Neural Networks 153 (2022) 235–253

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Review

Transfer learning for motor imagery based brain–computer interfaces: A tutorial

Dongrui Wu*, Xue Jiang, Ruimin Peng

Key Laboratory of the Ministry of Education for Image Processing and Intelligent Control, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China

ARTICLE INFO

Article history:
Received 5 January 2022
Received in revised form 22 April 2022
Accepted 6 June 2022
Available online 14 June 2022

Keywords:
Brain–computer interface
Electroencephalogram
Transfer learning
Euclidean alignment
Motor imagery

ABSTRACT

A brain–computer interface (BCI) enables a user to communicate directly with an external device, e.g., a computer, using brain signals. It can be used to research, map, assist, augment, or repair human cognitive or sensory–motor functions. A closed-loop BCI system performs signal acquisition, temporal filtering, spatial filtering, feature engineering and classification, before sending out the control signal to an external device. Transfer learning (TL) has been widely used in motor imagery (MI) based BCIs to reduce the calibration effort for a new subject, greatly increasing their utility. This tutorial describes how TL can be considered in as many components of a BCI system as possible, and introduces a complete TL pipeline for MI-based BCIs. Examples on two MI datasets demonstrated the advantages of considering TL in multiple components of MI-based BCIs. Especially, integrating data alignment and sophisticated TL approaches can significantly improve the classification performance, and hence greatly reduces the calibration effort.

Contents

* Corresponding author.
E-mail addresses: drwu@hust.edu.cn (D. Wu), xuejiang@hust.edu.cn (X. Jiang), rmpeng2019@hust.edu.cn (R. Peng).

https://doi.org/10.1016/j.neunet.2022.06.008
0893-6080/

1. Introduction

A brain–computer interface (BCI) (Lance, Kerick, Ries, Oie, & McDowell, 2012; Rao, 2013) enables a user to communicate directly with an external device, e.g., a computer, using his/her brain signals. It can be used to research, map, assist, augment, or repair human cognitive or sensory–motor functions (Krucoff, Rahimpour, Slutzky, Edgerton, & Turner, 2016), benefiting both patients (Pfurtscheller, Müller-Putz, Scherer, & Neuper, 2008) and able-bodied people (van Erp, Lotte, & Tangermann, 2012; Nicolas-Alonso & Gomez-Gil, 2012). For example, Willett, Avansino, Hochberg, Henderson, and Shenoy (2021) "*developed an intracortical BCI that decodes attempted handwriting movements from neural activity in the motor cortex and translates it to text in real time.*" A hand-paralyzed participant can use this BCI to type at the speed of "*90 characters per minute with 94.1% raw accuracy online, and greater than 99% accuracy offline with a general-purpose autocorrect.*" By monitoring the neural response of a driver in real-time, a BCI can also detect the fatigue/drowsiness level of the driver, and hence help prevent accidents (Cui, Xu, & Wu, 2019).

Based on how close the electrodes are to the brain tissue, BCIs can be categorized into non-invasive, partially invasive, and invasive ones (Martini et al., 2020; Rao, 2013). The last two need to open the skull to place the sensors, so they are mainly used in clinics. Non-invasive BCIs are preferred by able-bodied consumers, due to their low risk. Though different types of signals, e.g., electroencephalography (EEG) (Gu et al., 2021), magnetoencephalography (Mellinger et al., 2007), functional magnetic resonance imaging (Sitaram et al., 2007; Wang & Wu, 2017), functional near-infrared spectroscopy (Naseer & Hong, 2015), etc., have been used in non-invasive BCIs, EEG is currently the most popular one, due to its convenience and low cost.

There are three classical paradigms of EEG-based non-invasive BCIs: motor imagery (MI) (He, Baxter, Edelman, Cline, & Ye, 2015; Pfurtscheller & Neuper, 2001), P300 event-related potentials (Sellers & Donchin, 2006), and steady-state visual evoked potentials (Bin, Gao, Wang, Hong, & Gao, 2009). The first is the focus of this tutorial.

In an MI-based BCI, the user imagines the movement of his/her body parts (e.g., left hand, right hand, both feet, or tongue), which modulates different areas of the motor cortex of the brain, e.g., top-left for right-hand MI, top-right for left-hand MI, and top-central for feet MI. A classification algorithm can then be used to decode the recorded EEG signals and map the corresponding MI to a command for the external device. MI-based BCIs have been widely used in robotic device control (Edelman et al., 2019), hand orthosis control (Pfurtscheller & Neuper, 2001), post-stroke rehabilitation (Zimmermann-Schlatter, Schuster, Puhan, Siekierka, & Steurer, 2008), and so on.

The flowchart of a closed-loop MI-based BCI system is shown in Fig. 1. It consists of the following main components (Wu, Xu and Lu, 2022):

1. *Signal acquisition*, which uses a headset to collect EEG signal from the scalp, while the user is performing MI tasks.

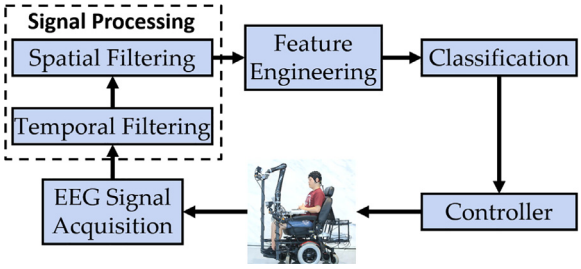

Fig. 1. A closed-loop MI-based BCI system, which usually consists of EEG signal acquisition, signal processing (usually includes temporal filtering and spatial filtering), feature engineering, classification, and controller blocks, and an external device such as a wheelchair.

2. *Signal processing* (Lotte, 2015). Because EEG signals are weak, and easily contaminated by artifacts and interferences, e.g., muscle movements, eye blinks, heartbeats, powerline noise, etc., sophisticated signal processing approaches must be used to increase the signal-to-noise ratio. Both temporal filtering and spatial filtering are usually performed. Temporal filtering may include notch filtering to remove the 50 Hz or 60 Hz powerline interference, and then bandpass filtering, e.g., [8, 30] Hz, to remove DC drift and high frequency noise. Spatial filters (Wu, King, Chuang, Lin, & Jung, 2018) include the basic ones, e.g., common average Ref. Teplan (2002), Laplacian filters (Lagerlund, Sharbrough, & Busacker, 1997), principal component analysis (Jolliffe, 2002), etc., and more sophisticated ones, e.g., independent component analysis (Delorme & Makeig, 2004), xDAWN (Rivet, Souloumiac, Attina, & Gibert, 2009), canonical correlation analysis (Roy, Bonnet, Charbonnier, Jallon, & Campagne, 2015), common spatial patterns (CSP) (Ramoser, Muller-Gerking, & Pfurtscheller, 2000), etc.
3. *Feature engineering*, which includes feature extraction, and sometimes also feature selection. Time domain, frequency domain, time–frequency domain, Riemannian space (Yger, Berar, & Lotte, 2017), and/or topoplot features (Lin et al., 2021) could be used.
4. *Classification* (Lotte et al., 2018), which uses a machine learning algorithm to decode the EEG signal from the extracted features. Commonly used classifiers include linear discriminant analysis (LDA) and support vector machine (SVM).
5. *Controller*, which sends a command to an external device, e.g., a wheelchair, according to the decoded EEG signal.

EEG signals usually have large inter-subject variability (individual differences). A t-distributed stochastic neighbor embedding (t-SNE) (van der Maaten & Hinton, 2008) visualization of the CSP features [computed by Eq. (8)] of EEG trials from the seven subjects in Dataset 1 (introduced in Section 3.1) is shown in the left panel of Fig. 2. Clearly, EEG trials from different subjects may not overlap at all. Because of the inter-subject variability, and also the non-stationarity of EEG signals, an MI-based BCI usually needs a long calibration session for a new subject, from

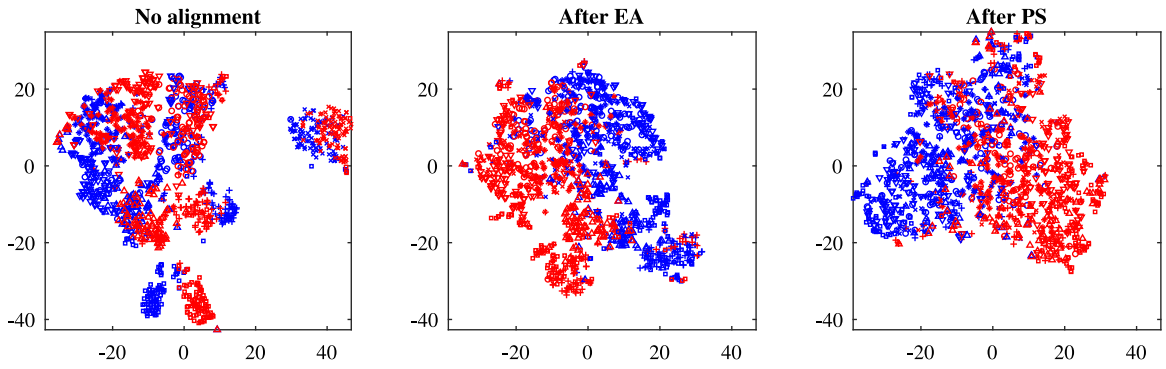

Fig. 2. T-SNE visualization of the CSP features of EEG trials from the seven subjects in Dataset 1. EEG trials from different subjects are denoted by different shapes, and different colors represent different classes. Without data alignment, EEG trials from different subjects may not overlap at all. After EA or PS alignment, the data distributions become more consistent. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

20–30 min (Saha, Ahmed, Mostafa, Hadjileontiadis, & Khandoker, 2018) to hours or even days. This lengthy calibration significantly reduces the utility of BCI systems. Hence, many sophisticated signal processing and machine learning approaches have been proposed recently to reduce or eliminate the calibration (Dai, Zheng, Liu, & Zhang, 2018; He & Wu, 2020a, 2020b; Jayaram, Alamgir, Altun, Scholkopf, & Grosse-Wentrup, 2016; Rodrigues, Jutten, & Congedo, 2019; Wang, Lu, Zhang, & Tang, 2015; Wu, 2017; Wu, Xu and Lu, 2022; Zanini, Congedo, Jutten, Said, & Berthoumieu, 2018; Zhang & Wu, 2020).

One of the most promising such approaches is transfer learning (TL) (Pan & Yang, 2010; Wu, Xu and Lu, 2022), which uses data/knowledge from source domains (existing subjects) to help the calibration in the target domain (new subject). TL has been used in various components of the flowchart shown in Fig. 1. For example, to consider TL in spatial filtering, Dai et al. (2018) proposed transfer kernel CSP to integrate kernel CSP (Albalawi & Song, 2012) and transfer kernel learning (Long, Wang, Sun, & Philip, 2015) for EEG trial filtering. To consider TL in feature engineering, Chen, Zhang, and Lou (2019) extended ReliefF (Kononenko, 1994) and minimum redundancy maximum relevancy (mRMR) (Peng, Long, & Ding, 2005) feature selection approaches to class-separate and domain-fused (CSDF)-ReliefF and CSDF-mRMR, which optimizes both the class separability and the domain similarity simultaneously. They then further integrated CSDF-ReliefF and CSDF-mRMR with an adaptation regularization-based TL classifier (Long, Wang, Ding, Pan, & Yu, 2014). To consider TL in the classifier, Jayaram et al. (2016) proposed a multi-task learning (which is a subfield of TL) framework for cross-subject MI classification.

This tutorial introduces typical algorithms used in MI-based BCIs, describes how TL can be considered in as many components of a BCI system as possible, and introduces a complete TL pipeline for MI-based BCIs, shown in Fig. 3:

1. *Temporal filtering*, where band-pass filtering is performed on both the source and target domain data.
2. *Data alignment*, which aligns EEG trials from the source domains and the target domain so that their distributions are more consistent. This is a new component, which does not exist in Fig. 1, but will greatly facilitate TL in sequential components, as shown later in this tutorial.
3. *Spatial filtering*, where TL can be used to design better spatial filters, especially when the amount of target domain labeled data is small.
4. *Feature engineering*, where TL may be used to extract or select more informative features.
5. *Classification*, where TL can be used to design better classifiers or regression models, especially when there are no or very few target domain labeled data.

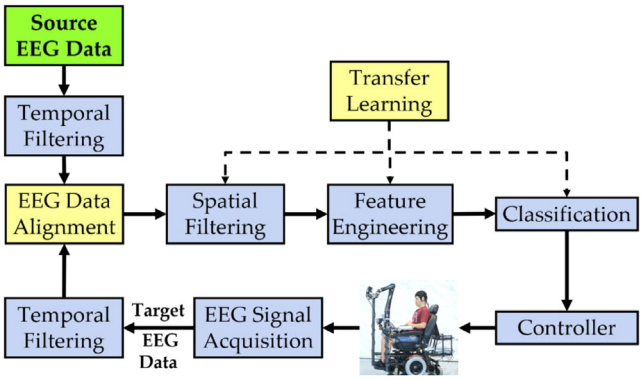

Fig. 3. A complete TL pipeline for closed-loop MI-based BCIs. EA should be performed between temporal filtering and spatial filtering. TL can be used in spatial filtering, feature engineering, and classification.

