2021 International Joint Conference on Neural Networks (IJCNN) | 978-1-6654-3900-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/IJCNN52387.2021.9534286

Filter Bank Adversarial Domain Adaptation For Motor Imagery Brain Computer Interface

Yukun Zhang[*†], Shuang Qiu[†], Wei Wei[†*], Xuelin Ma[†*§], Huiguang He[*†‡]

[*]*School of Artificial Intelligence, University of Chinese Academy of Sciences,* Beijing 100049, China

[†]*Research Center for Brain-Inspired Intelligence, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences,* Beijing 100190, China

[‡]*Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences,* Beijing 100190, China

[§]*JD.com*

Email: {zhangyukun2019, shuang.qiu, weiwei2018}@ia.ac.cn, maxuelin5@jd.com, huiguang.he@ia.ac.cn

***Abstract*—Motor imagery (MI) based Brain–computer interface (BCI) is a promising BCI paradigm that can help neuromuscular injury patients to recover or replace their motor abilities. However, electroencephalography (EEG) based MI-BCI suffers from its long calibration time and low classification accuracy, which restrict its application. Thus, it is important to reduce the calibration time of MI-BCI and enhance its prediction accuracy. In this study, we propose a filter bank Wasserstein adversarial domain adaptation framework (FBWADA) that uses a short amount of training data from a new target subject, and all collected data from an existing subject. A Convolutional Neural Networks (CNN) based feature extractor is designed to extract feature from EEG data. Filter bank strategy is employed to extract feature from multiple sub bands and integrate predictions from all sub bands. Wasserstein Generative Adversarial Networks (WGAN) based domain adaptation network aligns the marginal and conditional distribution of target and source. We evaluate our method on Data set 2a of BCI competition IV. Experiment results show that our method achieves the best performance among compared methods under different amount of training data. Performance of our method trained with certain blocks of data is similar to or better than the best comparing method trained with one more block. This indicates that our method could reduce the need for training data for at least one block.**

Keywords*—*brain-computer interface, motor imagery, transfer learning, domain adaptation, filter bank, calibration reduction

I. Introduction

Brain–computer interface (BCI) establishes a direct pathway that does not depend on the brain's normal output channels of peripheral nerves and muscles for users to communicate with outside world.[1] Motor imagery (MI) based BCI decodes spontaneous human motor intention from brain signals, which can help neuromuscular injury patients to recover or replace their motor abilities.[2-6] Also, MI-BCI can be applied in education, entertainment, and smart home applications.[7-12] Compared with electrocorticography (ECoG), magnetoencephalography (MEG), functional magnetic resonance imaging (fMRI), and functional Near Infrared (fNIR), electroencephalography (EEG) is the most commonly used neuroimaging method in BCI systems due to its high temporal resolution, easy access and high safety.

Traditional EEG decoding methods mainly consist of feature extraction and classification. Common Spatial Pattern (CSP) is one of the most effective MI feature extraction methods.[13] Filter bank common spatial pattern (FBCSP) which won the BCI competition IV in 2008[14] is a powerful of advanced version of CSP. Support vector machine (SVM) and Linear Discriminant Analysis (LDA) are the most widely used classification method. Recently, deep learning (DL) has achieved great success in the field of computer vision and nature language processing. Some researchers proposed some DL-based algorithms to decode EEG signals, and which gain equal or better performance than traditional machine learning methods.[15-17] For example, R. T. Schirrmeister et.al proposed shallow and deep convolutional neural networks (CNN) for MI decoding.[15] V. J. Lawhern et.al proposed a compact CNN structure named EEGNet.[16] S. Sakhavi et.al proposed a CSP based neural network C2CM.[17]

Despite the great achievement and the bright future in the field of MI-BCI，EEG based MI-BCI still suffers from its long calibration time and low classification accuracy, which restrict its application. Building an MI decoding model needs a lot of data. Especially for a DL method, more data brings the higher accuracy and robustness of the model. However, acquiring the training data for building model is time consuming. Besides, MI-EEG signals have a large individual difference. Directly building a model using multiple subjects' data usually leads to a poor classification performance.

Transfer learning can transfer knowledge from one domain to another domain.[18] Through the transfer learning method, we can utilize MI data from an existing subject to facilitate the model training of a new subject and reduce the needs for collecting new data. Subject with already collected data is called source and the new subject is called target. However, the EEG signals of different subjects have large pattern difference.

Transfer learning for MI classification is still a difficult problem. This may be caused by two factors. Different subjects have various sensitive frequency bands during motor imagery. Also, there are large marginal and conditional distribution gap between subjects.

