2021 43rd Annual International Conference of the
IEEE Engineering in Medicine & Biology Society (EMBC)
Oct 31 - Nov 4, 2021. Virtual Conference

Improving Transfer Performance of Deep Learning with Adaptive Batch Normalization for Brain-computer Interfaces*

Lichao Xu[1,2], Zhen Ma[1], Jiayuan Meng[1], *Student Member, IEEE*, Minpeng Xu[1,2,†], *Member, IEEE*, Tzyy-Ping Jung[1,2,3], *Fellow, IEEE*, and Dong Ming[1,2], *Senior Member, IEEE*

Abstract— **Recently, transfer learning and deep learning have been introduced to solve intra- and inter-subject variability problems in Brain-Computer Interfaces. However, the generalization ability of these BCIs is still to be further verified in a cross-dataset scenario. This study compared the transfer performance of manifold embedded knowledge transfer and pre-trained EEGNet with three preprocessing strategies. This study also introduced AdaBN for target domain adaptation. The results showed that EEGNet with Riemannian alignment and AdaBN could achieve the best transfer accuracy about 65.6% on the target dataset. This study may provide new insights into the design of transfer neural networks for BCIs by separating source and target batch normalization layers in the domain adaptation process.**

I. INTRODUCTION

Electroencephalography (EEG)-based Brain-computer Interfaces (BCIs) enable the human brain to control machines without any physical intervention [1]. EEG-based BCIs have been used in many ways, e.g., quadcopter control [2], stroke rehabilitation [3], and glaucoma detection [4]. However, intra- and inter-subject variability hinders practical usages of BCIs due to EEG signals are non-stationary, easily contaminated by noise signals and vary across different subjects or across time within the same subject [5]. A calibration stage is usually required to collect enough training data for rebuilding a subject-specific model before the beginning of each session, which is inconvenient and time-consuming.

Recently, deep learning and transfer learning have been introduced to the BCI community for alleviating the negative impacts of cross-subject variability [6]. ShallowConvNet [7] and EEGNet [8], both imitate temporal and spatial filters in FBCSP [9], have shown the cross-subject generalization ability without fine-tuning. Zhang et al. proposed a manifold embedded knowledge transfer (MEKT) method [10] by fusing the transfer component analysis (TCA) [11] and the joint distribution adaptation (JDA) [12], which outperforms several transfer learning approaches for BCI datasets.

However, current deep learning and transfer learning algorithms are usually validated in single datasets. It assumes that data of all subjects are acquired under the same conditions, e.g. the same acquisition system and lab environment, which are generally violated in the cross-dataset scenario. For this problem, Chiang et al. proposed a least-squares transformation (LST)-based transfer learning method that significantly improved the decoding accuracy of steady-state visually evoked potentials (SSVEP) across devices [13]. In our previous work [14], we have also shown that the cross-dataset variability would weaken the generalizability of deep learning models across motor imagery (MI) datasets. A possible solution to alleviate the impact of cross-dataset variability is to apply the whitening procedure [15] for each subject. This procedure would align the covariance centroid of each subject and is often used as a preprocessing step in many transfer learning algorithms [10], [16].

This study further investigates the generalization performance of deep learning models across datasets. We validated the performance of EEGNet with three preprocessing strategies, namely channel normalization, trial normalization, and Riemannian alignment. This study also introduced Adaptive Batch Normalization(AdaBN) [17]. The results show that the Riemannian aligning and AdaBN could easily improve the generalizability of EEGNet without fine-tuning.

The organization of the rest of the work is as follows. Section II describes the datasets and methods used in this work. Section III presents the study results and discussions. Finally, Section IV concludes this study.

II. METHODS

A. Datasets

This study used the EEG data from two public datasets BNCI2014001 [18] and PhysioNetMI [19]. Both datasets were downloaded using the MOABB package [20]. Table I lists the details of the datasets. This study selected twenty-two common channels between both datasets. Only the left-hand and right-hand MI classes were included in our analysis. Raw data were filtered with an FIR bandpass filter of 4-40Hz and zero-phase forward-and-reverse filtering was implemented using raw.filter() function in MNE [21]. Each trial was 3s in length and downsampled to 128Hz such that the size was 22×384. Each trial has been centered by subtracting its mean vector of each channel. BNCI2014001

*This work was supported by National Key Research and Development Program of China, grant number 2017YFB1300300; National Natural Science Foundation of China, grant number 81925020, 61976152, 81671861; Young Elite Scientist Sponsorship Program by CAST, grant number 2018QNRC001.

