \mathbb{R}^{T*S}$. Here the input is represented as a 2D matrix where the height of the matrix is the number of the time steps and the width of the matrix is the number of electrodes [15]. One advantage of this input format is that a specific convolution block is adopted to extract spatial and temporal features separately. Therefore, instead of using two dimension convolutional kernels like most deep ConvNets for natural images, one-dimensional convolutional kernels are adopted.

Fig. 1 illustrates the overall architecture of our model *DMTL-BCI*. It consists of three modules. Given the EEG input, the representation module in Fig. 1 first learns the fixed-size representation shared by the other two modules. The detailed network architecture of the representation module will be introduced below. On the one hand, the classification module shown in Fig.1 makes predictions based on the learned features. On the other hand, the reconstruction module shown in Fig.1 reconstructs the input based on the learned features. The shared intermediate features act as a bridge to unite the two modules. These three modules are jointly trained in an end-to-end manner. Because of utilizing the shared intermediate features, the classification module and reconstruction module interact and promote each other.

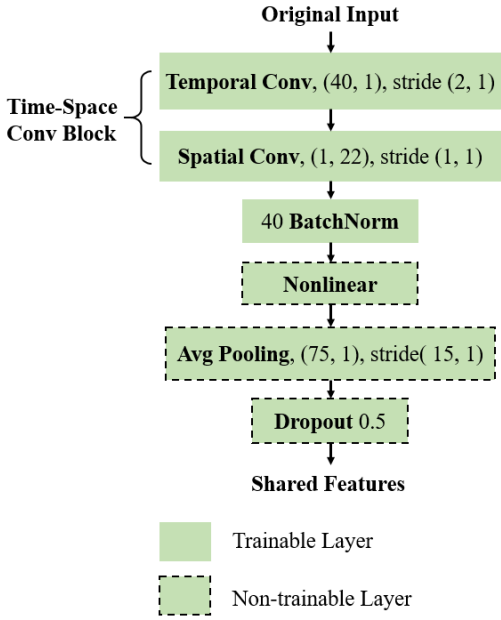

Fig. 2. Detailed nework architecture of the representation module.

C. Representation Learning Module

The representation learning module processes the EEG input to obtain the shared intermediate features for the downstream two tasks. As shown in Fig. 2, the module consists of a special convolutional block composed of two consecutive one-dimension convolutions, extracting the temporal features and the spatial features respectively. The detailed network architecture is described as follows:

- The Convolution Layer (time-specific). Since the temporal features are extracted separately, 1D convolution whose kernel size is $[K_{time}, 1]$ is adopted instead of the 2D convolution kernel commonly used in other tasks. This layer processes the EEG input along with the height of the matrix resulting a more compact structure. X^k denotes the input of the k-th trial. The processing can be written as follows:

$$h_{time}^{k} = Conv_{time}(X^{k}) \quad (1)$$

where $Conv_{time}$ is a function that transform the original EEG input X^k into features h_{time}^{k} using a 1D temporal convolution.

- The Convolution Layer (space-specific). Targeting at the spatial features, 1D convolution whose kernel size is $[1, \mathrm{K}_{space}]$ is adopted. The size of our convolution kernel K_{space} equals to the number of electrodes. After the processing of the spatial convolution layer, the width of EEG input becomes size 1.

$$h_{space}^{k} = Conv_{space}(h_{time}^{k}) \quad (2)$$

where $Conv_{space}$ denotes the 1D spatial convolution and h_{space}^{k} denotes the output features.

- Batch Normalization [32] and Nonlinear Layer. After the processing of the two convolutional layers, the distribution of input may change and the shift of the data distribution would affect the training of the network [32]. Therefore, batch normalization is introduced to eliminate the influence of the distribution shift.

$$h_{BN}^{k} = BN(h_{space}^{k}) \quad (3)$$

$$h_{Nonlin}^{k} = Nonlin(h_{BN}^{k}) \quad (4)$$

where BN denotes the batch normalization and $Nonlin$ denotes the nonlinear units applied to h_{BN}^{k}. Output features are denoted as h_{Nonlin}^{k}.