In this study, we proposed a Filter Bank Wasserstein Adversarial Domain Adaptation (FBWADA) framework to reduce the need for data during calibration and improve the classification accuracy for MI-BCI. This framework uses a short amount of training data from a new target subject, and all collected data from a source subject. Our FBWADA framework includes three parts: First, we construct a classification model. We perform convolution operation through each EEG channel, which can be taken as time domain filtering. Then, convolution across channels is applied to achieve spatial filtering. A full connection neural network is employed to output predictions. Secondly, to deal with sensitive frequency band difference across subject, filter bank strategy is employed to extract the features from multiple sub bands and integrate predictions from all sub bands. Thirdly, a Wasserstein Generative Adversarial Networks (WGAN) based domain adaptation network align the marginal and conditional distribution of target and source. To further align the conditional distribution of target and source, we apply adversarial training separately for each class. In other words, we pull the intra-class cross domain difference.

The main contribution of this paper is summarized as follow:

- We propose a filter bank CNN model for MI classification. To the best of our knowledge, this is the first work that use filter bank method in deep learning MI classification model.
- We incorporate WGAN into our model for adversarial training, which transfer knowledge from source data to target data.
- We evaluate our method on Data set 2a of BCI competition IV. Results show that our method archives better performance compared with existing methods that use short amount of target data and gets better classification accuracy.

The rest of this paper is organized as follows. In Section II, we introduce the proposed method. In Section III, we describe the experiment setting. Experiment result and discussion is presented in section IV. Section V is the conclusion if this study.

II. METHOD

A. Notifications

We first introduce the notations and definitions for later use. Let $x \in \mathbb{R}^{C\times T}$ represents one EEG sample with C channels and T time points. $y \in \{1, \ldots, N_{class}\}$ represents the class label, where N_{class} is the number of classes. $X = \{x_i\}_{i=1}^{N}$ is a set of N samples and $Y = \{y_i\}_{i=1}^{N}$ is the corresponding label set. The marginal distribution of X is denoted as $P(X)$ and the conditional distribution of X is $P(X|Y)$. Target subject is the one we want to train a predict model for. A source subject with many already collected samples is adopt to assist the model training process. Let $D^t = \{(x_i^t, y_i^t)\}_{i=1}^{N_t}$ denotes N_t labeled EEG samples from target subject and $D^s = \{(x_j^s, y_j^s)\}_{j=1}^{N_s}$ denotes N_s labeled EEG samples from source subject, where x_i^t, x_j^s are the i-th and j-th sample from target subject and source subject respectively and y_i^t, y_j^s are the corresponding label. There is a gap between target distribution and source distribution, $P(X^t) \neq P(X^s)$, $P(X^t|Y^t) \neq P(X^s|Y^s)$.

Our goal is to train a model to predict class label of test sample with little amount of training data from target subject. While utilizing data from source subject could increase the size of training sample, the distribution gap between target and source data may lead to worse model accuracy. The motivation of our work is to align the distribution of target and source in feature space, such that $P(Y^t|F^t) \approx P(Y^s|F^s)$, where $F^t = \{f_i^t\}_{i=1}^{N_t}$, $F^s = \{f_j^s\}_{j=1}^{N_s}$. f_i^t, f_j^s is the feature of x_i^t and x_j^s. Then, training a classifier on the common feature space would work for both target and source domains.

B. Network architecture

The pipeline of our method is illustrated in Fig. 1. A filter bank including N_{band} FIR band pass filters is employed to extract multiband EEG signals. The passbands we choose are 4-10, 8-14, …, 32-38 Hz. These passbands are selected because they cover and uniformly distributed in the range of 4-38Hz which is the main response frequency range of MI.[13-15]

A feature extractor is then used to extract feature of the band passed signal. A temporal convolutional network is first adopted to extract temporal feature for each channel. The kernel shape of the temporal convolutional network is 1×25. Next, a spatial convolutional network is used to find the spatial pattern of multiband EEG signal. Spatial convolutional network could also be regarded as a spatial filter which takes linear combination of EEG channels. The kernel shape of the spatial filter is $N_c \times 1$, where N_c is the number of channels, thus each spatial convolutional kernel act as one combination of all EEG channels. Temporal and spatial filters are designed by the inspiration of traditional MI classification methods.[13] We apply spatial and temporal filter separately because that it simulates the behavior of those traditional methods. Besides, we want each spatial filter extracts one unique spatial pattern. In other words, each spatial filter keeps same combination of channels for every temporal point, which is reasonable for multi-channel EEG processing. Furthermore, compared with one 2-dimension kernel, separately apply two 1-demension kernels reduce parameters which might enhance the robustness of our model. Filtered signal is then squared to obtain power information. Finally, an average pooling model with kernel shape of 1×75 is used to reduce the feature dimension. The stride of pooling is 15. The pooling parameters are selected according to shallow CNN.[15] Dimension of the extracted feature is about 40×70. A classifier with two fully connected layers is adopted to output class label. The activate function of the first layer is Leaky ReLU. Log SoftMax is used to output class predicts. Batch Normalization and dropout techniques are adopted to enhance the model performance.