We introduce some representative TL approaches in data alignment, spatial filtering, feature selection and classification, and demonstrate using three MI datasets that incorporating TL in all these components can indeed achieve better classification performance than not using TL, or using TL in only a subset of these components.

Our main contributions are:

1. We introduce a complete TL pipeline for closed-loop MI-based BCI systems, as shown in Fig. 3, and point out that explicitly including a data alignment component before spatial filtering is very important to the TL performance, for both traditional machine learning and deep learning, both offline and online classification, and both cross-subject and cross-session classification.
2. We demonstrated through experiments that usually considering TL in more components in Fig. 3 can result in better classification performance, and more sophisticated TL approaches are usually more beneficial than simple TL approaches, or not using TL at all.
3. We publicize our Matlab source code at https://github.com/drwuHUST/TLBCI, which should be very helpful to BCI researchers, especially those new to this area.

The remainder of this tutorial is organized as follows: Section 2 introduces some representative TL approaches at different components of a BCI system. Section 3 demonstrates the performance of the complete TL pipeline in offline and online cross-subject/-session MI classifications. Section 4 draws conclusions and points out some future research directions.

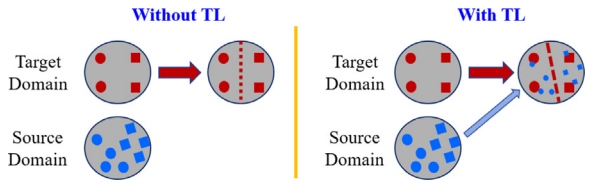

Fig. 4. Illustration of simple TL in classification. Without TL, only the target domain data are used to train a target domain classifier. With TL, both source domain and target domain data are used, but the source domain data may be assigned smaller weights.

2. TL approaches

This section introduces the basic concepts of TL, and how TL could be used in data alignment, spatial filtering, feature selection and classification of MI-based BCIs.

We mainly consider offline binary classification, and would like to use labeled EEG trials from a source subject to help classify trials from a target subject. When there are multiple source subjects, we can combine data from all source subjects and then view that as a single source subject, or perform TL for each source subject separately and then aggregate them. The former is used in this tutorial.

Assume the source subject has N_s labeled samples $\{(X_s^n, y_s^n)\}_{n=1}^{N_s}$, where $X_s^n \in \mathbb{R}^{c\times t}$ is the nth EEG trial and y_s^n the corresponding class label, in which c is the number of EEG channels, and t the number of time domain samples. The target subject has N_l (N_l could be 0) labeled samples $\{(X_t^n, y_t^n)\}_{n=1}^{N_l}$, and N_u unlabeled samples $\{X_t^n\}_{n=N_l+1}^{N_l+N_u}$. In offline calibration, the N_u unlabeled samples are also known to us, and we need to design a classifier to obtain their labels.

2.1. TL

TL (Pan & Yang, 2010) uses data/knowledge from a source domain to help solve a task in a target domain. A domain consists of a feature space $\mathcal{X}$ and its associated marginal probability distribution $P(X)$, i.e., $\{\mathcal{X}, P(X)\}$, where $X \in \mathcal{X}$. Two domains are different if they have different feature spaces, and/or different $P(X)$. A task consists of a label space $\mathcal{Y}$ and a prediction function $f(X)$, i.e., $\{\mathcal{Y}, f(X)\}$. Two tasks are different if they have different label spaces, and/or different conditional probability distributions $P(y|X)$.

For BCI calibration, TL usually means to use labeled EEG trials from an existing subject to help the calibration for a new subject. This tutorial considers the scenario that both subjects have the same feature space and label space, i.e., the subjects wear the same EEG headset and perform the same types of MIs, but different $P(X)$ and $P(y|X)$. This is the most commonly encountered TL scenario in BCI calibration.

A very simple idea of TL in classifier training is illustrated in Fig. 4. Assume the target domain has only four training samples belonging to two classes (represented by different shapes), whereas the source domain has more. Without TL, we can build a classifier in the target domain using only its own four training samples. Since the number of training samples is very small, this classifier is usually unreliable. With TL, we can combine samples from the source domain with those in the target domain to train a classifier. Since the two domains may not be completely consistent, e.g., the marginal probability distributions may be different, we may assign the source domain samples smaller weights than the target domain samples. If optimized properly, the resulting classifier can usually achieve better generalization performance.

Fig. 4 illustrates maybe the simplest TL approach in classification. Similar approaches may also be used in spatial filtering and feature engineering components in Fig. 3. Some of them are introduced next.

2.2. Euclidean alignment (EA)

Due to the inter-subject variability, the marginal probability distributions of the EEG trials from different subjects are usually (significantly) different; so, it is very beneficial to perform data alignment to make different domains more consistent, before other operations in Fig. 3.

Different EEG data alignment approaches have been proposed recently (He & Wu, 2020a, 2020b; Rodrigues et al., 2019; Zanini et al., 2018; Zhang & Wu, 2020). A summary and comparison of them is given in Wu, Xu and Lu (2022). Among them, Euclidean alignment (EA), proposed by He and Wu (2020b) and illustrated in Fig. 5, is easy to perform and completely unsupervised (it does not need any labeled data from any subject). So, it is used as an example in this tutorial.

For the source subject, EA first computes

$$\bar{R}_s = \frac{1}{N_s}\sum_{n=1}^{N_s} X_s^n \left(X_s^n\right)^\top, \tag{1}$$

i.e., the Euclidean arithmetic mean of all spatial covariance matrices from the source subject, then performs the alignment by

$$\tilde{X}_s^n = \bar{R}_s^{-1/2} X_s^n. \tag{2}$$

Similarly, for the target subject, EA computes the arithmetic mean of all N_l+N_u spatial covariance matrices and then performs the alignment.

After EA, the aligned EEG trials are whitened (Zhang & Wu, 2020), and their mean spatial covariance matrix from each subject equals the identity matrix (He & Wu, 2020b); hence, the distributions of EEG trials from different subjects become more consistent (an example of the seven subjects in Dataset 1 after EA is shown in the middle panel of Fig. 2). This will greatly benefit TL in subsequent steps.

2.3. Pre-alignment strategy (PS)

Xu et al. (2020) proposed an online pre-alignment strategy (PS) to match the distributions from different domains. An example of the seven subjects in Dataset 1 after PS is shown in the right panel of Fig. 2. When used in offline classification, its formula is essentially identical to Eq. (2), except that the Riemannian mean (Pennec, Fillard, & Ayache, 2006) instead of the Euclidian mean is used in computing $\bar{R}_s$.

The *Riemannian distance* $\delta(R, R^n)$ between two symmetric positive-definite covariance matrices $R \in \mathbb{R}^{c\times c}$ and $R^n \in \mathbb{R}^{c\times c}$ is the minimum length of a curve connecting them on the Riemannian manifold, computed as Moakher (2005):

$$\delta(R, R^n) = \left\|\log\left(R^{-1}R^n\right)\right\|_F = \left[\sum_{i=1}^{c}\log^2\lambda_i\right]^{\frac{1}{2}}, \tag{3}$$

where the subscript $_F$ denotes the Frobenius norm, and λ_i, $i = 1, \ldots, c$, are the real eigenvalues of $R^{-1}R^n$.

The *Riemannian mean* (Pennec et al., 2006) of N covariance matrices is defined as the matrix minimizing the sum of the squared Riemannian distances, i.e.,

$$\bar{R} = \arg\min_R \sum_{n=1}^{N}\delta^2(R, R^n). \tag{4}$$

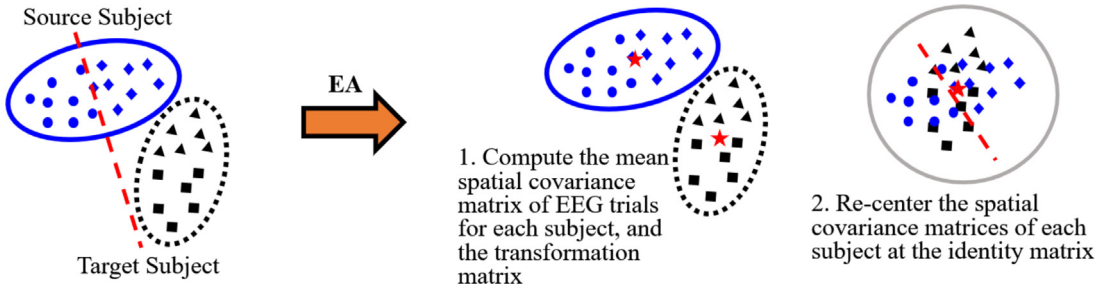

Fig. 5. EA for aligning EEG trials from different subjects (domains). EEG trials from different domains become more consistent after EA, which benefits TL.

The Riemannian mean does not have a closed-form solution, but can be computed by an iterative gradient descent algorithm (Fletcher & Joshi, 2004).

The characteristics of PS are almost identical to EA, i.e., it is completely unsupervised, and aligns the EEG trials directly, except that its computational cost is higher than EA, as the Riemannian mean does not have a closed-form solution.

2.4. CSP

CSP (Blankertz, Tomioka, Lemm, Kawanabe, & Muller, 2008; Ramoser et al., 2000) performs supervised spatial filtering for EEG trials, aiming to find a set of spatial filters to maximize the ratio of variance between two classes.

The traditional CSP uses data from the target subject only. For Class $k \in \{-1, 1\}$, CSP tries to find a spatial filter matrix $W_k^* \in \mathbb{R}^{c\times f}$, where f is the number of spatial filters, to maximize the variance ratio between Class k and Class $-k$:

$$W_k^* = \arg\max_{W\in\mathbb{R}^{c\times f}} \frac{\mathrm{tr}(W^\top \bar{C}_t^k W)}{\mathrm{tr}(W^\top \bar{C}_t^{-k} W)}, \tag{5}$$

where $\bar{C}_t^k \in \mathbb{R}^{c\times c}$ is the mean spatial covariance matrix of the N_l labeled EEG trials in Class k, and tr the trace of a matrix. The solution W_k^* is the concatenation of the f leading eigenvectors of $(\bar{C}_t^{-k})^{-1}\bar{C}_t^k$.

Then, CSP concatenates the $2f$ spatial filters from both classes to obtain the complete filter matrix:

$$W^* = \left[W_{-1}^*, W_1^*\right] \in \mathbb{R}^{c\times 2f}, \tag{6}$$

and computes the spatially filtered X_t^n by:

$$\tilde{X}_t^n = W^{*\top} X_t^n \in \mathbb{R}^{2f\times t}. \tag{7}$$

Finally, the log-variances of $\tilde{X}_t^n$ can be extracted as features $\boldsymbol{x}_t^n \in \mathbb{R}^{1\times 2f}$ in later classification:

$$\boldsymbol{x}_t^n = \log\left(\frac{\mathrm{diag}\left(\tilde{X}_t^n \left(\tilde{X}_t^n\right)^\top\right)}{\mathrm{tr}\left(\tilde{X}_t^n \left(\tilde{X}_t^n\right)^\top\right)}\right), \tag{8}$$

where diag means the diagonal elements of a matrix, and log is the logarithm operator.

2.5. Combined CSP (CCSP)

Because the target subject has very few labeled samples, i.e., N_l is small, W^* computed above may not be reliable. The source domain samples can be used to improve W^*.

In the combined CSP (CCSP), we simply concatenate the N_s source domain labeled samples and N_l target domain labeled samples to compute W^*. Note that all samples have the same weight, i.e., source domain and target domain samples are treated equally.

CCSP may be the simplest TL-based CSP approach.

2.6. Regularized CSP (RCSP)

Regularized CSP (RCSP) (Lu, Eng, Guan, Plataniotis, & Venetsanopoulos, 2010) was specifically proposed to handle the problem that the target domain has very few labeled samples. Though the original paper did not mention TL, it actually used the idea of TL.

RCSP computes the regularized average spatial covariance matrix for Class k as:

$$\hat{C}^k(\beta,\gamma) = (1-\gamma)\hat{C}^k(\beta) + \frac{\gamma}{c}\mathrm{tr}\left(\hat{C}^k(\beta)\right) I, \tag{9}$$

where β and γ are two parameters in $[0, 1]$, $I \in \mathbb{R}^{c\times c}$ is an identity matrix, and

$$\hat{C}^k(\beta) = \frac{\beta N_l \bar{C}_t^k + (1-\beta)N_s \bar{C}_s^k}{\beta N_l + (1-\beta)N_s}. \tag{10}$$

$\hat{C}^k(\beta,\gamma)$ can then be used to replace $\bar{C}_t^k$ in Eq. (5) to compute the CSP filter matrix.

Note that when $\beta = 1$ and $\gamma = 0$, RCSP becomes the traditional CSP. When $\beta = 0.5$ and $\gamma = 0$, RCSP becomes CCSP.

2.7. ReliefF

ReliefF (Kononenko, 1994) is a classical feature selection approach. Next we introduce its basic idea for binary classification.