[1] Department of Biomedical Engineering, College of Precision Instruments and Optoelectronics Engineering, Tianjin University, Tianjin, China, 300072

[2] Academy of Medical Engineering and Translational Medicine, Tianjin University, Tianjin, China, 300072

[3] Swartz Center for Computational Neuroscience, University of California, San Diego, CA 92093 USA

[†] Corresponding Author: minpeng.xu@tju.edu.cn

L. Xu and Z. Ma contributed equally to the work.

978-1-7281-1178-0/21/$31.00 ©2021 IEEE

TABLE I

DETAILS OF DATASETS

Dataset	Classes	Subjects	Trial Duration(s)	Channels	Sampling Rate(Hz)	Number of trials
BNCI2014001	left/right/feet/tongue	9	4	22	250	2592
PhysionetMI	left/right/hands/feet	109	3	64	250	4918

TABLE II

EEGNET ARCHITECTURE

Layer	Input Size	Output Size	Channel	Kernel	Stride	Padding	Constraint
Conv2d	$1 \times 22 \times 384$	$8 \times 22 \times 384$	8	(1, 64)	(1, 1)	SAME	
BatchNorm2d	$8 \times 22 \times 384$	$8 \times 22 \times 384$					
Depthwise Conv2d	$8 \times 22 \times 384$	$16 \times 1 \times 384$	16	(22, 1)	(1, 1)		MaxNormConstraint(1)
BatchNorm2d	$16 \times 1 \times 384$	$16 \times 1 \times 384$					
Elu	$16 \times 1 \times 384$	$16 \times 1 \times 384$					
AvgPool2d	$16 \times 1 \times 384$	$16 \times 1 \times 96$		(1, 4)	(1, 4)		
Dropout	$16 \times 1 \times 96$	$16 \times 1 \times 96$					
Separable Conv2d	$16 \times 1 \times 96$	$16 \times 1 \times 96$	16	(1, 16)	(1, 1)	SAME	
BatchNorm2d	$16 \times 1 \times 96$	$16 \times 1 \times 96$					
Elu	$16 \times 1 \times 96$	$16 \times 1 \times 96$					
AvgPool2d	$16 \times 1 \times 96$	$16 \times 1 \times 12$		(1, 8)	(1, 8)		
Dropout	$16 \times 1 \times 12$	$16 \times 1 \times 12$					
Flatten	$16 \times 1 \times 12$	132					
Linear	132	2					MaxNormConstraint(0.25)

was used as the source dataset and PhyisoNetMI was used as the target dataset.

For evaluating the performance of models, trials in the source dataset were randomly split into training, validation and test sets by a 5-fold stratified sampler with preserving the percentage of trials for each class in each subject. The test set was 20% of the available data. 20% of the remaining data were referred to as the validation set and the rest of them were training set. The training and validation sets of the source dataset were used for training MEKT and EEGNet. The test set of the source dataset was used for evaluating the within-subject classification. Data of each subject in the target dataset were kept for the cross-dataset classification.

B. MEKT

This work used MEKT as the baseline algorithm. MEKT first aligns the covariance matrices of EEG trials in the Riemannian manifold and then extracts tangent space features. Domain adaptation is performed to map source and target features to a new space such that the difference between the joint probability distributions of the source and target domains is minimized while preserving their geometric structures. A linear discriminant analysis (LDA) classifier is trained with the transformed source features and applied to the transformed target features to predict their labels. Parameters of the MEKT were set to default as the original authors proposed [10]. More details of MEKT can be found in [10] and https://github.com/chamwen/MEKT.

C. EEGNet

Table II lists the architecture of EEGNet. This architecture is the latest version of the original authors proposed. Our EEGNet model was implemented in PyTorch framework [22]. The optimizer was Adam with the learning rate set to 0.001 and the batch size was 256 in the source dataset. We trained the model for 200 epochs and selected the best model on the validation set for further analyses. More details of EEGNet can be found in [8] and https://github.com/vlawhern/arl-eegmodels.