Inspired by the adversarial training, we employ a discriminator to close the feature distribution between the target and source domains. While the classifier is used to predict class label from the extracted features, the discriminator aims to

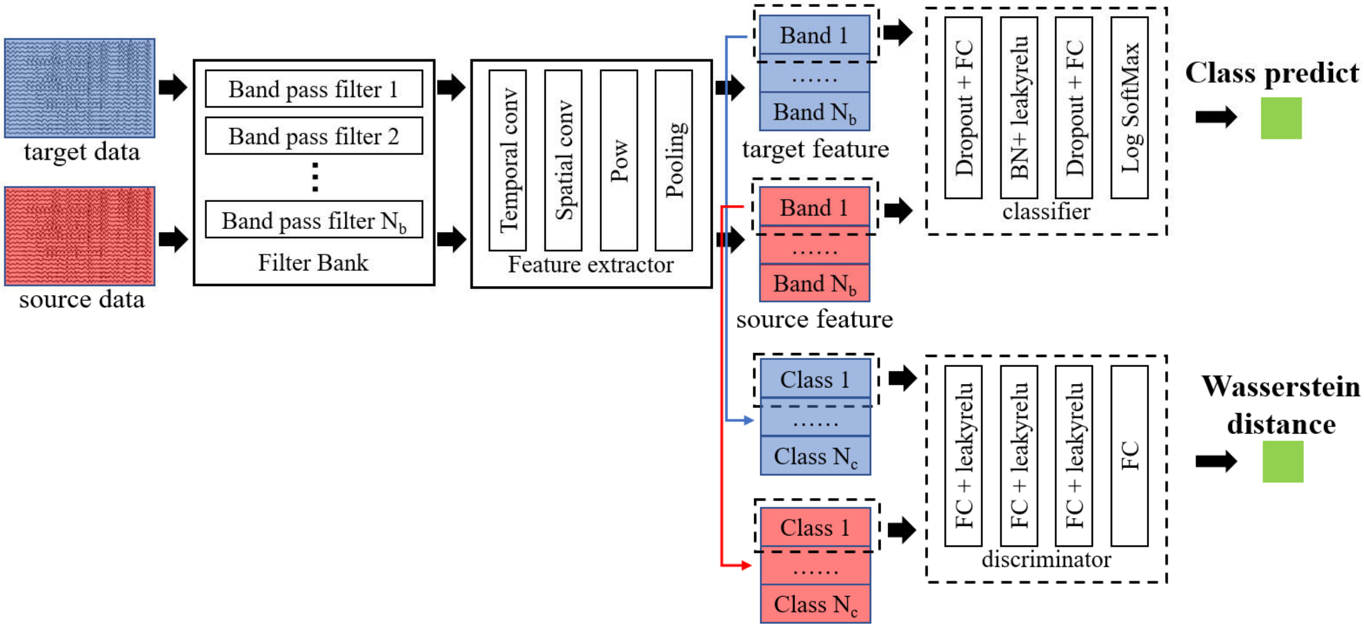

Fig. 1. Proposed Filter Bank Wasserstein Adversarial Domain Adaptation framework. Data from target and source are first separately feed into filter bank which outputs multiband signals. Multiband signals are then put into feature extractor to generate multiband features. A classifier takes feature from each band and outputs class predictions for each band. Multiband predictions are ensembled by averaging predict probability from each band. For features from each band each class, a unique discriminator is used to predict the Wasserstein distance between target feature batch and source feature batch. Adversarial training is adopted that the discriminator tries to maximize the output distance while the feature extractor tries to minimize the output distance. By adversarial training, feature extractor is forced to extract common feature for target and source and thus we could use source data to assist model training for target data.