Let x_i be the ith feature, whose importance $w(x_i)$ is initialized to 0. ReliefF randomly selects a sample $\boldsymbol{x}$, and finds its k ($k = 10$ in this tutorial) nearest neighbors $H = \{\boldsymbol{h}_j\}_{j=1}^k$ in the same class, and also k nearest neighbors $M = \{\boldsymbol{m}_j\}_{j=1}^k$ in the other class. It then updates $w(x_i)$ by

$$\begin{aligned} w(x_i) = w(x_i) &- \frac{1}{k}\sum_{j=1}^{k}\mathrm{diff}(x_i, \boldsymbol{x}, \boldsymbol{h}_j) \\ &+ \frac{1}{k}\sum_{j=1}^{k}\mathrm{diff}(x_i, \boldsymbol{x}, \boldsymbol{m}_j), \end{aligned} \tag{11}$$

where $\mathrm{diff}(x_i, \boldsymbol{x}, \boldsymbol{x}')$ denotes the difference between samples $\boldsymbol{x}$ and $\boldsymbol{x}'$ in terms of feature x_i. Eq. (11) is very intuitive: the importance of x_i should be decreased with its ability to discriminate samples from the same class [$\mathrm{diff}(x_i, \boldsymbol{x}, \boldsymbol{h})$], and increased with its ability to discriminate samples from different classes [$\mathrm{diff}(x_i, \boldsymbol{x}, \boldsymbol{m})$].

In this tutorial, ReliefF terminates after 100 iterations, i.e., 100 randomly selected $\boldsymbol{x}$ were used to compute the final $w(x_i)$. We then rank these $w(x_i)$ and select a few features corresponding to the largest $w(x_i)$.

2.8. Combined ReliefF (CReliefF)

The traditional ReliefF selects $\boldsymbol{x}$, H and M from only the target domain labeled samples. We propose a very simple TL extension of ReliefF, combined ReliefF (CReliefF), by selecting $\boldsymbol{x}$, H and M from labeled samples in both domains.

2.9. LDA

LDA is a popular linear classifier for binary classification. It assumes that the feature covariance matrices (not to be confused with the spatial covariance matrix of an EEG trial) from the two classes have full rank and are both equal to Σ_t. The classification for a new input $\boldsymbol{x}$ is then

$$\text{sign}\left(\boldsymbol{w}\boldsymbol{x}^\top - \theta\right), \tag{12}$$

where

$$\boldsymbol{w} = \Sigma_t^{-1}(\bar{\boldsymbol{x}}_{t,1} - \bar{\boldsymbol{x}}_{t,-1}), \tag{13}$$

$$\boldsymbol{\theta} = \frac{1}{2}\boldsymbol{w}(\bar{\boldsymbol{x}}_{t,1} + \bar{\boldsymbol{x}}_{t,-1}), \tag{14}$$

in which $\bar{\boldsymbol{x}}_{t,-1}$ and $\bar{\boldsymbol{x}}_{t,1}$ are the mean feature vector of Class -1 and Class 1 computed from the N_l target domain labeled samples, respectively.

2.10. Combined LDA (CLDA)

When N_l is small, the above LDA classifier may not be reliable. The combined LDA (CLDA) is a simple TL approach, which concatenates labeled samples from both the source domain and the target domain to train an LDA classifier. All samples from both domains are treated equally.

2.11. Weighted adaptation regularization (wAR)

Wu (2017) proposed weighted adaptation regularization (wAR), a TL approach for offline cross-subject EEG classification. Though the original experiments were conducted for event-related potential classification, wAR can also be used for MI classification.

wAR learns a classifier f^* by minimizing the following regularized loss function:

$$\begin{aligned} f^* = \arg\min_f & \sum_{n=1}^{N_s} w_s^n \ell(f(\boldsymbol{x}_s^n), y_s^n) \\ & + w_t \sum_{n=1}^{N_l} w_t^n \ell(f(\boldsymbol{x}_t^n), y_t^n) \\ & + \lambda_1 \|f\|_K^2 + \lambda_2 D_{f,K}(P_s(\boldsymbol{x}_s), P_t(\boldsymbol{x}_t)) \\ & + \lambda_3 D_{f,K}(P_s(\boldsymbol{x}_s|y_s), P_t(\boldsymbol{x}_t|y_t)), \end{aligned} \tag{15}$$

where ℓ is the classification loss, w_t is the overall weight of samples from the target subject, w_s^n and w_t^n are the weights for the nth sample from the source subject and the target subject, respectively, K is a kernel function, $P_s(\boldsymbol{x}_s)$ and $P_t(\boldsymbol{x}_t)$ are the marginal probability distributions of features from the source subject and the target subject, respectively, $P_s(\boldsymbol{x}_s|y_s)$ and $P_t(\boldsymbol{x}_t|y_t)$ are the conditional probability distributions from the source subject and the target subject, respectively, and λ_1, λ_2 and λ_3 are non-negative regularization parameters.

Briefly speaking, the five terms in Eq. (15) minimize the classification loss for the source subject, the classification loss for the target subject, the structural risk of the classifier, the distance between the marginal probability distributions of the two subjects, and the distance between the conditional probability distributions of the two subjects, respectively.

Although it looks complicated, Eq. (15) has a closed-form solution when the squared loss $\ell(f(\boldsymbol{x}) - y) = (y - f(\boldsymbol{x}))^2$ is used (Wu, 2017).

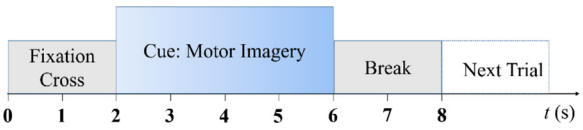

Fig. 6. Timing scheme of the MI tasks in the first two datasets.

2.12. Online wAR (OwAR)

Wu (2017) also proposed online wAR (OwAR), a TL approach for online cross-subject EEG classification.

OwAR learns a classifier f^*, also by minimizing the regularized loss function in Eq. (15). The only difference is that now the kernel matrix K can only be computed from the N_s labeled source domain samples and N_l labeled target domain samples, but not the N_u unlabeled target domain samples, which are unavailable in online classification.

3. Examples

This section demonstrates the offline and online cross-subject and cross-session classification performances of different combinations of TL approaches on three MI datasets. Our Matlab source code is available at https://github.com/drwuHUST/TLBCI.

3.1. MI datasets

Two MI datasets from BCI Competition IV[1] and a third MI dataset from BCI Horizon 2020[2] were used in the examples. They were also used in many previous studies (He & Wu, 2020a, 2020b; Xia, Deng, Duch and Wu, 2022; Zhang & Wu, 2020).

In the first two experiments, the subject sat in front of a computer and performed visual cue based MI tasks, as shown in Fig. 6. A fixation cross on the black screen ($t = 0$) prompted the subject to be prepared, and marked the start of a trial. After two seconds, a visual cue, which was an arrow pointing to a certain direction, was displayed for four seconds, during which the subject performed the instructed MI task. The visual cue disappeared at $t = 6$ s, and the MI also stopped. After a two-second break, the next trial started.

The first dataset[3] (Dataset 1 (Blankertz, Dornhege, Krauledat, Muller, & Curio, 2007)) consisted of seven healthy subjects. Each subject performed two types of MIs, selected from three classes: left-hand, right-hand, and foot. We used the 59-channel EEG data collected from the calibration phase, which included complete marker information. Each subject had 100 trials per class.

The second MI dataset[4] (Dataset 2a) included nine heathy subjects. Each subject performed four different MIs: left-hand, right-hand, both feet, and tongue. We used the 22-channel EEG data and two classes of MIs (left-hand and right-hand) collected from the training phase. Each subject had 72 trials per class.

EEG data preprocessing steps for these two datasets were identical to those in He and Wu (2020b). A causal [8, 30] Hz bandpass filter was used to remove muscle artifacts, powerline noise, and DC drift. Next, we extracted EEG signals between [0.5, 3.5] seconds after the cue onset as our trials for both datasets.

The third MI dataset (Steyrl, Scherer, Förstner, & Müller-Putz, 2014) (Dataset 3) was 002–2014 from BCI Horizon 2020. It includes 14 subjects, each with 100 trials, 50 per class (right hand

[1] http://www.bbci.de/competition/iv/.
[2] http://bnci-horizon-2020.eu/database/data-sets.
[3] http://www.bbci.de/competition/iv/desc_1.html.
[4] http://www.bbci.de/competition/iv/desc_2a.pdf.

Table 1
Summary of the three MI datasets.

Dataset	No. of subjects	No. of channels	No. of trials per subject	No. of classes
1	7	59	200	2
2a	9	22	144	2
3	14	15	100	2

and feet). 15-channel 512 Hz EEG signals were recorded. A causal [8, 35] Hz band-pass filter was applied before we extracted EEG signals between [5, 8] seconds after the cue onset as trials. Note that the data quality of Dataset 3 is worse than the other two datasets, as indicated by the much lower average classification accuracies in the following subsections.

A summary of the three datasets are shown in Table 1.

3.2. Algorithms

We mainly compare the following 18 different algorithms, with various different configurations of TL components:

1. *CSP-LDA*, which uses only the target domain labeled data to design the CSP filters, and then trains an LDA classifier on the target domain labeled data. No source data are used at all, i.e., no TL is used at all.
2. *CSP-CLDA*, which uses only the target domain labeled data to design the CSP filters, and then trains a CLDA classifier by using labeled data from both domains, i.e., only the classifier uses a simple TL approach.
3. *CSP-wAR*, which uses only the target domain labeled data to design the CSP filters, and then trains a wAR classifier by using data from both domains, i.e., only the classifier uses a sophisticated TL approach.
4. *CCSP-LDA*, which concatenates labeled data from both domains to design the CCSP filters, and then trains an LDA classifier on target domain labeled data only, i.e., only spatial filtering uses a simple TL approach.
5. *CCSP-CLDA*, which concatenates labeled data from both domains to design the CCSP filters, and then trains a CLDA classifier also from the concatenated data, i.e., both spatial filtering and classification use a simple TL approach.
6. *CCSP-wAR*, which concatenates labeled data from both domains to design the CCSP filters, and then trains a wAR classifier also from the concatenated data, i.e., spatial filtering uses a simple TL approach, whereas classification uses a sophisticated TL approach.
7. *RCSP-LDA*, which uses labeled data from both domains to design the RCSP filters, and then trains an LDA classifier from the target domain labeled data only, i.e., spatial filtering uses a sophisticated TL approach, whereas classification does not use TL at all.
8. *RCSP-CLDA*, which uses labeled data from both domains to design the RCSP filters, and then trains a CLDA classifier also from this concatenated data, i.e., spatial filtering uses a sophisticated TL approach, whereas classification uses a simple TL approach.
9. *RCSP-wAR*, which uses labeled data from both domains to design the RCSP filters, and then trains a wAR classifier from this concatenated data, i.e., both spatial filtering and classification use a sophisticated TL approach.
10. *EA-CSP-LDA*, which performs EA before CSP-LDA.
11. *EA-CSP-CLDA*, which performs EA before CSP-CLDA, i.e., only the classifier uses a simple TL approach, after EA.
12. *EA-CSP-wAR*, which performs EA before CSP-wAR, i.e., only the classifier uses a sophisticated TL approach, after EA.
13. *EA-CCSP-LDA*, which performs EA before CCSP-LDA, i.e., only spatial filtering uses a simple TL approach, after EA.
14. *EA-CCSP-CLDA*, which performs EA before CCSP-CLDA, i.e., both spatial filtering and classification use a simple TL approach, after EA.
15. *EA-CCSP-wAR*, which performs EA before CCSP-wAR, i.e., spatial filtering uses a simple TL approach, whereas classification uses a sophisticated TL approach, after EA.
16. *EA-RCSP-LDA*, which performs EA before RCSP-LDA, i.e., only spatial filtering uses a sophisticated TL approach, after EA.
17. *EA-RCSP-CLDA*, which performs EA before RCSP-CLDA, i.e., spatial filtering uses a sophisticated TL approach, whereas classification uses a simple TL approach, after EA.
18. *EA-RCSP-wAR*, which performs EA before RCSP-wAR, i.e., both spatial filtering and classification use a sophisticated TL approach, after EA.

Additionally, there were nine PS based approaches, which replace EA in the nine EA based approaches by PS, respectively. A summary of the 27 algorithms is shown in Table 2.

Six (a typical number Rao & Scherer, 2010) spatial filters were used in all CSP algorithms. $\beta = \gamma = 0.1$ were used in RCSP. $w_t = 10$, $\lambda_1 = 0.1$, and $\lambda_2 = \lambda_3 = 10$ were used in wAR, as in Wu (2017) and Wu, Lawhern, Hairston, and Lance (2016), except that w_t was increased from 2 to 10 because the combined source domain has much more labeled samples than the target domain.

By comparing between different pairs of the above algorithms, we can individually understand the effect of TL in different components of Fig. 3.

3.3. Offline experimental settings

For each dataset, we sequentially selected one subject as the target subject and all remaining ones as the source subjects, i.e., we performed cross-subject evaluations. As in He and Wu (2020b), we combined all source subjects as a single source domain, and performed the corresponding TL. This procedure was repeated for each subject, so that each one became the target subject once.