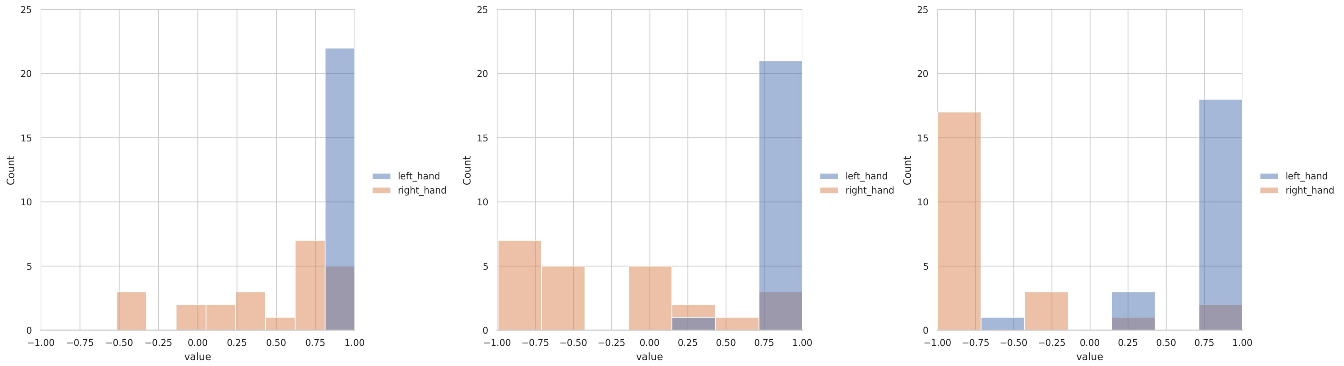
D. Preprocessing strategies

Three preprocessing strategies were compared in this work:

- Channel normalization. In this strategy, normalization was implemented in each channel. It is calculated by dividing the standard deviation of each channel. The model trained with this strategy was named EEGNet-cnorm.
- Trial normalization. Trial normalization was implemented in each trial instead of each channel. It is calculated by dividing its standard deviation of the flattened array. Compared to channel normalization, trial normalization would keep the relative magnitude of channels. The model trained with this strategy was named EEGNet-tnorm.
- Riemannian alignment. In this strategy, each trial was whitened by a reference matrix M for each subject. The transformation is as follows:

$$\hat{E}_i = M^{-1/2} E_i \tag{1}$$

where E_i is the EEG data matrix of the i-th trial. The reference matrix M is the Riemannian mean of covariance matrices of all trials for each subject. The model trained with this strategy was named EEGNet-rie.



(a) EEGNet

(b) EEGNet-rie

(c) EEGNet-rie-AdaBN

Fig. 1. Distributions of the difference between the softmax output of the last layer of EEGNet of a typical target subject. (a) distribution of the EEGNet. (b) distribution of the EEGNet with Riemannian alignment. (c) distribution of the EEGNet with Riemannian alignment and AdaBN.

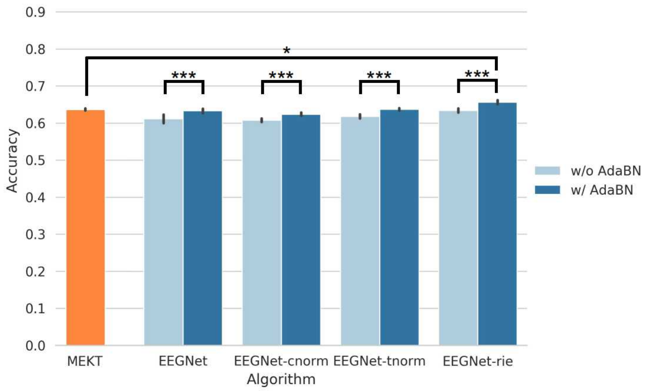


Fig. 2. Classification results of models with AdaBN (w/ AdaBN) or without AdaBN (w/o AdaBN) on the target dataset. * indicates $p < 0.05$ and *** indicates $p < 0.001$.

E. AdaBN

The idea behind AdaBN is simple. It hypothesizes that label-related information is stored in the weights of each layer whereas domain-related information is represented by the statistics of the Batch Normalization (BN). Therefore, it would be easy to transfer the pre-trained model to the target domain by modulating the statistics in the BN layer [17]. In this work, the running mean and variance parameters of the BN layers were replaced by the mean and variance of data of the target subject. Meanwhile, other parameters of pre-trained models were fixed. More details of AdaBN can be found in [17].