- The Average Pooling Layer. After the processing of the temporal convolution layer, more compact features along with the height of the input matrix are obtained. The average pooling layer is utilized to aggregate the features of time dimension and transform the low-level features to high-level features. Then.

$$h_{pooling}^{k} = AvgPool(h_{Nonlin}^{k}) \quad (5)$$

where AvgPool denotes the 1D temporal pooling and $h_{pooling}^{k}$ denotes the output features.

- The Dropout Layer. The dropout layer randomly discard a portion of the features with a certain probability to reduce the risk of overfitting.

$$h_{Shared}^{k} = Dropout(h_{pooling}^{k}) \quad (6)$$

where the output of the representation learning module is denoted as shared intermediate features h_{shared}^{k}.

D. Classification Module

The classification module consists of a convolutional layer and a softmax classification layer as shown in Fig. 1. The detailed network architecture is described as follows:

- The Convolution Layer (fully). The fully convolutional layer is designed to process the shared features. The size of the output is [1, 1]. Since there are four classes in this experiment, the convolution layer consists of four kernels leading to four channels. The output of this layer is the final activations of the network.

$$h_{fully}^{k} = Conv_{fully}(h_{Shared}^{k}) \quad (7)$$

where $Conv_{fully}$ denotes the fully convolution applied to shared intermediate features. h_{fully}^{k} denotes the output features.

- The softmax Layer. This layer works in a fully connected manner. The previously obtained activations are sent to the softmax layer and the output is a probability distribution which denotes the likelihood of each class.

$$\hat{y}^{k} = Softmax(h_{fully}^{k}) \quad (8)$$

where $\hat{y}^{k}$ denotes the probability distribution of each class.

For the classification task, we use the Cross Entropy (CE) loss to evaluate the model. The formulation is as follows:

$$\mathrm{Loss}_{CE} = \frac{1}{N}\sum_{k=1}^{N} \mathrm{loss}(y^{k}, \hat{y}^{k}) \quad (9)$$

where N denotes the number of the trials, y^k is the label of the k-th trial and $\hat{y}^k$ is the output of the model correspondingly. $\mathrm{loss}(y^k, \hat{y}^k)$ denotes the cross entropy loss of the k-th trial.

E. Reconstruction Module

The shared intermediate features are sent to this module for reconstruction. It is a structure of autoencoder where the representation module is the encoder and the reconstruction module is the decoder as demonstrated in Fig. 1. We apply the deconvolution with stride (i.e., transposed convolution) to decode the shared feature representation as a mirroring step for the representation learning part [18]. Before the consecutive deconvolution layers, we apply upsampling as the mirror operation for the average pooling layer to increase the size of the feature. After the process of two deconvolution layers, the size of the feature gradually increases and the output of the reconstruction module has the same size of the original input. The detailed network architecture is described as follows:

- The Upsampling Layer. This layer works as a mirror operation as the average pooling to increase the size of the shared intermediate features. A common implementation is interpolation.

$$h_{up}^{k} = Upsampling(h_{shared}^{k}) \quad (10)$$

where h_{up}^{k} denotes the outputs of upsampling layer.

- The Deconvolution Layer (space-specific). By applying the deconvolution, the output size would restore to the same as the output of the temporal convolutional layer.

$$h_{deconv1}^{k} = Deconv_{space}(h_{up}^{k}) \quad (11)$$

where $Deconv_{space}$ denotes the 1D deconvolution along with the space dimension and $h_{deconv1}^{k}$ denotes the output.

- The Deconvolution Layer (time-specific). By applying this deconvolution layer, the output size would restore to the same as the original input.

$$\hat{X}^k = Deconv_{time}(h^k_{deconv1}) \tag{12}$$

where $Deconv_{time}$ denotes the 1D deconvolution along with the time dimension and the reconstructed input is denoted as $\hat{X}^k$.

For the reconstruction task, we use the Mean Square Error (MSE) to evaluate the model. The loss is computing as follows:

$$\text{Loss}_{MSE} = \frac{1}{N}\sum_{k=1}^{N}\left\|X^k - \hat{X}^k\right\|^2 \tag{13}$$

where N denotes the number of trails, X^k is the original input of the k-th trial and $\hat{X}^k$ denotes the reconstructed input of the k-th trial.