The number of randomly selected labeled samples in the target domain (N_l) increased from zero to $N_l^{\max}$ with a step of 10, where $N_l^{\max} = 100$ on Dataset 1, $N_l^{\max} = 70$ on Dataset 2a, and $N_l^{\max} = 50$ on Dataset 3, i.e., $N_l^{\max}$ was half the number of trials per subject on the corresponding dataset. Because there was randomness involved, we repeated this process 30 times and report the average results.

Note that for algorithms whose spatial filtering component did not involve TL, e.g., those with *CSP-*, when $N_l = 0$, no CSP filters can be trained, and hence no model can be built. All other algorithms used TL in CSP, and hence the source domain labeled data can be used to train the CSP filters even when $N_l = 0$. Similarly, for algorithms whose classifier did not involve TL, e.g., those with *-LDA*, when $N_l = 0$, no LDA classifier can be trained.

Note also that in offline classification, all unlabeled samples in the target domain are known, and can be used in EA, PS and wAR. There is no data leakage here.

The offline cross-subject classification accuracies, averaged over 30 random runs, are shown in Fig. 7. The average performances over all subjects are shown in the last panel of each subfigure. To ensure that the curves are distinguishable, we omitted the curves from the nine PS based algorithms, which were very similar to their EA counterparts.

Table 2
Summary of the 27 algorithms with various degrees of TL.

Algorithm	Data alignment	Spatial filtering		Classifier	
		Simple TL	Sophisticated TL	Simple TL	Sophisticated TL
CSP-LDA	–	–	–	–	–
CSP-CLDA	–	–	–	✓	–
CSP-wAR	–	–	–	–	✓
CCSP-LDA	–	✓	–	–	–
CCSP-CLDA	–	✓	–	✓	–
CCSP-wAR	–	✓	–	–	✓
RCSP-LDA	–	–	✓	–	–
RCSP-CLDA	–	–	✓	✓	–
RCSP-wAR	–	–	✓	–	✓
EA-CSP-LDA	✓	–	–	–	–
EA-CSP-CLDA	✓	–	–	✓	–
EA-CSP-wAR	✓	–	–	–	✓
EA-CCSP-LDA	✓	✓	–	–	–
EA-CCSP-CLDA	✓	✓	–	✓	–
EA-CCSP-wAR	✓	✓	–	–	✓
EA-RCSP-LDA	✓	–	✓	–	–
EA-RCSP-CLDA	✓	–	✓	✓	–
EA-RCSP-wAR	✓	–	✓	–	✓
PS-CSP-LDA	✓	–	–	–	–
PS-CSP-CLDA	✓	–	–	✓	–
PS-CSP-wAR	✓	–	–	–	✓
PS-CCSP-LDA	✓	✓	–	–	–
PS-CCSP-CLDA	✓	✓	–	✓	–
PS-CCSP-wAR	✓	✓	–	–	✓
PS-RCSP-LDA	✓	–	✓	–	–
PS-RCSP-CLDA	✓	–	✓	✓	
PS-RCSP-wAR	✓	–	✓	–	✓

3.4. General effect of TL

In Fig. 7, by comparing CSP-LDA, which did not use TL at all, with the other 17 algorithms, which used simple or sophisticated TL in one or more components of Fig. 3, we can see that when N_l was small, TL almost always resulted in better performance (especially on Datasets 1 and 2a), no matter how much TL was used. However, when N_l increased, CSP-LDA gradually outperformed certain simple TL approaches, e.g., CSP-CLDA and CCSP-CLDA, whereas sophisticated TL approaches, e.g., EA-RCSP-wAR, almost always outperformed CSP-LDA. These results suggest that sophisticated TL may always be beneficial.

To quantitatively demonstrate the general effect of TL, we computed the mean classification accuracies of the 27 approaches when N_l increased from 10 to $N_l^{\max}$ (we did not use $N_l = 0$ because certain approaches did not work in this case), and compared them with that of CSP-LDA. The results are shown in Table 3. We also performed paired t-tests to verify if the performance improvements over CSP-LDA were statistically significant ($\alpha = 0.05$), and marked the insignificant ones by an underline. Table 3 confirms again that generally more sophisticated TL approaches achieved larger performance improvements, especially on Datasets 1 and 2a. The results on Dataset 3 were not as consistent as those on the first two datasets, because Dataset 3 has worse quality, as evidenced from its much lower classification accuracies.

Since generally EA outperformed PS (especially on Dataset 3), and EA can be performed much faster than PS, EA will be emphasized in the remaining of this tutorial.

3.5. Effect of data alignment

In Fig. 7, comparing algorithms without EA and their counterparts with EA, e.g., CSP-CLDA and EA-CSP-CLDA, we can observe that every EA version almost always outperformed its non-EA counterpart. Similar observations can also be made for PS. These results suggest that a data alignment approach such as EA or PS should always be included as a TL preprocessing step in MI-based BCIs.

To quantitatively verify the above conclusion, we also show the mean classification accuracies of algorithms without and with EA/PS in Table 4. Clearly, EA and PS significantly improved the classification accuracies when TL is used in at least one component of spatial filtering and classification, especially on Datasets 1 and 3.

Interestingly, when CSP-LDA was used, i.e., no TL was used at all in spatial filtering and classification, EA/PS slightly reduced the classification performance. We were not able to find an explanation for this; however, when no TL will be performed, there is no point to align EEG trials from different subjects. So, this negative transfer will not happen in practice, and should not be a concern.

3.6. Effect of TL in spatial filtering

In Fig. 7, comparing algorithms without TL in spatial filtering (CSP), with simple TL in spatial filtering (CCSP), and with sophisticated TL in spatial filtering (RCSP), e.g., CSP-wAR, CCSP-wAR and RCSP-wAR, we can observe that:

1. Simple TL in spatial filtering may not always work, e.g., CCSP-wAR had worse performance than CSP-wAR on Dataset 1, but better performance on Dataset 2a.
2. Sophisticated TL in spatial filtering may not always be beneficial either, e.g., RCSP-wAR had better performance than CSP-wAR on Dataset 2a, but worse performance on Dataset 3.

To quantitatively verify the above conclusion, we also show the mean classification accuracies of algorithms without and with TL in spatial filtering in Table 5. EA-RCSP (sophisticated TL in spatial filtering) always outperformed the corresponding EA-CSP

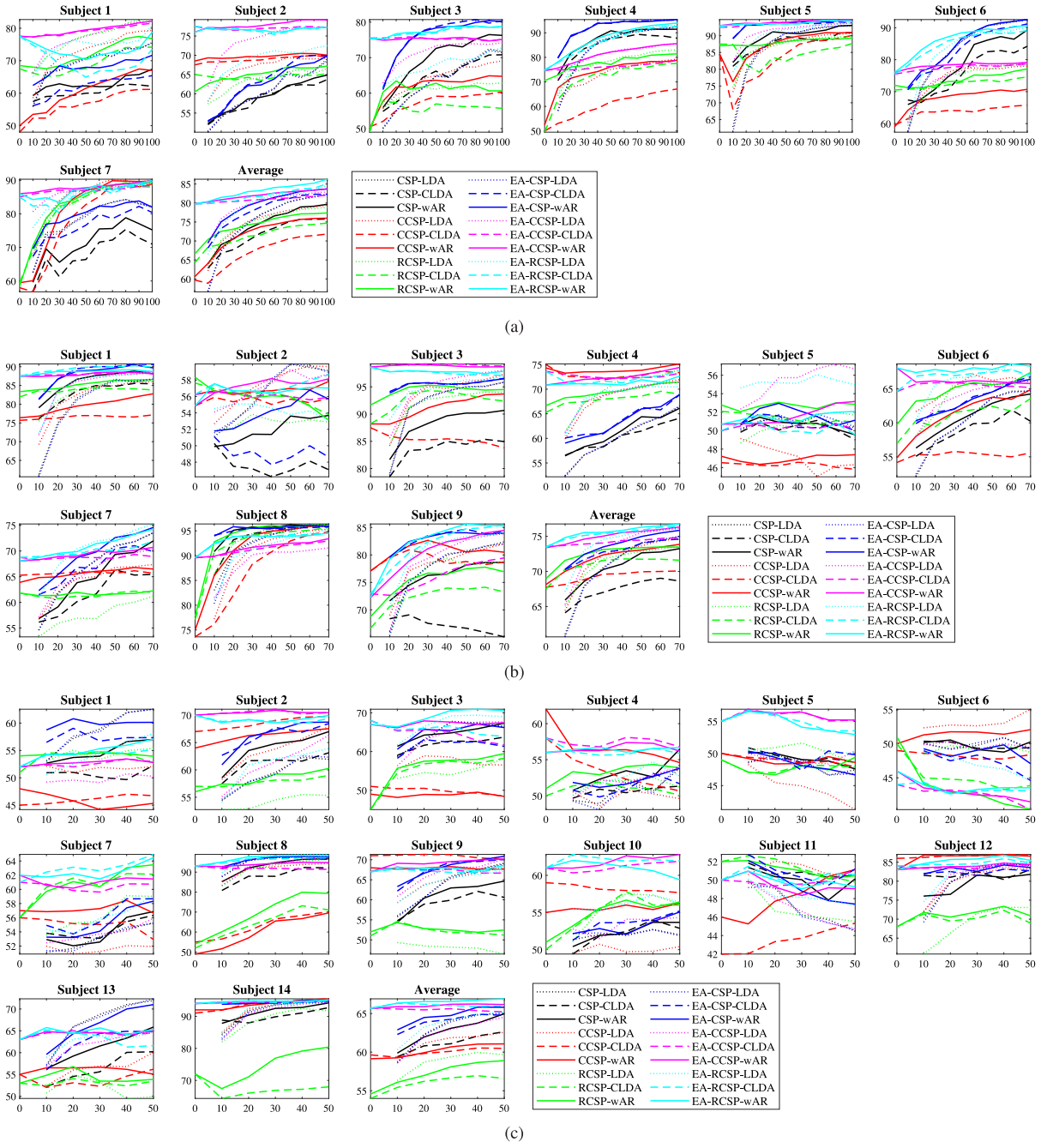

Fig. 7. Offline cross-subject classification accuracies (vertical axis), with different N_l (horizontal axis). (a) Dataset 1; (b) Dataset 2a; (c) Dataset 3.

(no TL in spatial filtering) and EA-CCSP (simple TL in spatial filtering) versions, suggesting it is important to apply EA first before using TL in spatial filtering.

3.7. Effect of TL in feature selection

Assume $2f$ spatial filters are needed. Then, as introduced in Section 2.4, in traditional CSP, f of them are the leading eigenvectors of $(\bar{C}^N)^{-1}\bar{C}^P$, and the other f are the leading eigenvectors of $(\bar{C}^P)^{-1}\bar{C}^N$, where $\bar{C}^P$ and $\bar{C}^N$ are the mean covariance matrices of the positive and negative classes, respectively. $f = 3$ is typically used in the literature.

In this subsection, in order to show the effect of TL in feature selection, we use $f = 10$ to extract 20 CSP filters, compute the 20 corresponding features in Eq. (8), and then use different versions of ReliefF to select the best six among them. More specifically, without data alignment, we compared the following algorithms:

1. *CSP6-LDA*, which is identical to *CSP-LDA* in previous subsections. Here we add '6' to emphasize that it uses six CSP filters, obtained from the leading eigenvectors.
2. *CSP20-ReliefF6-LDA*, which uses the traditional CSP algorithm to extract 20 spatial filters, compute the 20 corresponding features in Eq. (8), and then use ReliefF to select the best six from them. ReliefF uses the target domain labeled data only, i.e., no TL is used.
3. *CSP20-CReliefF6-LDA*, which uses the traditional CSP to extract 20 filters, compute the 20 corresponding features in

Table 3
Offline cross-subject classification accuracies (mean ± std) of different TL approaches, and their improvements over CSP-LDA. Performance improvements not statistically significant ($\alpha = 0.05$) are marked with an underline.