III. Results and Discussion

A. Within-subject Classification Results

The accuracy of EEGNet on the test set of the source dataset was 0.795(0.012). The accuracy of EEGNet-cnom and EEGNet-tnorm was 0.808(0.017) and 0.807(0.017), respectively. No significance was found between these three models. However, the accuracy of EEGNet-rie was 0.877(0.011), which was significantly better than that of others (paired sample T-test, $p < 0.05$).

The improvement of performance may be owned to more consistent distributions of inputs. Riemannian alignment not only eliminates the difference in the magnitudes of subjects' data, but also compensates for covariate shift which describes the difference in distributions of inputs. Due to the congruance invariance of the Riemannian distance, while the distances between trials of the same subject remain unchanged, the centroids of all subjects are centered at the identity matrix [15]. Deep learning models with Riemannian alignment can converge more easily and find the common features of all subjects.

B. Cross-dataset Classification Results

Fig.1 shows the distributions of the difference between the softmax output of the last layer of EEGNet for a typical target subject. Fig.1(a) is the distribution of the naive EEGNet model, which indicates that the decision boundary was about 0.75 instead of the normal zero. The decision boundary bias could be eliminated with Riemannian alignment and AdaBN, as shown in Fig.1(b) and Fig.1(c).

Fig.2 shows the accuracies of the pre-trained models on the target dataset. The accuracy of MEKT was 0.637(0.003). The direct transfer accuracy of EEGNet was about 0.611(0.012). The accuracy of EEGNet-cnorm and EEGNet-tnorm was 0.608(0.005) and 0.618(0.006), respectively. EEGNet-tnorm was slightly better than EEGNet-cnorm and EEGNet, but no significance was found between them. The accuracy of EEGNet-rie was 0.634(0.005), which was significantly better than that of other preprocessing strategies (paired sample T-test, $p < 0.05$). No significance was found between EEGNet-rie and MEKT. All preprocessing strategies including naive EEGNet could be significantly improved with AdaBN as shown in Fig.2 (paired sample T-test, $p < 0.001$). The best transfer performance was 0.656(0.006) for the EEGNet-rie model with AdaBN, which was significantly better than that of MEKT (paired sample T-test, $p < 0.05$).

The results of cross-dataset classification further show that Riemannian alignment is better than other preprocessing strategies. Although channel normalization and trial normalization can elimiate the difference in magnitudes of the inputs, they can not correct covariate shift of each subject. Moreover, an unreasonable normalization strategy may weaken the generalizability of models. Fig.2 shows the accuracy of EEGNet-tnorm was slightly better than that of EEGNet-cnorm. Trial normalization can keep the relative magnitudes of channels, which may be useful in the training of models.

The BN layer is originally designed to alleviate the issue of internal covariate shift, which is the change in the distribution of network activations due to the change in network parameters during training [23]. The BN layer makes the input distribution of each layer remains unchanged during the training process and accelerates the convergence of models. However, the BN layer still assumes that the source domain and the target domain have the same distribution, which is usually violated due to high variability in EEG data. Improper parameters of BN layers would cause the distribution of outputs to deviate further from the expected distribution, even with Riemannian alignment preprocessing strategy. To solve this problem, AdaBN replaces the parameters in the BN layers with parameters computed from the data of the target domain. Fig.1 shows how the distribution of outputs changes with different methods. Fig.1(a) shows that the decision boundary of naive EEGNet was shifted to about 0.75 instead of the normal zero, meaning that the distribution of outputs is highly biased due to the negative impacts of covariate shift and internal covariate shift. Covariate shift could be corrected with Riemannian alignment, as shown in Fig.1(b), and internal covariate shift could be further alleviated with AdaBN, as shown in Fig.1(c).

IV. CONCLUSIONS

This study evaluated the performance of EEGNet models with different preprocessing strategies for the cross-dataset scenario. This study also introduced AdaBN for better transfer performance. The results showed that EEGNet with Riemannian alignment and AdaBN could easily achieve better transfer performance without fine-tuning. This study may provide new insights into the design of transfer neural networks for BCIs by separating source and target batch normalization layers in the domain adaptation process.

REFERENCES

[1] Jonathan R Wolpaw, Niels Birbaumer, Dennis J McFarland, Gert Pfurtscheller, and Theresa M Vaughan. Brain–computer interfaces for communication and control. *Clinical neurophysiology*, 113(6):767–791, 2002.