F. Joint Learning Framework

The three modules for representation, classification and reconstruction for EEG-based classification tasks are introduced above. Instead of following the unsupervised learning paradigm that utilizes the encoder-decoder structure for pre-training, we cast the supervised task as a multi-task learning problem [18]. The representation learning module, the classification module and the reconstruction module are jointly trained. In principle, the shared intermediate features learned in this joint optimizing framework keep both reconstruction and classification ability [18]. This method allows the two tasks to promote each other and improves the generalization ability of the model. Using Θ to denotes all the parameters of the model, then the training objective function can be written as follows:

$$\mathcal{L}(\Theta) = \text{Loss}_{CE} + \alpha \cdot \text{Loss}_{MSE} + \lambda\|\Theta\|^2 \tag{14}$$

where Loss_{CE} denotes the loss for the classification task which optimizes the model in a supervised manner. Loss_{MSE} denotes the loss for the reconstruction task which optimizes the model in an unsupervised manner. The two losses are described above. Hyperparameter $\alpha > 0$ is utilized to balance the relative importance of the supervised and unsupervised loss. In practice, α is fixed as a prior for convenience. We apply l_2 regularization term with coefficient to alleviate overfitting. Our task is to minimize $\mathcal{L}(\Theta)$. All trainable parameters of the network are trained in an end-to-end manner.

IV. EXPERIMENTS

Eexperiments are conducted on a benchmark dataset and the proposed method is compared with the state-of-the-art approaches. We first introduce the public dataset BCI Competition IV dataset 2a and the baseline methods used in this experiment. Next, evaluation metrics and implementation details are explained. Two training paradigm are adopted: (1) One is the commonly used method on BCI Competition IV dataset 2a, that is, models for 9 subjects are trained and tested individually. (2) Another training method as a supplement is to train the model with the combination of all subjects' data then test individually. The purpose of this experiment is to fully evaluated all the methods and report performances of each model when the amount of training data is not small.

A. Dataset Description

The performance of models are evaluated on a public dataset for motor imagery, BCI Competition IV dataset 2a. BCI Competition IV dataset 2a consists of EEG data from a total of 9 subjects. There are two sessions recorded on different days for each subject. One is for training and the other is for testing. Each session includes 288 trials. The model performance is evaluated in a 5-fold cross validation manner. The experimental results are based on the second session. Each trial are recorded with 22 EEG electrodes and 3 electrooculogram (EOG) channels. But only the 22 EEG channels are utilized in this experiment [21]. There are four type of labels in BCI Competition IV dataset 2a, which correspond to movements of the left hand, the right hand, the feet and the tongue.

The BCI Competition Dataset IV 2a is sampled at 250Hz and bandpass-filtered between 0.5Hz and 100Hz [21]. In this experiment we low-pass filter the dataset to below 38Hz. Also, in our study the length of each trials is set to 4.5 seconds as the input to the network. It starts from 500ms before the start cue of each trial until the end cue [15]. To show the effectiveness of deep models learning from raw signals and ensure that the proposed method can be applied to wider range of tasks, only minimum per-processing is conducted following the procedure described in [15].

B. Baseline Methods

To show the effectiveness of our model, the state of the art methods on BCI Competition IV dataset 2a are chosen for comparison. The baseline methods are listed as follows:

1) Filter Bank Common Spatial Patterns (FBCSP) [25]: It is designed to extract band power features of EEG. A classifier is trained to predict labels based on the features.

2) Shallow ConvNet [15]: Inspired by FBCSP algorithm, Shallow ConvNet extract features in a similar way. But Shallow ConvNet uses convolutional neural network to do all the computations and all the steps are optimized in an end-to-end manner.

3) Deep ConvNet [15]: It has four convolution-pooling blocks and is much deeper than Shallow ConvNet.

4) EEGNet [12]: It has two convolution-pooling blocks. The difference between EEGNet and ConvNets introduced above is that EEGNet uses depthwise and separable convolution.