predict which domain the current sample comes from. Feature extractor tries to find features that could not be distinguished by the discriminator. In other word, the feature extractor will learn to find a common feature space for the target and the source domain, where the distribution of two domain are similar. Traditional discriminator with cross entropy loss could hardly work when there is little or none overlap between two distribution. Considering there are only a few samples from the target domain and the distribution gap between EEG from different subject is large, our discriminator estimates the Wasserstein distance between target and source domain instead of predict domain label. Wasserstein distance could reveal the distribution difference even when the two distribution have no overlap. The discriminator we proposed has four fully connected layers. Leaky ReLU is selected as the activate function. A batch of features from two domain is taken as the input. The output of the discriminator is a scalar that indicates the Wasserstein distance between two domains of the input batch. To further enhance the predict accuracy of the proposed model and align the conditional distribution between target and source, each class has a unique discriminator. That is to say, the discriminator narrows the distribution between the two domains within the same class.

C. *Loss function*

For each band, we have one feature extractor, one classifier and N_{class} discriminators. During training, data from different bands are separately trained. For a test sample comes from class c_i, the classifier of $band_i$ outputs the possibility $p_{c_i}^{band_i}$. Possibility of one sample that comes from class c_i is averaged between all bands $p_{c_i} = \frac{1}{N_{band}} \sum p_{c_i}^{band_i}$.

For classifier C, the loss function is:

$$\min_{F,C} \mathcal{L}_C = -\mathbb{E}_{x,y \sim D^t \cup D^s} \log p_{model}(y|x) \tag{1}$$

Where $p_{model}(y|x)$ is the predicted probability of y given x.

For discriminator, the following loss function is maximized to estimate the Wasserstein distance between two domains:

$$\max_{D} \mathcal{L}_D = \mathbb{E}_{x^t \sim D^t}[D(F(x^t))] - \mathbb{E}_{x^s \sim D^s}[D(F(x^s))] \tag{2}$$

The feature extractor, on the contrary, try to minimize the Wasserstein distance between two domains:

$$\min_{F} \mathcal{L}_{adv} = \mathbb{E}_{x^t \sim D^t}[D(F(x^t))] - \mathbb{E}_{x^s \sim D^s}[D(F(x^s))] \tag{3}$$

The final loss function for the feature extractor and the classifier is:

$$\min_{F,C} \mathcal{L} = w_C \mathcal{L}_C + w_{adv} \mathcal{L}_{adv} \tag{4}$$

Where w_C and w_{adv} is the weight for classification loss and adversarial training loss.

D. *Training pipeline*

In training process, discriminator is alternatively trained with feature extractor and classifier. In other word, in each training epoch, we first train D for n_d iterations, then train F and C for n_c iterations. Our training pipeline is summarized in algorithm 1.

Algorithm 1. The training process of the proposed framework

Input: training data from target and source domain D^s, D^t, maximum training epoch n_{epoch}, number of iterations for training classifier and feature extractor per epoch n_c, number of iterations for training discriminator per epoch n_d, batch size m

Output: feature extractor F, classifier C, discriminator $D_1,\ldots, D_{N_{class}}$

initialize $F, C, D_1,\ldots, D_{N_{class}}$, with parameters $\theta_F, \theta_C, \theta_{D_1},\ldots\theta_{D_{N_{class}}}$

for t = 1, …, n_{epoch}:

  for i_{class} = 0,1,…, N_{class}:

    for t_d = 1, …, n_d:

      sample a batch $\{(x_i^t, y_i^t)\}_{i=1}^m$ from target class i_{class}

      sample a batch $\{(x_j^s, y_j^s)\}_{j=1}^m$ from source class i_{class}

      $g_{\theta_{D_{i_{class}}}} \leftarrow \nabla_{\theta_{D_{i_{class}}}}[\frac{1}{m}\sum_{i=1}^m D_{i_{class}}(F(x_i^t)) - \frac{1}{m}\sum_{j=1}^m D_{i_{class}}(F(x_j^s))]$

      $\theta_{D_{i_{class}}} \leftarrow \theta_{D_{i_{class}}} + w_D \cdot \mathrm{RMSProp}(\theta_{D_{i_{class}}}, g_{\theta_{D_{i_{class}}}})$

    end for

  end for

  for t_c = 1, …, n_c:

    sample a batch $\{(x_i^t, y_i^t)\}_{i=1}^m$ from target

    sample a batch $\{(x_j^s, y_j^s)\}_{j=1}^m$ from source

    $g_{\theta_F} \leftarrow -\nabla_{\theta_F}[\frac{1}{m}\sum_{i=1}^m \log p_{model}(y_i^t|x_i^t) + \frac{1}{m}\sum_{j=1}^m \log p_{model}(y_j^s|x_j^s)]$

    for i_{class} = 0,1,…, N_{class}:

      $g_{\theta_F} \leftarrow g_{\theta_F} - \nabla_{\theta_F}[\frac{1}{m}\sum_{i=1}^m D_{i_{class}}(F(x_i^t)) - \frac{1}{m}\sum_{j=1}^m D_{i_{class}}(F(x_j^s))]$

    end for

    $g_{\theta_C} \leftarrow -\nabla_{\theta_C}[\frac{1}{m}\sum_{i=1}^m \log p_{model}(y_i^t|x_i^t) + \frac{1}{m}\sum_{j=1}^m \log p_{model}(y_j^s|x_j^s)]$

    $\theta_F \leftarrow \theta_F + w_F \cdot \mathrm{Adam}(\theta_F, g_{\theta_F}), \theta_C \leftarrow \theta_C + w_C \cdot \mathrm{Adam}(\theta_C, g_{\theta_C})$

  end for

end for

E. *Source selection*

Source selection is important for our transfer approach. Inappropriate source subject may bring negative transfer, which will in turn damage the model performance. Therefore, we propose a source selection method.

We choose source subject base on two criterions. First, the source distribution should be close to target distribution. In this case, it would be easier to pulling closer the features from two domain. Second, source data should provide useful information for classification. That is to say, inter-class distribution difference of source subject should be large. In other words, source data itself should has high classification accuracy.

For each available source subject, we divide the source data into training set and evaluation set. We pretrain a classification model on the training set and test it on the evaluation set. The source subjects are then ranked according to their evaluation accuracy. The first half of the source subjects are taken as optional subjects. For each target subject, we evaluate all pretrained models of optional subjects on the target training set. The subject with highest classification accuracy is finally selected for the adversarial training.

F. *Cropped training*

To effectively train the neural network with limited training samples, cropped training strategy is adopted. Cropped training use cropped samples generate by a sliding window to train the model and is widely adopted by deep learning based MI decoding models. Concretely, for each training sample with shape of $C \times T$, a sliding window of length 500, stride 10 is used to crop original sample into N_{slice} slices. In this study, N_{slice} is 63. Thus, one EEG block with 48 training samples results in 48×63 = 3024 cropped slices. For each original training sample, the predict result is generated by taking average of all cropped slices.

G. *Implementation details*

Implementation details are given in Table I. We use Adam optimizer for classification and RMSprop optimizer for adversarial training.

TABLE I. IMPLEMENTATION DETAILS

Modules	layers	params
Feature extractor	Convolution	(1,25)×40, stride=1
	Convolution	(N_c,1)×40
	Pow	
	pool	(1,75), stride=15
Classifier	Fc	32, batch norm, leaky Relu
	Fc	4, SoftMax
Discriminator	Fc	64, leaky Relu
	Fc	32, leaky Relu
	Fc	16, leaky Relu
	Fc	1

III. EXPERIMENT

A. *Dataset*

Our method is evaluated on Data set 2a of BCI competition IV[19] which contains EEG data of 9 subjects. Two sessions on different days were collected. Each session includes 288 trails. There are four different motor imagery tasks (left hand, right hand, both feet, and tongue). Each class contains 72 trails in one session. 22 EEG channels were recorded at sampling rate 250Hz.

B. *Pre-process*

EEG data is preprocessed to improve the signal to noise ratio. We follow a general preprocess pipeline as described by R. T. Schirrmeister[15]. Raw data is band passed to 4-38Hz by a FIR filter. An exponential moving average method with a decay factor of 0.999 is then applied. Time segment from 0.5s before to 4s after the onset of the visual cue were used.

C. *Experiment Settings*

We evaluated the methods on two sessions separately. For each session, we first randomly separate the target data into six blocks and choose one block as training set and the rest blocks as test set. One other subject is selected as source subject and all of the data from source subject is used for training. We repeat this procedure for five times and report the average result. To fully evaluate the model performance when trained with different amount of data, we gradually increase the training data, use two, three, four, five of target data blocks to train the model and rest of the blocks to test.