Algorithm	Dataset 1		Dataset 2a		Dataset 3	
	Acc. (%)	Imp. (%)	Acc. (%)	Imp. (%)	Acc. (%)	Imp. (%)
CSP-LDA	75.16 ± 8.02	–	70.77 ± 4.94	–	63.01 ± 2.66	–
CSP-CLDA	71.51 ± 4.51	−4.86	67.36 ± 1.74	−4.83	61.20 ± 1.25	−2.87
CSP-wAR	74.33 ± 5.13	−1.10	70.64 ± 2.65	−0.19	62.73 ± 1.93	−0.46
CCSP-LDA	74.83 ± 5.39	−0.44	71.84 ± 3.60	1.50	61.28 ± 1.49	−2.75
CCSP-CLDA	67.48 ± 4.39	−10.22	69.47 ± 0.72	−1.85	60.06 ± 0.50	−4.69
CCSP-wAR	72.64 ± 3.92	−3.35	72.40 ± 1.34	2.30	60.40 ± 0.78	−4.15
RCSP-LDA	74.73 ± 4.78	−0.58	71.18 ± 3.03	0.57	58.81 ± 1.47	−6.67
RCSP-CLDA	71.99 ± 2.32	−4.22	71.37 ± 0.57	0.84	56.37 ± 0.59	−10.54
RCSP-wAR	74.93 ± 2.35	−0.31	72.96 ± 0.71	3.08	57.75 ± 1.18	−8.36
EA-CSP-LDA	75.04 ± 8.06	−0.16	70.71 ± 5.10	−0.09	62.93 ± 2.67	−0.14
EA-CSP-CLDA	78.16 ± 4.66	3.99	72.97 ± 1.58	3.10	64.02 ± 1.05	1.59
EA-CSP-wAR	79.56 ± 4.50	5.85	73.81 ± 1.94	4.30	64.75 ± 1.21	2.76
EA-CCSP-LDA	79.48 ± 4.19	5.74	73.15 ± 3.84	3.36	62.39 ± 2.30	−1.00
EA-CCSP-CLDA	80.99 ± 0.69	7.75	74.29 ± 0.41	4.96	65.46 ± 0.17	3.88
EA-CCSP-wAR	82.21 ± 1.13	9.37	75.29 ± 0.86	6.38	66.08 ± 0.17	4.86
EA-RCSP-LDA	79.40 ± 5.50	5.63	73.97 ± 3.21	4.51	63.57 ± 1.96	0.88
EA-RCSP-CLDA	82.24 ± 1.72	9.42	75.62 ± 0.65	6.84	66.10 ± 0.16	4.89
EA-RCSP-wAR	83.47 ± 1.82	11.05	75.90 ± 0.67	7.24	66.62 ± 0.21	5.72
PS-CSP-LDA	75.17 ± 8.11	0.01	70.74 ± 5.06	−0.05	62.95 ± 2.67	−0.10
PS-CSP-CLDA	79.72 ± 4.13	6.06	72.97 ± 1.60	3.10	64.85 ± 1.20	2.91
PS-CSP-wAR	80.07 ± 4.13	6.53	73.82 ± 1.93	4.30	65.08 ± 1.25	3.28
PS-CCSP-LDA	77.95 ± 4.20	3.70	72.24 ± 3.13	2.07	60.36 ± 1.93	−4.21
PS-CCSP-CLDA	79.41 ± 0.88	5.65	74.00 ± 0.23	4.55	63.44 ± 0.14	0.67
PS-CCSP-wAR	80.51 ± 1.07	7.11	75.56 ± 0.61	6.76	63.20 ± 0.23	0.29
PS-RCSP-LDA	78.42 ± 4.15	4.34	73.36 ± 3.29	3.66	61.01 ± 1.60	−3.18
PS-RCSP-CLDA	80.05 ± 0.85	6.51	75.41 ± 0.87	6.55	63.71 ± 0.11	1.11
PS-RCSP-wAR	80.45 ± 0.79	7.04	75.80 ± 0.71	7.10	63.79 ± 0.15	1.22

Table 4
Offline cross-subject classification accuracies (mean ± std) of algorithms without and with EA/PS, and the improvements of the latter over the former. Performance improvements not statistically significant ($\alpha = 0.05$) are marked with an underline.

Dataset	Algorithm	w/o EA (%)	w/EA (%)	Imp. (%)	w/o PS (%)	w/PS (%)	Imp. (%)
1	CSP-LDA	75.16 ± 8.02	75.04 ± 8.06	−0.16	75.16 ± 8.02	75.17 ± 8.11	0.01
	CSP-CLDA	71.51 ± 4.51	78.16 ± 4.66	9.30	71.51 ± 4.51	79.72 ± 4.13	11.48
	CSP-wAR	74.33 ± 5.13	79.56 ± 4.50	7.03	74.33 ± 5.13	80.07 ± 4.13	7.72
	CCSP-LDA	74.83 ± 5.39	79.48 ± 4.19	6.22	74.83 ± 5.39	77.95 ± 4.20	4.17
	CCSP-CLDA	67.48 ± 4.39	80.99 ± 0.69	20.01	67.48 ± 4.39	79.41 ± 0.88	17.67
	CCSP-wAR	72.64 ± 3.92	82.21 ± 1.13	13.17	72.64 ± 3.92	80.51 ± 1.07	10.83
	RCSP-LDA	74.73 ± 4.78	79.40 ± 5.50	6.25	74.73 ± 4.78	78.42 ± 4.15	4.94
	RCSP-CLDA	71.99 ± 2.32	82.24 ± 1.72	14.24	71.99 ± 2.32	80.05 ± 0.85	11.20
	RCSP-wAR	74.93 ± 2.35	83.47 ± 1.82	11.40	74.93 ± 2.35	80.45 ± 0.79	7.37
2a	CSP-LDA	70.77 ± 4.94	70.71 ± 5.10	−0.09	70.77 ± 4.94	70.74 ± 5.06	−0.05
	CSP-CLDA	67.36 ± 1.74	72.97 ± 1.58	8.32	67.36 ± 1.74	72.97 ± 1.60	8.33
	CSP-wAR	70.64 ± 2.65	73.81 ± 1.94	4.49	70.64 ± 2.65	73.82 ± 1.93	4.50
	CCSP-LDA	71.84 ± 3.60	73.15 ± 3.84	1.83	71.84 ± 3.60	72.24 ± 3.13	0.55
	CCSP-CLDA	69.47 ± 0.72	74.29 ± 0.41	6.94	69.47 ± 0.72	74.00 ± 0.23	6.52
	CCSP-wAR	72.40 ± 1.34	75.29 ± 0.86	3.99	72.40 ± 1.34	75.56 ± 0.61	4.37
	RCSP-LDA	71.18 ± 3.03	73.97 ± 3.21	3.92	71.18 ± 3.03	73.36 ± 3.29	3.07
	RCSP-CLDA	71.37 ± 0.57	75.62 ± 0.65	5.96	71.37 ± 0.57	75.41 ± 0.87	5.67
	RCSP-wAR	72.96 ± 0.71	75.90 ± 0.67	4.03	72.96 ± 0.71	75.80 ± 0.71	3.90
3	CSP-LDA	63.01 ± 2.66	62.93 ± 2.67	−0.14	63.01 ± 2.66	62.95 ± 2.67	−0.10
	CSP-CLDA	61.20 ± 1.25	64.02 ± 1.05	4.59	61.20 ± 1.25	64.85 ± 1.20	5.95
	CSP-wAR	62.73 ± 1.93	64.75 ± 1.21	3.23	62.73 ± 1.93	65.08 ± 1.25	3.76
	CCSP-LDA	61.28 ± 1.49	62.39 ± 2.30	1.81	61.28 ± 1.49	60.36 ± 1.93	−1.50
	CCSP-CLDA	60.06 ± 0.50	65.46 ± 0.17	8.99	60.06 ± 0.50	63.44 ± 0.14	5.63
	CCSP-wAR	60.40 ± 0.78	66.08 ± 0.17	9.41	60.40 ± 0.78	63.20 ± 0.23	4.64
	RCSP-LDA	58.81 ± 1.47	63.57 ± 1.96	8.10	58.81 ± 1.47	61.01 ± 1.60	3.74
	RCSP-CLDA	56.37 ± 0.59	66.10 ± 0.16	17.25	56.37 ± 0.59	63.71 ± 0.11	13.02
	RCSP-wAR	57.75 ± 1.18	66.62 ± 0.21	15.37	57.75 ± 1.18	63.79 ± 0.15	10.46

Eq. (8), and then use CReliefF to select the best six from them. CReliefF uses labeled data from both domains, so there is TL.

LDA above can also be replaced by wAR, and CSP by RCSP. So, there could be 12 different configurations. Additionally, EA can also be added before each algorithm. So, there are a total of 24 different algorithms.

The cross-subject classification results of the 24 algorithms are shown in Table 6. When CReliefF was used to select the best six spatial filters from the 20 candidates (CSP20→CReliefF6), the resulting classification performance was generally better than

Table 5
The effect of TL in spatial filtering. Performance improvements not statistically significant ($\alpha = 0.05$) are marked with an underline.

Dataset	Algorithm	No TL	Simple TL		Sophisticated TL	
		CSP (%)	CCSP (%)	Imp. (%)	RCSP (%)	Imp. (%)
1	CSP-LDA	75.16 ± 8.02	74.83 ± 5.39	−0.44	74.73 ± 4.78	−0.58
	CSP-CLDA	71.51 ± 4.51	67.48 ± 4.39	−5.63	71.99 ± 2.32	0.67
	CSP-wAR	74.33 ± 5.13	72.64 ± 3.92	−2.28	74.93 ± 2.35	0.80
	EA-CSP-LDA	75.04 ± 8.06	79.48 ± 4.19	5.91	79.40 ± 5.50	5.80
	EA-CSP-CLDA	78.16 ± 4.66	80.99 ± 0.69	3.62	82.24 ± 1.72	5.22
	EA-CSP-wAR	79.56 ± 4.50	82.21 ± 1.13	3.33	83.47 ± 1.82	4.91
	PS-CSP-LDA	75.17 ± 8.11	77.95 ± 4.20	3.70	78.42 ± 4.15	4.33
	PS-CSP-CLDA	79.72 ± 4.13	79.41 ± 0.88	−0.39	80.05 ± 0.85	0.42
	PS-CSP-wAR	80.07 ± 4.13	80.51 ± 1.07	0.55	80.45 ± 0.79	0.48
2a	CSP-LDA	70.77 ± 4.94	71.84 ± 3.60	1.50	71.18 ± 3.03	0.57
	CSP-CLDA	67.36 ± 1.74	69.47 ± 0.72	3.13	71.37 ± 0.57	5.95
	CSP-wAR	70.64 ± 2.65	72.40 ± 1.34	2.49	72.96 ± 0.71	3.28
	EA-CSP-LDA	70.71 ± 5.10	73.15 ± 3.84	3.45	73.97 ± 3.21	4.61
	EA-CSP-CLDA	72.97 ± 1.58	74.29 ± 0.41	1.81	75.62 ± 0.65	3.63
	EA-CSP-wAR	73.81 ± 1.94	75.29 ± 0.86	2.00	75.90 ± 0.67	2.82
	PS-CSP-LDA	70.74 ± 5.06	72.24 ± 3.13	2.12	73.36 ± 3.29	3.71
	PS-CSP-CLDA	72.97 ± 1.60	74.00 ± 0.23	1.41	75.41 ± 0.87	3.34
	PS-CSP-wAR	73.82 ± 1.93	75.56 ± 0.61	2.36	75.80 ± 0.71	2.69
3	CSP-LDA	63.01 ± 2.66	61.28 ± 1.49	−2.75	58.81 ± 1.47	−6.67
	CSP-CLDA	61.20 ± 1.25	60.06 ± 0.50	−1.87	56.37 ± 0.59	−7.90
	CSP-wAR	62.73 ± 1.93	60.40 ± 0.78	−3.72	57.75 ± 1.18	−7.94
	EA-CSP-LDA	62.93 ± 2.67	62.39 ± 2.30	−0.86	63.57 ± 1.96	1.03
	EA-CSP-CLDA	64.02 ± 1.05	65.46 ± 0.17	2.25	66.10 ± 0.16	3.25
	EA-CSP-wAR	64.75 ± 1.21	66.08 ± 0.17	2.05	66.62 ± 0.21	2.89
	PS-CSP-LDA	62.95 ± 2.67	60.36 ± 1.93	−4.11	61.01 ± 1.60	−3.08
	PS-CSP-CLDA	64.85 ± 1.20	63.44 ± 0.14	−2.17	63.71 ± 0.11	−1.75
	PS-CSP-wAR	65.08 ± 1.25	63.20 ± 0.23	−2.90	63.79 ± 0.15	−1.99

Table 6
The effect of TL in feature selection. Performance improvements not statistically significant ($\alpha = 0.05$) are marked with an underline.