C. Evalution Metric

The overall accuracy for each subject is computed and the the average accuracy for each methods are reported. The overall accuracy is calculated as follows:

$$accuracy = \frac{\sum_{c=1}^{C} TP_c}{N} \tag{15}$$

where TP_c is the number of the true positive samples of class c, C is the number of classes which is four in this experiment and N is the number of the trials.

D. Implementation Details

The training phase for deep learning methods is separated into two stages [15]. The first stage is that the model is optimized on the training set and updates its parameters according to the performance on the validation set. After training for 300 epochs, the model enters the second training stage. The validation set is added to the training set as the whole training set for the second stage. At last, report the performance of the model on the test set.

The learning rate is set to 0.001 and Adam optimizer is adopted whose parameters beta1 and beta2 are set to 0.9, 0.999 respectively. The dropout rate is set to 0.5. For the batch normalization layer used in the experiment, a epsilon constant is added for numerical stability. The mean value and variance for the batch normalization are fixed during validation and test. The batch size is set to 30

E. Results and Discussion

As introduced above, models are trained and tested on each subject's data independently following the previous works [25]. This section includes the performance comparison of different models for each subject. Besides that, performance of models with and without the reconstruction module are reported to evaluate the effectiveness of multi-task framework. We also reveal how the coefficient α influences performance of the model. At last, performance of models trained on the combination of all the data from 9 subjects are reported and analyzed.

TABLE I. PERFORMANCE COMPARISON OF DIFFERENT METHODS UNDER INDEPENDENT TRAINING PARADIGM. BEST SCORES ARE IN BOLD.

Subject	Accuracy % (mean ± std. dev.)				
	FBCSP	Shallow ConvNet	Deep ConvNet	EEGNet	DMTL-BCI
1	-	79.0±1.1	76.5± 3.2	77.3±2.4	**83.5±1.2**
2	-	49.8±4.0	50.6± 3.0	**60.2±3.7**	49.0±2.3
3	-	90.9±0.7	85.0±2.1	88.6±1.1	**92.7±1.0**
4	-	63.3±2.8	67.6±2.0	63.1±1.4	**74.9±1.0**
5	-	70.8±2.9	**72.4±1.3**	69.7±1.5	71.3±3.0
6	-	58.0±1.7	55.1±2.2	57.8±3.1	**63.7±1.2**
7	-	79.7±5.4	71.7±1.0	69.2±2.0	**80.8±2.5**
8	-	**81.8±1.4**	74.4±2.6	73.7±1.5	80.0±1.0
9	-	77.4±1.6	79.2±3.4	72.3±4.2	**81.7±1.2**
AVG	0.68	72.3±2.4	70.3±2.3	70.2±2.3	**75.3±1.6**

Table I shows the performance of the different models on the BCI Competition IV dataset 2a. including the accuracy of the model on each test set and its average accuracy. The results of FBCSP are from [15] which does not provide results of individuals. Overall, all methods achieve promising performances. All the deep learning models in this experiment outperform the FBCSP method. To the best of our knowledge, FBCSP is the best performing traditional method on BCI Competition IV dataset 2a [15]. The results demonstrate the effectiveness and potential of deep convolutional neural networks for EEG-based classification tasks.

According to the average accuracy results in Table I, our proposed model performs the best, reaching 75.3%， which exceeds the state-of-the-art methods by 3.0%. In addition to the final average accuracy, our model achieves the best results in 6 subjects (1, 3, 4, 6, 7, and 9). The above results show the effectiveness of our method. EEGNet performs the worst among the deep learning models and the average accuracy is 70.2%. Except for the subject 2, EEGNet achieved the worst results in other subjects showing that deep separable convolutions may not help for the classification task. EEGNet may be a good alternative choice for online tasks because it can reduce the amount of computation and speed up training and testing [12]. The average accuracy of the Deep ConvNet is 2% lower than that of Shallow ConvNet. Shallow ConvNet outperforms the Deep in 5 subjects (1, 3, 6, 7, and 8) and the Deep performs better in several other subjects. The Shallow has only one convolution-pooling module while the Deep has four. It shows that in this experiment, a deeper and more complex network does not help improve the performance. In contrast, it is easier for Deep ConvNet to overfit. The reason is that there is little annotated data for training. In this case, the size of the dataset becomes the limitation of classification task.