IV. RESULT AND DISCUSSION

We evaluated our method and different existing methods on dataset IIa of BCI Competition IV. Table II shows the classification accuracy of models under different amount of training data from the target subject in session 1 of dataset IIa. And Table III shows that in session 2. Each column represents one training scheme that use a certain amount of data as train set. In each column, method with the highest accuracy is highlighted in boldface. Also, existing comparing methods with the highest accuracy is highlighted in boldface. State-of-the-art methods are selected as comparing methods. CSP and FBCSP is also compared, although these two methods are quite old, their performance is still competitive. Among comparing methods, CSP and FBCSP is realized by adopting codes from MNE toolbox.[20] EEGNet[16], shallow and deep CNN[15] is realized by adopting the original source code provided by the authors. C2CM[17], which we don't have access to the source code, is re-implemented following the original paper.

Results in Table II and Table III show that our method achieves the best performance under each amount of training data. And , classification accuracy of our method improves with the increase of training data. Moreover, performance of our method trained with one and two block of data is better than the best comparing method trained with two and three blocks separately. For session 1, our method trained with three blocks of data gets better result than the best comparing method trained with five blocks. And for session 2, our method trained with three blocks of data gets better result than the best comparing method trained with four blocks. Therefore, our method could reduce the need for training data for at least one block.

Result shows that CSP and FBCSP have a good performance with one and two blocks of training data. in contrast, Deep learning methods is not robust enough when only few training samples are available. It indicates that CSP is a robust feature extraction method for MI classification and holds the performance only when there is only a few block of training data. This may be because that CSP method has lower structural risk.

Table II and Table III show that when training data increased, deep learning methods show advantages against traditional methods. ShallowCNN gains better accuracy compared with CSP and FBCSP with five blocks of training data. It reveals that neural networks hold the potential to work better when provided with enough data. Our method based on transfer learning utilizes source data to offset the limited training samples. Even given one block of target training data, we could also achieve a better accuracy.

TABLE II CLASSIFICATION ACCURACY OF DIFFERENT ALGORITHMS WITH DIFFERENT NUMBER OF TRAINING BLOCKS ON SESSION 1 OF DATA SET IIA OF BCI COMPETITION IV (IN PERCENTAGE %)

method	1	2	3	4	5
CSP[13]	**54.64**	57.92	60.22	60.79	60.92
FBCSP[14]	51.53	60.01	**67.38**	68.15	70.28
EEGNet[16]	31.46	42.43	49.58	54.79	63.01
shallowCNN[15]	51.94	60.05	66.56	**69.98**	**71.02**
deepCNN[15]	43.72	53.47	60.00	63.56	67.50
C2CM[17]	51.90	**60.87**	65.43	68.12	69.91
ours	**63.98**	**68.83**	**72.53**	**74.38**	**75.74**

TABLE III CLASSIFICATION ACCURACY OF DIFFERENT ALGORITHMS WITH DIFFERENT NUMBER OF TRAINING BLOCKS ON SESSION 2 OF DATA SET IIA OF BCI COMPETITION IV (IN PERCENTAGE %)

method	1	2	3	4	5
CSP[13]	**60.00**	62.33	62.53	63.40	63.47
FBCSP[14]	56.19	**65.85**	**69.09**	70.00	72.18
EEGNet[16]	33.32	46.44	54.88	58.27	64.40
shallowCNN[15]	54.50	62.53	67.67	70.58	**73.47**
deepCNN[15]	44.53	56.23	62.98	65.90	68.89
C2CM[17]	54.52	64.27	68.38	**72.18**	72.68
ours	**65.68**	**69.77**	**72.96**	**74.65**	**78.38**

Besides, we notice that all methods in session 2 generally gets better performance compared with in session 1. This may be due to the influence of subject experience. Session 2 is collected on a different day after session 1. Skill of motor imagery of subjects may be improved.

In order to evaluate the effectiveness of utilizing source data and proposed filter bank strategy and domain adaptation framework, we conduct the following experiments on session 1. Table IV shows the classification accuracy with or without source data, filter bank strategy and adversarial loss. Here, Data from Session 1 is again randomly separated into 6 blocks and 2 blocks is selected as train set while rest blocks are test set. This procedure is conducted five times. Adversarial loss is not used while comparing the effectiveness of source data.