Dataset	Algorithm	No TL		TL	
		CSP6 (%)	CSP20→ReliefF6 (%)	CSP20→CReliefF6 (%)	Imp. (%)
1	CSP-LDA	75.28 ± 7.83	73.90 ± 7.72	74.18 ± 7.00	0.38
	CSP-wAR	75.17 ± 5.11	72.10 ± 6.31	74.68 ± 5.33	3.59
	RCSP-LDA	74.41 ± 4.75	72.43 ± 5.73	73.47 ± 4.57	1.43
	RCSP-wAR	74.72 ± 2.15	71.02 ± 3.54	73.60 ± 1.69	3.64
	EA-CSP-LDA	75.18 ± 7.81	72.07 ± 7.43	75.64 ± 7.40	4.95
	EA-CSP-wAR	79.90 ± 4.33	74.69 ± 4.90	81.36 ± 4.70	8.92
	EA-RCSP-LDA	79.43 ± 5.43	75.52 ± 5.92	79.49 ± 5.25	5.26
	EA-RCSP-wAR	83.85 ± 1.60	78.98 ± 3.64	84.66 ± 1.95	7.19
2a	CSP-LDA	70.64 ± 4.75	69.86 ± 5.05	70.30 ± 4.31	0.63
	CSP-wAR	70.05 ± 2.33	69.58 ± 2.85	69.60 ± 2.85	0.02
	RCSP-LDA	71.00 ± 3.11	69.72 ± 3.29	70.59 ± 3.25	1.25
	RCSP-wAR	72.83 ± 0.55	70.77 ± 1.00	72.46 ± 0.91	2.40
	EA-CSP-LDA	70.50 ± 4.87	69.64 ± 4.98	68.06 ± 4.11	−2.27
	EA-CSP-wAR	73.51 ± 1.84	73.00 ± 2.30	70.63 ± 1.32	−3.24
	EA-RCSP-LDA	74.09 ± 3.19	71.29 ± 3.57	68.57 ± 2.92	−3.80
	EA-RCSP-wAR	75.93 ± 0.66	74.42 ± 0.90	70.34 ± 0.70	−5.48
3	CSP-LDA	62.60 ± 2.49	62.10 ± 2.33	61.96 ± 2.20	−0.22
	CSP-wAR	62.59 ± 1.50	62.01 ± 1.72	61.04 ± 1.50	−1.55
	RCSP-LDA	58.50 ± 1.34	58.93 ± 1.35	58.14 ± 1.74	−1.34
	RCSP-wAR	58.23 ± 1.31	57.52 ± 1.93	57.30 ± 1.35	−0.39
	EA-CSP-LDA	62.49 ± 2.61	61.98 ± 2.45	61.33 ± 2.17	−1.05
	EA-CSP-wAR	64.68 ± 1.11	64.95 ± 1.20	63.64 ± 1.11	−2.02
	EA-RCSP-LDA	63.35 ± 1.96	62.70 ± 1.94	62.91 ± 1.84	0.34
	EA-RCSP-wAR	66.30 ± 0.21	65.45 ± 0.97	66.37 ± 0.14	1.40

using ReliefF directly (CSP20→ReliefF6) on Dataset 1, but mixed on Datasets 2a and 3. However, using the six leading eigenvectors in CSP (CSP6) generally gave the best performance, justifying the common practice in the literature.

In summary, we have shown that when CSP filters are used, using the leading eigenvectors is good enough, and we can safely omit feature selection.

3.8. Effect of TL in the classifier

In Fig. 7, comparing algorithms with simple and sophisticated TL in the classifier, e.g., CCSP-CLDA and CCSP-wAR, we can observe that sophisticated TL in the classifier almost always outperformed simple TL, regardless of whether TL was used in other components or not. So, sophisticated TL approaches, such as wAR, should be used in the classifier of BCIs.

Table 7
The effect of TL in the classifier. Performance improvements not statistically significant ($\alpha = 0.05$) are marked with an underline.

Dataset	Algorithm	No TL	Simple TL		Sophisticated TL	
		LDA (%)	CLDA (%)	Imp. (%)	wAR (%)	Imp. (%)
1	CSP-LDA	75.16 ± 8.02	71.51 ± 4.51	−4.86	74.33 ± 5.13	−1.10
	CCSP-LDA	74.83 ± 5.39	67.48 ± 4.39	−9.82	72.64 ± 3.92	−2.92
	RCSP-LDA	74.73 ± 4.78	71.99 ± 2.32	−3.67	74.93 ± 2.35	0.27
	EA-CSP-LDA	75.04 ± 8.06	78.16 ± 4.66	4.15	79.56 ± 4.50	6.02
	EA-CCSP-LDA	79.48 ± 4.19	80.99 ± 0.69	1.90	82.21 ± 1.13	3.43
	EA-RCSP-LDA	79.40 ± 5.50	82.24 ± 1.72	3.58	83.47 ± 1.82	5.13
	PS-CSP-LDA	75.17 ± 8.11	79.72 ± 4.13	6.06	80.07 ± 4.13	6.52
	PS-CCSP-LDA	77.95 ± 4.20	79.41 ± 0.88	1.88	80.51 ± 1.07	3.29
	PS-RCSP-LDA	78.42 ± 4.15	80.05 ± 0.85	2.08	80.45 ± 0.79	2.59
2a	CSP-LDA	70.77 ± 4.94	67.36 ± 1.74	−4.83	70.64 ± 2.65	−0.19
	CCSP-LDA	71.84 ± 3.60	69.47 ± 0.72	−3.30	72.40 ± 1.34	0.78
	RCSP-LDA	71.18 ± 3.03	71.37 ± 0.57	0.27	72.96 ± 0.71	2.50
	EA-CSP-LDA	70.71 ± 5.10	72.97 ± 1.58	3.19	73.81 ± 1.94	4.39
	EA-CCSP-LDA	73.15 ± 3.84	74.29 ± 0.41	1.55	75.29 ± 0.86	2.93
	EA-RCSP-LDA	73.97 ± 3.21	75.62 ± 0.65	2.23	75.90 ± 0.67	2.61
	PS-CSP-LDA	70.74 ± 5.06	72.97 ± 1.60	3.16	73.82 ± 1.93	4.36
	PS-CCSP-LDA	72.24 ± 3.13	74.00 ± 0.23	2.44	75.56 ± 0.61	4.60
	PS-RCSP-LDA	73.36 ± 3.29	75.41 ± 0.87	2.79	75.80 ± 0.71	3.32
3	CSP-LDA	63.01 ± 2.66	61.20 ± 1.25	−2.87	62.73 ± 1.93	−0.46
	CCSP-LDA	61.28 ± 1.49	60.06 ± 0.50	−1.99	60.40 ± 0.78	−1.44
	RCSP-LDA	58.81 ± 1.47	56.37 ± 0.59	−4.15	57.75 ± 1.18	−1.81
	EA-CSP-LDA	62.93 ± 2.67	64.02 ± 1.05	1.73	64.75 ± 1.21	2.90
	EA-CCSP-LDA	62.39 ± 2.30	65.46 ± 0.17	4.92	66.08 ± 0.17	5.91
	EA-RCSP-LDA	63.57 ± 1.96	66.10 ± 0.16	3.97	66.62 ± 0.21	4.80
	PS-CSP-LDA	62.95 ± 2.67	64.85 ± 1.20	3.02	65.08 ± 1.25	3.39
	PS-CCSP-LDA	60.36 ± 1.93	63.44 ± 0.14	5.10	63.20 ± 0.23	4.70
	PS-RCSP-LDA	61.01 ± 1.60	63.71 ± 0.11	4.43	63.79 ± 0.15	4.55

To quantitatively verify the above conclusion, we also show the mean classification accuracies of algorithms without and with TL in the classifier in Table 7. When EA or PS was used, on average wAR (sophisticated TL in the classifier) always outperformed CLDA (simple TL in the classifier).

3.9. Offline cross-session classification

It is well-known that EEG signals are non-stationary (Liyanage et al., 2013), i.e., EEG responses to the same stimulus from the same subject in different sessions are usually varying. This subsection evaluates how the proposed TL pipeline can be used to handle EEG non-stationarity in cross-session classification.

In Dataset 2a, each of the nine subjects had two sessions (training and evaluation), collected on different days. For each subject, we used the training session as the source domain, and the evaluation session as the target domain. Other experimental settings were identical to those in previous subsections, except that we used $w_t = 2$ in wAR, as in cross-session TL the number of labeled source domain samples was much smaller than that in cross-subject TL.

The results are shown in Fig. 8. Some subjects, e.g., Subjects 1, 3, 7, 8 and 9, demonstrated very stable classification performance when N_l increased, indicating that their EEG signals were quite stationary, at least in the two experimental sessions. However, the remaining four subjects' EEG signals were more non-stationary, and hence the classification performance had large fluctuations.

The average classification results are shown in Table 8. EA-RCSP-wAR, which was the best performer in cross-subject transfers in previous subsections, still achieved the best average performance. On average, almost all approaches with EA outperformed their counterparts without EA, suggesting again the importance and necessity of explicitly adding a data alignment block before TL.

In summary, TL is also effective in handling non-stationarity of EEG signals in cross-session MI classification, and considering TL in more components of the classification pipeline is generally more beneficial.

Table 8
Offline cross-session classification accuracies (mean ± std) of different TL approaches, and their improvements over CSP-LDA. Performance improvements not statistically significant ($\alpha = 0.05$) are marked with an underline.

Algorithm	Acc. (%)	Imp. (%)
CSP-LDA	70.82 ± 5.15	–
CSP-CLDA	73.28 ± 2.50	3.46
CSP-wAR	73.82 ± 2.24	4.23
CCSP-LDA	72.69 ± 3.17	2.63
CCSP-CLDA	74.00 ± 1.13	4.49
CCSP-wAR	74.79 ± 0.62	5.61
RCSP-LDA	70.89 ± 3.46	0.09
RCSP-CLDA	73.62 ± 0.85	3.94
RCSP-wAR	73.63 ± 0.72	3.96
EA-CSP-LDA	70.73 ± 5.25	−0.13
EA-CSP-CLDA	74.57 ± 2.09	5.29
EA-CSP-wAR	74.66 ± 2.09	5.43
EA-CCSP-LDA	73.32 ± 3.31	3.52
EA-CCSP-CLDA	74.87 ± 0.49	5.72
EA-CCSP-wAR	75.74 ± 0.25	6.94
EA-RCSP-LDA	72.84 ± 3.26	2.85
EA-RCSP-CLDA	75.57 ± 1.11	6.70
EA-RCSP-wAR	76.02 ± 0.83	7.34

3.10. Offline cross-subject and cross-session classification

This subsection evaluates the performance of the TL pipeline in a more challenging scenario on Dataset 2a, where all training sessions from eight subjects were combined as the source domain, some labeled samples from the training session of the remaining subject (target subject) was used as the target domain calibration data, and the TL performance was evaluated on the validation session of the target subject. This scenario is challenging, as it includes cross-subject variations and cross-session variations simultaneously.

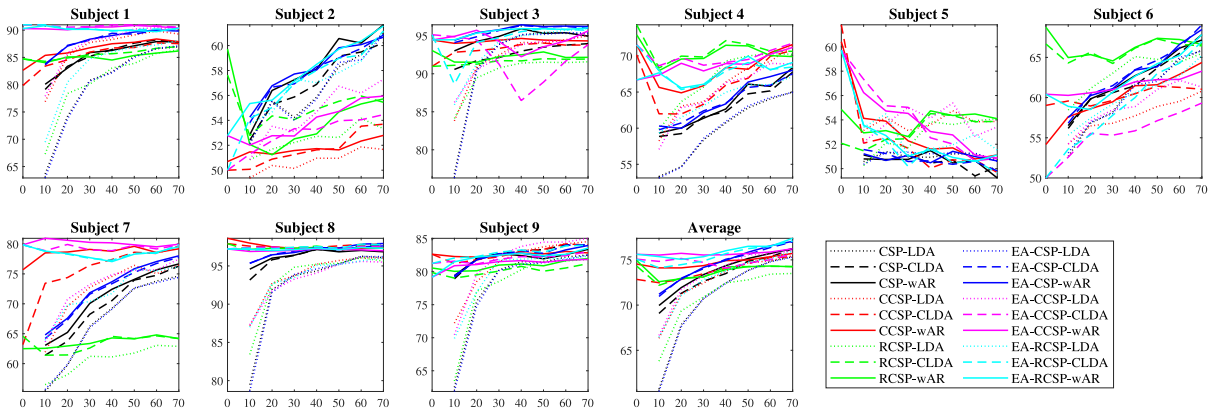

Fig. 8. Offline cross-session classification accuracies (vertical axis) on Dataset 2a, with different N_l (horizontal axis).

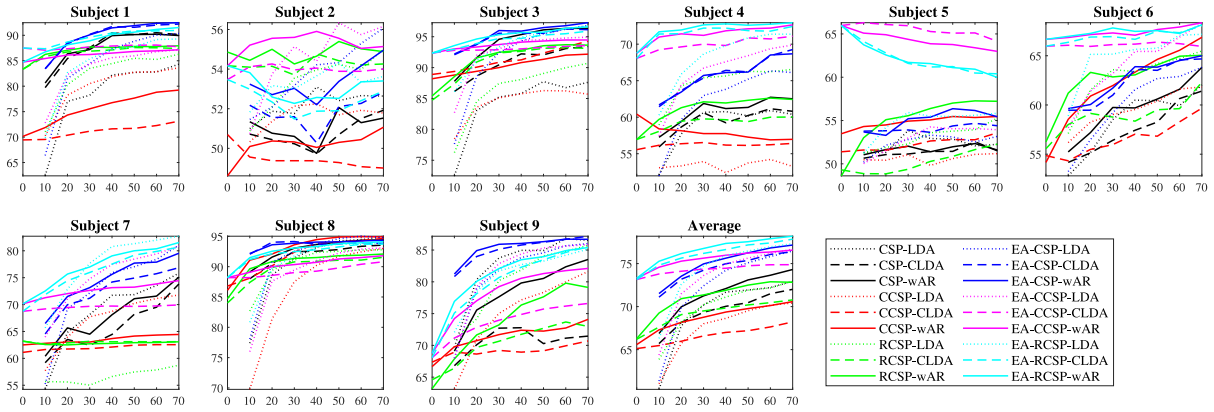

Fig. 9. Offline cross-subject and cross-session classification accuracies (vertical axis) on Dataset 2a, with different N_l (horizontal axis).