Fig. 3. Performance comparison of models with and without reconsruction module. 'w/o' is the abbreviation of 'without'.

The results in Table I show that our model outperforms all the other methods. Then, a ablation experiment is designed to demonstrate that the improvement is indeed due to our proposed classification-reconstruction multi-task learning framework. Fig. 3 shows the performance of models with and without the reconstruction module. The horizontal axis in Fig. 3 represents different subjects and the average accuracy. The vertical axis represents the corresponding accuracy. According to the average accuracy, the multi-task model outperforms the single-task model by 2.5%. Besides that, the multi-task model achieves better results in almost every subject (1, 3, 4, 5, 6, 7, and 9) than the single-task model which shows the effectiveness of our method. The above results and analysis show that the multi-task framework consisting of classification and reconstruction can improve the performance of the model without increasing training data. The reason is that due to multi-task learning, features obtained by the representation module keep both the

classification and reconstruction ability [18]. This greatly improve the generalization ability of our model [17].

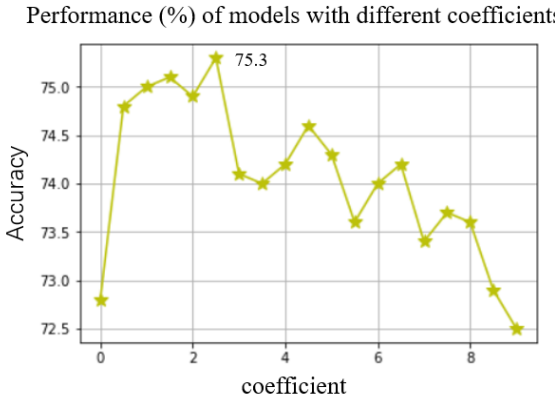

Fig. 4. Performance comparison of models trained with different weight coefficients (the hyperparameter α) ranging from 0-9. The vertical axis corresponds to the average accuracies (%). Model achieved the best performance (75.3%) when α was about 2.5.

For the multi-task learning framework, the relative importance of two tasks has a decisive influence on the performance of the model. In order to explore the relation of the value of weight coefficient α and the classification accuracy, an experiment is conducted and the results are shown in Fig. 4. The horizontal axis represents different values of the weight coefficient range from 0-9. The vertical axis corresponds to the average accuracies of the model with different coefficients on the test set. The curve in the figure shows a certain trend. When the coefficient is zero, only the classification task is performed. As the weight coefficient increases, the importance of the reconstruction task increases gradually, and the model achieves better performance. When the coefficient is greater than a certain value, that is, the proportion of the reconstruction task is too large, the performance of the model decreases rapidly. This result is instructive for determining the relative importance of two tasks in the experiment.

The above results are based on the first training paradigm that models are trained on each subject's data. This is also the method commonly used in previous works [25]. In this experiment setup, the dataset for training is small and our model performs the best in this case which proves the effectiveness of multi-task learning framework. At the last part of experimental section, another training paradigm is applied to evaluate each model. We train the model using the data from all the subjects and report the results on each test set. The model performance is evaluated in a 5-fold cross validation manner. As Table II shows, the final results are not better than the independent training paradigm because of the large difference between subjects [16]. According to the average accuracy, Deep ConvNet achieves the best performance with a small advantage compared with our model. It can be suggested that the increase of annotated data boost the performance of the deeper model. This shows that in the previous experiment, the amount of data is indeed the bottleneck of deep learning methods. However, our proposed method still outperforms other methods with significant improvement. It shows that our framework can break this limitation to a certain extent, enhance the generalization of the model and improve the performance with limited data.

TABLE II. PERFORMANCE COMPARISON OF DIFFERENT METHODS UNDER COMBINED TRAINING PARADIGM. BEST SCORES ARE IN BOLD.