In Table IV, performance of model trained with source data without adversarial training is similar to model trained with only target data (66.38%, 66.92%). It indicates that the distribution difference between the target data and the source data is large. Directly using source data and only two blocks of target data

TABLE IV PERFORMANCE OF OUR METHOD WITH OR WITHOUT SOURCE DATA AND FILTER BANK AND ADVERSARIAL LOSS (IN PERCENTAGE %)

		t1	t2	t3	t4	t5	t6	t7	t8	t9	mean
w/o SRC	o	78.02	54.06	75.94	56.67	54.27	46.35	83.96	83.02	70.00	66.92
	w	79.58	47.81	79.17	53.12	53.44	46.25	84.79	83.13	70.10	66.38
w/o FB	o	73.75	48.96	81.46	51.67	50.21	45.73	81.46	78.02	70.21	64.61
	w	79.69	53.54	79.90	55.94	58.44	48.75	86.56	84.27	72.40	68.83
w/o ADV	o	79.58	47.81	79.17	53.12	53.44	46.25	84.79	83.13	70.10	66.38
	w	79.69	53.54	79.90	55.94	58.44	48.75	86.56	84.27	72.40	68.83

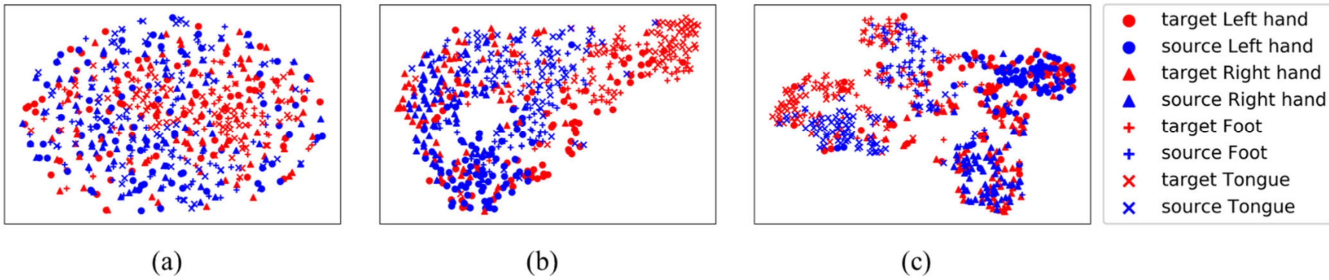

Fig. 2. Visualization of (a) raw data, (b) feature extracted by our neural network trained without adversarial loss (c) feature extracted by our neural network trained with adversarial loss.

cannot enhance model performance. It may even cause negative transfer.

Accuracy of model without filter bank is 64.61%. With filter bank, model accuracy enhanced to 68.83%. It shows that filter bank could effectively increase the model performance. This increase is benefit from multi band ability to grasp multi band information and the integration of multi band predictions.

Without the adversarial loss, the model classification accuracy is 66.38%, which is lower than model trained with target data only. Adding the adversarial loss, the model classification accuracy increased to 68.83%. It shows that our adversarial domain adaptation framework based on WGAN could effectively align target and source distribution.

To further demonstrate the significance of our adversarial domain adaptation framework, we applied t-distributed stochastic neighbor embedding (t-SNE)[21] to visualize the data distribution. Fig. 2 shows an example from target subject 1 and the corresponding source subject 7. Fig. 2 (a) is the distribution of raw data. Fig. 2 (b) is the distribution of feature extracted by model trained without adversarial training. Fig. 2 (c) is the distribution of feature extracted by model trained with our proposed adversarial training framework.

Result shows that raw EEG data don't have obvious clusters. After extracting feature thought feature extractor, distribution of target and source feature is separated into clusters, which could be seen in Fig. 2 (b) and (c). In Fig. 2 (b), without adversarial training, inter-class distribution is close. Besides, there exist a distribution gap between two domains. In Fig. 2 (c), after adding the adversarial training approach, distribution from different class is pushed away from each other. And the distribution gap between domains is closer. These results show that our adversarial training framework can effectively align the target and source distribution.

V. CONCLUSION

In this paper, we proposed a filter bank Wasserstein adversarial domain adaptation framework to reduce the need for calibration data and improve classification accuracy for MI-BCI. We design a CNN based feature extractor to extract feature from EEG data in MI tasks. Filter bank strategy is employed to deal with sensitive frequency band difference across subject. It extracts feature from multiple sub bands and integrates predictions from all sub bands. WGAN based domain adaptation network is used to extract common feature from target and source. The adversarial training is separately applied to each class to close the intra-class cross-domain distribution difference. Experiment results shows that our method achieves best performance given same amount of training data compared with existing methods. Besides, performance of our method trained with certain blocks of data is similar to or better than the best comparing method trained with one more block. This indicate that our method could reduce calibration time by saving on block training data.

ACKNOWLEDGMENT

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC2001302, in part by the National Natural Science Foundation of China under 61976209, Grant 61906188, and Grant 62006014 and in part by the Strategic Priority Research Program of CAS under Grant XDB32040200. (Corresponding author: Huiguang He).