The results are shown in Fig. 9. Most subjects, e.g., Subjects 1, 3, 4, 6, 7, 8 and 9, demonstrated very stable classification performance when N_l increased, indicating that their EEG signals were quite stationary, at least in the two experimental sessions. However, the remaining two subjects' EEG signals were more non-stationary, and hence the classification performance had large fluctuations.

The average classification results are shown in Table 9. EA-RCSP-wAR, which was the best performer in cross-subject transfers in previous subsections, still achieved the best performance. On average, all approaches with EA outperformed their counterparts without EA, suggesting again the importance and necessity of explicitly adding a data alignment block before TL.

In summary, TL is also effective in handling non-stationarity of EEG signals in cross-subject and cross-session MI classification, and considering TL in more components of the classification pipeline is generally more beneficial.

Table 9
Offline cross-subject and cross-session classification accuracies (mean ± std) of different TL approaches, and their improvements over CSP-LDA. Performance improvements not statistically significant ($\alpha = 0.05$) are marked with an underline.

Algorithm	Acc. (%)	Imp. (%)
CSP-LDA	69.59 ± 4.41	–
CSP-CLDA	69.60 ± 2.15	0.02
CSP-wAR	71.61 ± 2.55	2.91
CCSP-LDA	67.64 ± 3.32	−2.79
CCSP-CLDA	66.87 ± 0.95	−3.90
CCSP-wAR	69.15 ± 1.15	−0.62
RCSP-LDA	70.06 ± 3.14	0.68
RCSP-CLDA	69.64 ± 1.06	0.07
RCSP-wAR	71.66 ± 1.30	2.99
EA-CSP-LDA	72.19 ± 5.27	3.74
EA-CSP-CLDA	74.47 ± 1.84	7.02
EA-CSP-wAR	75.12 ± 2.00	7.96
EA-CCSP-LDA	73.27 ± 4.30	5.30
EA-CCSP-CLDA	74.46 ± 0.41	7.00
EA-CCSP-wAR	75.80 ± 0.69	8.93
EA-RCSP-LDA	74.33 ± 3.88	6.82
EA-RCSP-CLDA	76.53 ± 1.06	9.97
EA-RCSP-wAR	77.02 ± 1.03	10.69

3.11. Source Subject Selection (SSS)

When there are many source subjects, performing one-time TL by combining labeled EEG data from all source subjects, or performing separate TL for each source subject individually and then aggregating the results, is computationally expensive. Additionally, including source subjects whose EEG data distributions are significantly different from that of the target subject may deteriorate the TL performance, i.e., leading to negative transfer (Zhang, Deng, Zhang, & Wu, 2022).

Source subject selection (SSS) (Wu, 2017; Zhang et al., 2022; Zhang & Wu, 2020), which selects source subjects that are most similar to the target subject in TL, may be used to alleviate the computational cost and negative transfer problems. Unfortunately, reliable SSS is not easy. Our previous experiments (Wu, 2017; Zhang & Wu, 2020) showed that SSS generally results in slightly worse performance than, or at most similar performance

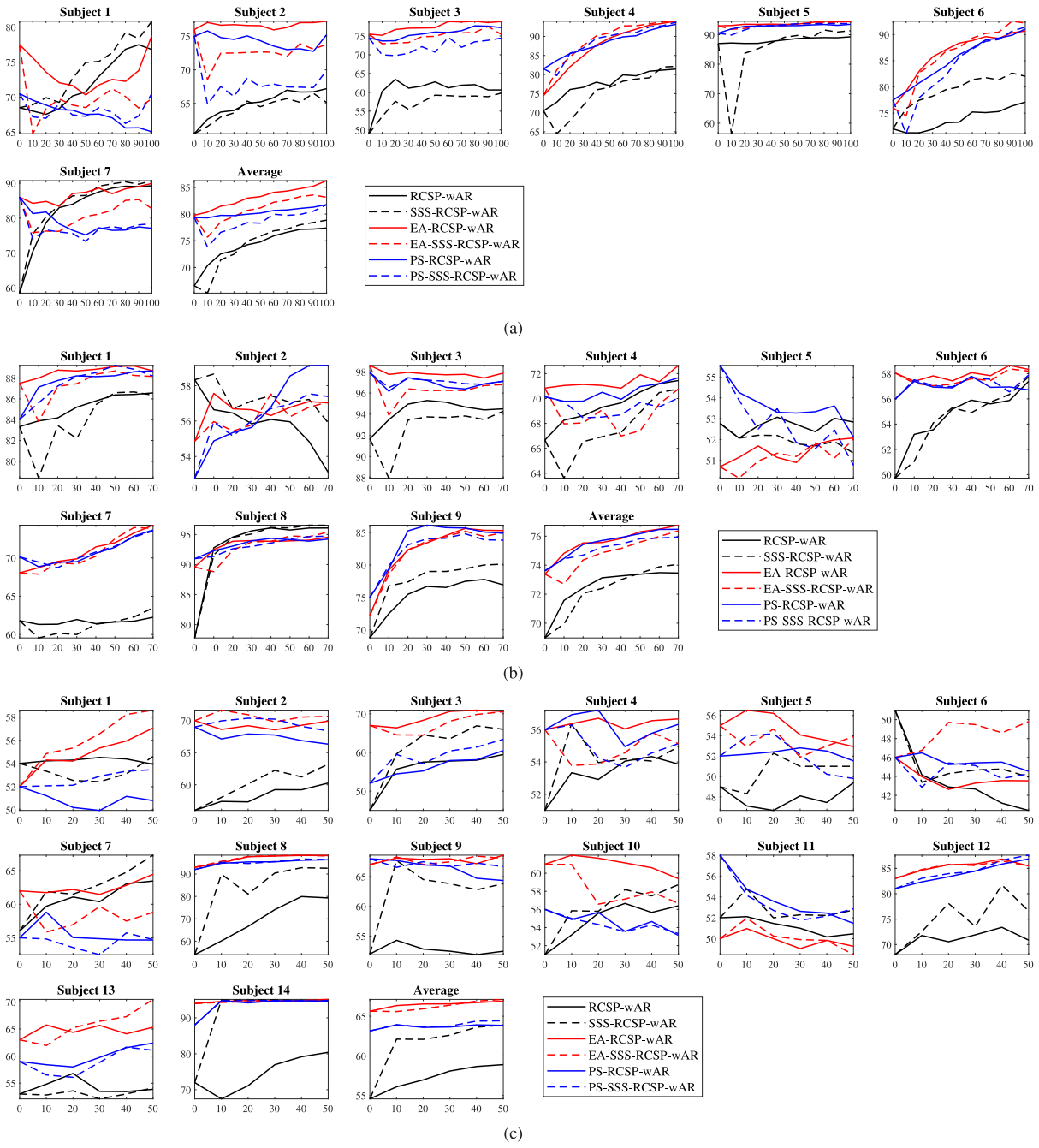

Fig. 10. Offline cross-subject classification accuracies (vertical axis), with different N_l (horizontal axis). SSS means source subject selection (Wu, 2017) was used. (a) Dataset 1; (b) Dataset 2a; (c) Dataset 3.

with, using all source subjects. So far the main benefit of SSS seems to reducing the computational cost. A lot more research is needed to reliably improve the learning performance through SSS, which is out of the scope of this tutorial.

Nevertheless, it is interesting to study whether the proposed TL pipeline also works with existing SSS approaches. Fig. 10 shows the experimental results on the source domain (subject) selection approach proposed in Wu (2017), which first computes a distance measure between each source subject and the target subject, and then uses k-means ($k = 2$) clustering to select the group of source subjects closer to the target subject. For simplicity, we only augmented SSS to the three best-performing approaches (RCSP-wAR, EA-RCSP-wAR, and PS-RCSP-wAR).

We can observe from Fig. 10 that:

1. Without data alignment (EA or PS), using SSS on average led to better or comparable performance, especially when N_l was large. For example, in Fig. 10(c), the average performance of SSS-RCSP-wAR is much better than RCSP-wAR.
2. With data alignment (EA or PS), using SSS on average resulted in worse or similar performance, consistent with the results in Zhang and Wu (2020). For example, in Fig. 10(a), the average performance of EA-SSS-RCSP-wAR (PS-SSS-RCSP-wAR) is much worse than EA-RCSP-wAR (PS-RCSP-wAR).
3. On all three datasets, on average both EA-SSS-RCSP-wAR and PS-SSS-RCSP-wAR outperformed SSS-RCSP-wAR, i.e., data alignment was still helpful when SSS was used.

Table 10
The effect of EA in deep learning. Performance improvements not statistically significant ($\alpha = 0.05$) are marked with an underline.

Dataset	Model	w/o EA (%)	w/EA (%)	Imp. (%)
1	EEGNet	64.49 ± 3.78	69.24 ± 3.23	7.36
	ShallowCNN	63.97 ± 3.52	68.97 ± 3.05	7.82
2a	EEGNet	63.20 ± 4.80	69.61 ± 5.09	10.15
	ShallowCNN	65.90 ± 3.00	74.50 ± 2.77	13.04
3	EEGNet	75.10 ± 4.31	75.68 ± 3.10	0.77
	ShallowCNN	68.94 ± 3.12	74.53 ± 2.85	8.11

In summary, when the computational cost is not a big concern, it is suggested to perform EA first and then use all source subjects in TL, which usually results in better performance than SSS.

3.12. Offline cross-subject classification using deep learning

The previous subsections verified the effectiveness of TL in a traditional machine learning pipeline. Deep learning (Goodfellow, Bengio, & Courville, 2016) has made significant breakthroughs in many fields, including EEG-based BCIs. This subsection considers the TL pipeline in offline cross-subject MI classification using deep learning models.

Two popular convolutional neural network (CNN) classifiers, EEGNet (Lawhern et al., 2018) and ShallowCNN (Schirrmeister et al., 2017), were used in our experiments. EEGNet is a compact convolutional network with only about 1000 parameters (the number may change slightly according to the nature of the task) for EEG-based BCIs. It introduces depthwise and separable convolutions into the construction of EEG-specific CNNs, which encapsulate well-known EEG feature extraction concepts and simultaneously reduce the number of model parameters. ShallowCNN has a very shallow architecture, consisting of a convolutional block and a classification block. The convolutional block is specially designed to handle EEG signals.

Because CNN models perform simultaneously spatial filtering, feature engineering and classification, and their computational cost is much higher than traditional machine learning models, we only compared the performances with and without EA in Fig. 3. More specifically, we considered offline unsupervised cross-subject classification, i.e., all samples from the target subject were unlabeled ($N_l = 0$), and all samples from the source subjects were combined and partitioned into 80% training and 20% validation (for early stopping). We used Adam optimizer (Kingma & Ba, 2015), cross-entropy loss, and batch size 32. The experiments were repeated 15 times for each target subject.

The results are shown in Table 10. Clearly, using EA can improve the classification performance of both deep learning models, especially on Dataset 1. These results were consistent with a more comprehensive study in Kostas and Rudzicz (2020), which shows that EA generally benefits deep learning classification of movement and motor imagery, P300, and error related negativity.

3.13. Online cross-subject classification

All previous results considered offline MI classification. It is also interesting to study if the TL pipeline in Fig. 3 can be used in online cross-subject MI classification.

To this end, we make sure EA and PS use only the available labeled target domain samples in computing the reference matrix and performing data alignment in the target domain. We also replace wAR by OwAR (Wu, 2017), which does not make use of offline unlabeled samples in the target domain in classifier training.

The online cross-subject classification results are shown in Fig. 11 and Table 11. Generally, all observations made from offline cross-subject classification in previous subsections, e.g., considering TL in more components of Fig. 3 benefits the performance more, and EA is very important to subsequent TL, still hold in online cross-subject classification.

Comparing Tables 3 and 11 shows that the offline classification performances were generally slightly better than their online counterparts, which is intuitive, as offline classification makes use of the unlabeled target domain samples, which provides extra information in EA, PS and wAR.

4. Conclusions and future research

Transfer learning has been widely used in MI-based BCIs to reduce the calibration effort for a new subject, greatly increasing their utility. This tutorial demonstrates that TL could be considered in spatial filtering, feature engineering, and classification blocks of a closed-loop MI-based BCI system, and it is also very important to specifically add a data alignment component before spatial filtering to make the source domain and target domain data more consistent. Offline and online classification experiments on three MI datasets verified that:

1. Generally, using TL in different components of Fig. 3 can achieve better classification performance than not using it, in both cross-subject and cross-session classifications, for both online and offline classifications.
2. Generally, a more sophisticated TL approach outperforms a simple one.
3. Data alignment, particularly EA, is a very important preprocessing step in TL. It benefits both traditional machine learning and deep learning.
4. TL in different components of Fig. 3 could be complementary to each other, so integrating them can further improve the classification performance.