Subject	Accuracy (mean ± std. dev.)			
	Shallow ConvNet	Deep ConvNet	EEGNet	DMTL-BCI
1	79.6±2.4	78.8±2.6	77.5±1.9	**80.3±1.6**
2	52.5±1.7	51.8±0.7	**60.6±4.3**	50.3±1.6
3	86.6±0.8	**86.8±1.8**	84.2±1.9	85.5±1.8
4	67.5±1.9	**71.6±2.6**	65.7±4.3	70.6±2.0
5	64.4±2.6	**68.7±3.8**	66.4±2.7	66.2±4.7
6	59.1±2.9	**64.6±2.0**	58.5±1.8	60.6±1.5
7	**84.1±1.5**	82.3±1.8	76.2±1.1	83.0±2.5
8	**83.5±1.5**	80.9±1.7	81.5±0.8	82.8±0.9
9	78.2±1.5	75.4±2.5	67.7±1.7	**78.4±2.0**
AVG	72.8±1.9	**73.4±2.2**	70.9±2.3	73.1±2.0

V. CONCLUSION

In this paper, A deep learning method based on multi-task learning framework is proposed for classification based on EEG signals. The proposed model consists of the representation module, the classification module and the reconstruction module. Three modules are jointly optimized with multi-task learning in an end-to-end manner. By sharing representation between related tasks, the classification module and reconstruction module could interact and promote each other. Shared intermediate features keep both the classification and reconstruction ability and enhance the generalization ability of our model for the classification task. Experiments are conducted on the public dataset, BCI Competition IV dataset 2a. The experimental results show that out model outperforms the state of the art methods. Furthermore, ablation study and parameter analysis show the effectiveness of our model. In the future, we plan to combine semi-supervised learning and make full use of unlabeled data to improve the performance of EEG based classification.

ACKNOWLEDGMENT

This research is supported by the National Key Research and Development Program under Grant No. 2016YFB0401202, and the National Natural Science Foundation of China under Grant No. 61872363, 61672507, 61272325, 61501463 and 61562063.

REFERENCES

[1] F. Lotte, L. Bougrain, and M. Clerc, "Electroencephalography (EEG)-Based Brain-Computer Interfaces," in Wiley Encyclopedia of Electrical and Electronics Engineering, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2015, pp. 1–20.

[2] B. Graimann, B. Allison, and G. Pfurtscheller, "Brain–Computer Interfaces: A Gentle Introduction," in Brain-Computer Interfaces, B. Graimann, G. Pfurtscheller, and B. Allison, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 1–27.

[3] A. Ramos-Murguialday et al., "Brain-Machine-Interface in Chronic Stroke Rehabilitation: A Controlled Study," Ann Neurol, vol. 74, no. 1, pp. 100–108, Jul. 2013.

[4] J. R. Wolpaw, D. J. McFarland, G. W. Neat, and C. A. Forneris, "An EEG-based brain-computer interface for cursor control," Electroencephalography and Clinical Neurophysiology, vol. 78, no. 3, pp. 252–259, Mar. 1991.

[5] Y. R. Tabar and U. Halici, "A novel deep learning approach for classification of EEG motor imagery signals," Journal of Neural Engineering, vol. 14, no. 1, p. 016003, Feb. 2017.

[6] D. Coyle, J. Principe, F. Lotte, and A. Nijholt, "Guest Editorial: Brain/neuronal - Computer game interfaces and interaction," IEEE Transactions on Computational Intelligence and AI in Games, vol. 5, no. 2, pp. 77–81, Jun. 2013.

[7] L. F. Nicolas-Alonso and J. Gomez-Gil, "Brain Computer Interfaces, a Review," Sensors (Basel), vol. 12, no. 2, pp. 1211–1279, Jan. 2012.

[8] X. Ma et al., "EEG based topography analysis in string recognition task," Physica A: Statistical Mechanics and its Applications, vol. 469, pp. 531–539, Mar. 2017.

[9] S. Kar, M. Bhagat, and A. Routray, "EEG signal analysis for the assessment and quantification of driver's fatigue," Transportation Research Part F: Traffic Psychology and Behaviour, vol. 13, no. 5, pp. 297–306, Sep. 2010.

[10] F. Lotte et al., "A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update," Journal of Neural Engineering, vol. 15, no. 3, p. 031005, Jun. 2018.