[2] Karl LaFleur, Kaitlin Cassady, Alexander Doud, Kaleb Shades, Eitan Rogin, and Bin He. Quadcopter control in three-dimensional space using a noninvasive motor imagery-based brain–computer interface. *Journal of neural engineering*, 10(4):046003, 2013.

[3] Fabricio A Jure, Lucía C Carrere, Gerardo G Gentiletti, and Carolina B Tabernig. Bci-fes system for neuro-rehabilitation of stroke patients. In *J Phys Conf Ser*, volume 705, page 012058, 2016.

[4] Masaki Nakanishi, Yu-Te Wang, Tzyy-Ping Jung, John K Zao, Yu-Yi Chien, Alberto Diniz-Filho, Fabio B Daga, Yuan-Pin Lin, Yijun Wang, and Felipe A Medeiros. Detecting glaucoma with a portable brain-computer interface for objective assessment of visual function loss. *JAMA ophthalmology*, 135(6):550–557, 2017.

[5] Dongrui Wu, Yifan Xu, and Bao-Liang Lu. Transfer learning for eeg-based brain-computer interfaces: A review of progress made since 2016. *IEEE Transactions on Cognitive and Developmental Systems*, 2020.

[6] Xiaotong Gu, Zehong Cao, Alireza Jolfaei, Peng Xu, Dongrui Wu, Tzyy-Ping Jung, and Chin-Teng Lin. Eeg-based brain-computer interfaces (bcis): A survey of recent studies on signal sensing technologies and computational intelligence approaches and their applications. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2021.

[7] Robin Tibor Schirrmeister, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and Tonio Ball. Deep learning with convolutional neural networks for eeg decoding and visualization. *Human brain mapping*, 38(11):5391–5420, 2017.

[8] Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. *Journal of neural engineering*, 15(5):056013, 2018.

[9] Kai Keng Ang, Zheng 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)*, pages 2390–2397. IEEE, 2008.

[10] Wen Zhang and Dongrui Wu. Manifold embedded knowledge transfer for brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 28(5):1117–1127, 2020.

[11] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2010.

[12] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE international conference on computer vision*, pages 2200–2207, 2013.

[13] Kuan-Jung Chiang, Chun-Shu Wei, Masaki Nakanishi, and Tzyy-Ping Jung. Boosting template-based ssvep decoding by cross-domain transfer learning. *Journal of Neural Engineering*, 2020.

[14] Lichao Xu, Minpeng Xu, Yufeng Ke, Xingwei An, Shuang Liu, and Dong Ming. Cross-dataset variability problem in eeg decoding with deep learning. *Frontiers in human neuroscience*, 14, 2020.

[15] Paolo Zanini, Marco Congedo, Christian Jutten, Salem Said, and Yannick Berthoumieu. Transfer learning: A riemannian geometry framework with applications to brain–computer interfaces. *IEEE Transactions on Biomedical Engineering*, 65(5):1107–1116, 2017.

[16] Pedro Luiz Coelho Rodrigues, Christian Jutten, and Marco Congedo. Riemannian procrustes analysis: transfer learning for brain–computer interfaces. *IEEE Transactions on Biomedical Engineering*, 66(8):2390–2401, 2018.

[17] Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, 80:109–117, 2018.

[18] Michael Tangermann, Klaus-Robert Müller, Ad Aertsen, Niels Birbaumer, Christoph Braun, Clemens Brunner, Robert Leeb, Carsten Mehring, Kai J Miller, Gernot Mueller-Putz, et al. Review of the bci competition iv. *Frontiers in neuroscience*, 6:55, 2012.

[19] Gerwin Schalk, Dennis J McFarland, Thilo Hinterberger, Niels Birbaumer, and Jonathan R Wolpaw. Bci2000: a general-purpose brain-computer interface (bci) system. *IEEE Transactions on biomedical engineering*, 51(6):1034–1043, 2004.

[20] Vinay Jayaram and Alexandre Barachant. MOABB: trustworthy algorithm benchmarking for BCIs. *Journal of Neural Engineering*, 15(6):066011, sep 2018.

[21] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A Engemann, Daniel Strohmeier, Christian Brodbeck, Roman Goj, Mainak Jas, Teon Brooks, Lauri Parkkonen, et al. Meg and eeg data analysis with mne-python. *Frontiers in neuroscience*, 7:267, 2013.

[22] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[23] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.