REFERENCES

[1] J. R. Wolpaw *et al.*, "Brain-computer interface technology: a review of the first international meeting," *IEEE Trans Rehabil Eng,* vol. 8, no. 2, pp. 164-73, Jun 2000, doi: 10.1109/tre.2000.847807.

[2] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain–computer interfaces for communication and control," *Clinical neurophysiology,* vol. 113, no. 6, pp. 767-791, 2002.

[3] A. Zimmermann-Schlatter, C. Schuster, M. A. Puhan, E. Siekierka, and J. Steurer, "Efficacy of motor imagery in post-stroke rehabilitation: a systematic review," *Journal of NeuroEngineering and Rehabilitation,* vol. 5, no. 1, p. 8, 2008, doi: 10.1186/1743-0003-5-8.

[4] J. J. Shih, D. J. Krusienski, and J. R. Wolpaw, "Brain-Computer Interfaces in Medicine," *Mayo Clinic Proceedings,* vol. 87, no. 3, pp. 268-279, 2012, doi: 10.1016/j.mayocp.2011.12.008.

[5] O. A. Mokienko, L. A. Chernikova, A. A. Frolov, and P. D. Bobrov, "Motor Imagery and Its Practical Application," *Neuroscience and Behavioral Physiology,* vol. 44, no. 5, pp. 483-489, 2014, doi: 10.1007/s11055-014-9937-y.

[6] J. Meng, S. Zhang, A. Bekyo, J. Olsoe, B. Baxter, and B. He, "Noninvasive Electroencephalogram Based Control of a Robotic Arm for Reach and Grasp Tasks," *Sci Rep,* vol. 6, p. 38565, Dec 14 2016, doi: 10.1038/srep38565.

[7] D. Coyle, J. Garcia, A. R. Satti, and T. M. McGinnity, "EEG-based continuous control of a game using a 3 channel motor imagery BCI: BCI game," in *2011 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB)*, 2011: IEEE, pp. 1-7.

[8] L. Bonnet, F. Lotte, and A. Lécuyer, "Two brains, one game: design and evaluation of a multiuser BCI video game based on motor imagery," *IEEE Transactions on Computational Intelligence and AI in games,* vol. 5, no. 2, pp. 185-198, 2013.

[9] J. Asensio-Cubero, J. Q. Gan, and R. Palaniappan, "Multiresolution analysis over graphs for a motor imagery based online BCI game," *Computers in biology and medicine,* vol. 68, pp. 21-26, 2016.

[10] N. Kosmyna, F. Tarpin-Bernard, N. Bonnefond, and B. Rivet, "Feasibility of BCI control in a realistic smart home environment," *Frontiers in human neuroscience,* vol. 10, p. 416, 2016.

[11] A. A. Jais, W. Mansor, K. Y. Lee, and W. Fauzi, "Motor imagery EEG analysis for home appliance control," in *2017 IEEE 13th International Colloquium on Signal Processing & its Applications (CSPA)*, 2017: IEEE, pp. 314-317.

[12] H.-J. Kim, M.-H. Lee, and M. Lee, "A BCI based smart home system combined with event-related potentials and speech imagery task," in *2020 8th International Winter Conference on Brain-Computer Interface (BCI)*, 2020: IEEE, pp. 1-6.

[13] 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.

[14] 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, p. 39, 2012, doi: 10.3389/fnins.2012.00039.

[15] R. T. Schirrmeister *et al.*, "Deep learning with convolutional neural networks for EEG decoding and visualization," *Hum Brain Mapp,* vol. 38, no. 11, pp. 5391-5420, Nov 2017, doi: 10.1002/hbm.23730.

[16] 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," *J Neural Eng,* vol. 15, no. 5, p. 056013, Oct 2018, doi: 10.1088/1741-2552/aace8c.

[17] S. Sakhavi, C. Guan, and S. Yan, "Learning Temporal Information for Brain-Computer Interface Using Convolutional Neural Networks," *IEEE Trans Neural Netw Learn Syst,* vol. 29, no. 11, pp. 5619-5629, Nov 2018, doi: 10.1109/TNNLS.2018.2789927.

[18] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering,* vol. 22, no. 10, pp. 1345-1359, 2009.

[19] M. Tangermann *et al.*, "Review of the BCI competition IV," *Frontiers in neuroscience,* vol. 6, p. 55, 2012.

[20] A. Gramfort *et al.*, "MEG and EEG data analysis with MNE-Python," *Frontiers in neuroscience,* vol. 7, p. 267, 2013.

[21] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research,* vol. 9, no. 11, 2008.