[11] C. Tan, F. Sun, W. Zhang, T. Kong, C. Yang, and X. Zhang, "Adaptive Adversarial Transfer Learning for Electroencephalography Classification," in 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 2018, pp. 1–8.

[12] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "EEGNet: A Compact Convolutional Network for EEG-based Brain-Computer Interfaces," Journal of Neural Engineering, vol. 15, no. 5, p. 056013, Oct. 2018.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," arXiv:1512.03385 [cs], Dec. 2015.

[14] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional Neural Networks for Speech Recognition," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 22, no. 10, pp. 1533–1545, Oct. 2014.

[15] R. T. Schirrmeister et al., "Deep learning with convolutional neural networks for EEG decoding and visualization," Human Brain Mapping, vol. 38, no. 11, pp. 5391–5420, Nov. 2017.

[16] V. Jayaram, M. Alamgir, Y. Altun, B. Schölkopf, and M. Grosse-Wentrup, "Transfer Learning in Brain-Computer Interfaces," Dec. 2015.

[17] S. Ruder, "An Overview of Multi-Task Learning in Deep Neural Networks," arXiv:1706.05098 [cs, stat], Jun. 2017.

[18] Y. Zhang, D. Shen, G. Wang, Z. Gan, R. Henao, and L. Carin, "Deconvolutional Paragraph Representation Learning," arXiv:1708.04729 [cs, stat], Aug. 2017.

[19] G. Lu, X. Zhao, J. Yin, W. Yang, and B. Li, "Multi-task learning using variational auto-encoder for sentiment classification," Pattern Recognition Letters, Jun. 2018.

[20] Y. Zhang and Q. Yang, "A Survey on Multi-Task Learning," arXiv:1707.08114 [cs], Jul. 2017.

[21] C. Brunner, R. Leeb, G. R. Muller-Putz, and A. Schlogl, "BCI Competition 2008 – Graz data set A," p. 6.

[22] Kai Keng Ang, Zhang Yang Chin, Haihong Zhang, and Cuntai Guan, "Filter Bank Common Spatial Pattern (FBCSP) in Brain-Computer Interface," in 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 2008, pp. 2390–2397.

[23] J. Müller-Gerking, G. Pfurtscheller, and H. Flyvbjerg, "Designing optimal spatial filters for single-trial EEG classification in a movement task," Clinical Neurophysiology, vol. 110, no. 5, pp. 787–798, May 1999.

[24] E. A. Mousavi, J. J. Maller, P. B. Fitzgerald, and B. J. Lithgow, "Wavelet Common Spatial Pattern in asynchronous offline brain computer interfaces," Biomedical Signal Processing and Control, vol. 6, no. 2, pp. 121–128, Apr. 2011.

[25] 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," Front Neurosci, vol. 6, Mar. 2012.

[26] P. Comon, "Independent component analysis, A new concept?," Signal Processing, vol. 36, no. 3, pp. 287–314, Apr. 1994.

[27] L. Bozhkov and P. Georgieva, "Overview of Deep Learning Architectures for EEG-based Brain Imaging," International Joint Conference on Neural Networks, p. 7, 2018.

[28] X. An, D. Kuang, X. Guo, Y. Zhao, and L. He, "A Deep Learning Method for Classification of EEG Data Based on Motor Imagery," in Intelligent Computing in Bioinformatics, vol. 8590, D.-S. Huang, K. Han, and M. Gromiha, Eds. Cham: Springer International Publishing, 2014, pp. 203–210.

[29] H. Yang, S. Sakhavi, K. K. Ang, and C. Guan, "On the use of convolutional neural networks and augmented CSP features for multi-class motor imagery of EEG signals classification," in 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2015, pp. 2620–2623.

[30] Y. Li, C. Guan, H. Li, and Z. Chin, "A self-training semi-supervised SVM algorithm and its application in an EEG-based brain computer interface speller system," Pattern Recognition Letters, vol. 29, no. 9, pp. 1285–1294, Jul. 2008.

[31] M. Alamgir, M. Grosse-Wentrup, and Y. Altun, "Multitask Learning for Brain-Computer Interfaces," p. 8.

[32] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," arXiv:1502.03167 [cs], Feb. 2015.