The following directions could be considered in future research:

1. It has been shown that TL, particularly data alignment, also benefits several other spatial filters, e.g., xDAWN (Rivet et al., 2009; Wu et al., 2018), and classifiers, e.g., SVM and Minimum Distance to Riemannian Mean (MDRM) (Barachant, Bonnet, Congedo, & Jutten, 2012; Yger et al., 2017), in BCIs. As new classifiers keep emerging, e.g., sparse representation classification (Jiao et al., 2019; Oikonomou, Nikolopoulos, & Kompatsiaris, 2020; Shin, Lee, Lee, & Lee, 2012), it is also interesting to study if TL benefits them; if not, then how to adapt them to TL settings.
2. Compared with other components, not enough attention has been paid to TL in feature engineering of BCI systems. More sophisticated TL approaches for feature engineering, and also other components in Fig. 3, could be developed in the future.
3. Too many source subjects may incur high computational cost, and including source subjects dramatically different from the target subject in TL may deteriorate the learning performance. Source subject selection could be used to alleviate these problems. Unfortunately, no existing source subject selection approach can reliably reduce the computational cost and improve the TL performance simultaneously, and hence more investigation is needed.
4. This tutorial considers only binary MI classification problems in BCIs. The analysis may be extended to multiclass MI classification, and also other BCI classification paradigms, e.g., event-related potentials and steady-state

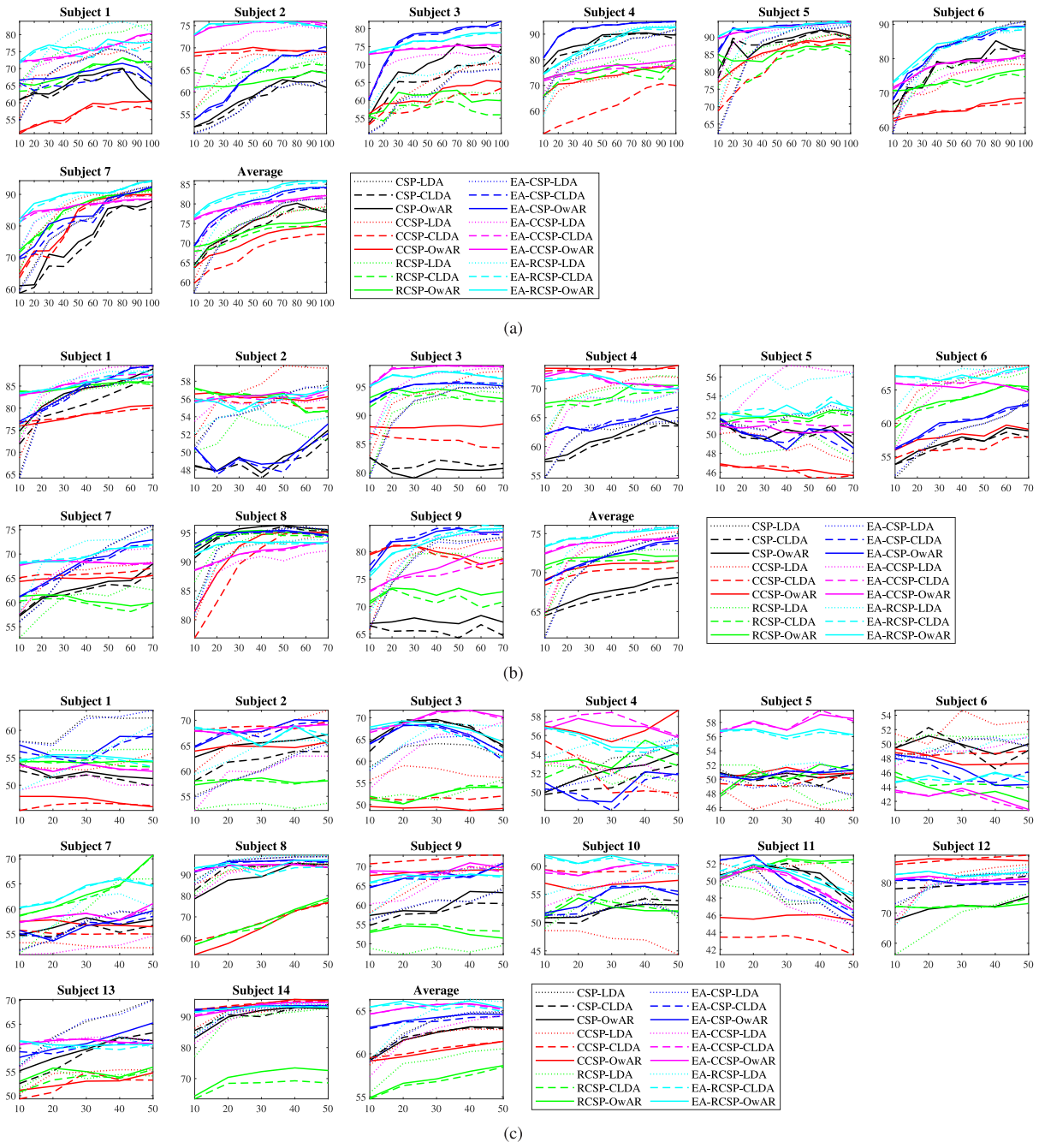

Fig. 11. Online cross-subject classification accuracies (vertical axis) on the MI datasets, with different N_l (horizontal axis). (a) Dataset 1; (b) Dataset 2a; (c) Dataset 3.

visual evoked potentials, and also BCI regression problems, e.g., driver drowsiness estimation (Cui et al., 2019; Wu, Lawhern, Gordon, Lance and Lin, 2017) and user reaction-time estimation (Wu et al., 2017).

5. It has been shown (Wu, Lance, & Lawhern, 2014; Wu et al., 2016) that integrating TL with active learning (Marathe, Lawhern, Wu, Slayback, & Lance, 2016) in the classifier can further improve the offline classification performance. It is interesting to study if TL and active learning can be integrated in other components of the BCI system, e.g., spatial filtering and feature engineering.
6. TL in BCIs is subject to adversarial attacks (Wu et al., 2022), particularly, poisoning (backdoor) attacks (Meng et al., 2022). The attacker can inject poisoning samples into an auxiliary EEG dataset. Once it is used in TL, a dangerous backdoor could be created in the resulting classifier, and any test sample with the backdoor key will then be classified into a class specified by the attacker, i.e., the output of the BCI system can easily be manipulated. It is necessary to identify more such risks, and most importantly, approaches to defend against these attacks, to increase the security and robustness of BCIs.
7. EEG signals in BCIs contain rich private information (Xia et al., 2022), e.g., health conditions, mental states, etc. As there are increasing requirements for privacy protection from both the governments (e.g., General Data Protection Regulation of the European Union General Data Protection Regulation, 2016) and the end-users, future TL

Table 11
Online cross-subject classification accuracies (mean ± std) of different TL approaches, and their improvements over CSP-LDA. Performance improvements not statistically significant ($\alpha = 0.05$) are marked with an underline.

Algorithm	Dataset 1		Dataset 2a		Dataset 3	
	Acc. (%)	Imp. (%)	Acc. (%)	Imp. (%)	Acc. (%)	Imp. (%)
CSP-LDA	74.65 ± 7.84	–	70.94 ± 4.52	–	63.04 ± 2.32	–
CSP-CLDA	73.69 ± 5.20	−1.28	66.81 ± 1.48	−5.82	61.98 ± 1.50	−1.68
CSP-OwAR	74.65 ± 5.12	0.00	67.50 ± 1.59	−4.84	61.91 ± 1.61	−1.80
CCSP-LDA	73.98 ± 5.86	−0.90	72.42 ± 3.86	2.10	61.70 ± 1.77	−2.14
CCSP-CLDA	67.76 ± 4.45	−9.23	70.00 ± 0.80	−1.31	60.52 ± 0.77	−4.01
CCSP-OwAR	70.61 ± 3.65	−5.41	70.77 ± 0.82	−0.24	60.29 ± 0.92	−4.37
RCSP-LDA	74.70 ± 4.86	0.07	70.97 ± 3.09	0.05	58.91 ± 2.04	−6.56
RCSP-CLDA	72.10 ± 2.66	−3.41	71.36 ± 0.43	0.59	56.76 ± 1.39	−9.96
RCSP-OwAR	73.19 ± 2.46	−1.96	71.91 ± 0.47	1.37	57.03 ± 1.43	−9.54
EA-CSP-LDA	74.46 ± 8.03	−0.25	70.90 ± 4.64	−0.05	63.00 ± 2.49	−0.07
EA-CSP-CLDA	79.32 ± 4.85	6.26	71.84 ± 1.79	1.28	63.80 ± 0.56	1.20
EA-CSP-OwAR	79.94 ± 4.83	7.09	71.98 ± 1.86	1.48	64.07 ± 0.65	1.63
EA-CCSP-LDA	78.14 ± 4.90	4.68	73.21 ± 3.97	3.21	62.03 ± 2.85	−1.61
EA-CCSP-CLDA	79.57 ± 1.83	6.60	73.76 ± 0.61	3.98	65.31 ± 0.48	3.60
EA-CCSP-OwAR	79.87 ± 1.88	6.99	73.78 ± 0.71	4.02	65.36 ± 0.47	3.68
EA-RCSP-LDA	79.10 ± 6.88	5.96	73.72 ± 3.25	3.92	64.09 ± 2.92	1.66
EA-RCSP-CLDA	82.57 ± 2.89	10.62	74.89 ± 0.84	5.58	65.38 ± 0.37	3.70
EA-RCSP-OwAR	83.10 ± 2.90	11.33	74.78 ± 0.85	5.43	65.71 ± 0.39	4.24
PS-CSP-LDA	74.50 ± 8.01	−0.21	70.90 ± 4.63	−0.04	62.95 ± 2.47	−0.15
PS-CSP-CLDA	79.88 ± 4.53	7.00	72.28 ± 1.65	1.90	64.56 ± 0.72	2.40
PS-CSP-OwAR	80.21 ± 4.57	7.46	72.30 ± 1.80	1.93	64.55 ± 0.78	2.38
PS-CCSP-LDA	76.53 ± 4.25	2.52	72.38 ± 3.49	2.03	60.53 ± 2.24	−3.98
PS-CCSP-CLDA	78.57 ± 1.33	5.25	73.76 ± 0.57	3.98	62.97 ± 0.43	−0.11
PS-CCSP-OwAR	78.92 ± 1.30	5.72	74.15 ± 0.78	4.54	62.58 ± 0.41	−0.74
PS-RCSP-LDA	78.33 ± 5.08	4.93	73.01 ± 3.39	2.92	61.35 ± 2.17	−2.69
PS-RCSP-CLDA	79.72 ± 1.93	6.79	75.20 ± 1.27	6.01	63.09 ± 0.13	0.07
PS-RCSP-OwAR	79.74 ± 1.66	6.82	75.31 ± 1.26	6.16	63.01 ± 0.13	−0.05

algorithms should not directly use other users' raw EEG data. Thus, privacy-preserving TL (Xia, Deng et al., 2022) deserves more attention.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was supported by the Hubei Province Funds, China for Distinguished Young Scholars (2020CFA050), and the Key Laboratory of Brain Machine Collaborative Intelligence of Zhejiang Province, China (2020E10010-01).

References

Albalawi, H., & Song, X. (2012). A study of kernel CSP-based motor imagery brain computer interface classification. In *Proc. IEEE signal processing in medicine and biology symposium* (pp. 1–4). New York City, NY.

Barachant, A., Bonnet, S., Congedo, M., & Jutten, C. (2012). Multiclass brain-computer interface classification by Riemannian geometry. *IEEE Transactions on Biomedical Engineering*, *59*, 920–928.

Bin, G., Gao, X., Wang, Y., Hong, B., & Gao, S. (2009). VEP-based brain-computer interfaces: Time, frequency, and code modulations. *IEEE Computational Intelligence Magazine*, *4*, 22–26.

Blankertz, B., Dornhege, G., Krauledat, M., Muller, K. R., & Curio, G. (2007). The non-invasive Berlin brain-computer interface: Fast acquisition of effective performance in untrained subjects. *NeuroImage*, *37*, 539–550.

Blankertz, B., Tomioka, R., Lemm, S., Kawanabe, M., & Muller, K. R. (2008). Optimizing spatial filters for robust EEG single-trial analysis. *IEEE Signal Processing Magazine*, *25*, 41–56.

Chen, L.-L., Zhang, A., & Lou, X.-G. (2019). Cross-subject driver status detection from physiological signals based on hybrid feature selection and transfer learning. *Expert Systems with Applications*, *137*, 266–280.

Cui, Y., Xu, Y., & Wu, D. (2019). EEG-based driver drowsiness estimation using feature weighted episodic training. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *27*, 2263–2273.

Dai, M., Zheng, D., Liu, S., & Zhang, P. (2018). Transfer kernel common spatial patterns for motor imagery brain-computer interface classification. *Computational and Mathematical Methods in Medicine*, *2018*.

Delorme, A., & Makeig, S. (2004). EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods*, *134*, 9–21.

Edelman, B. J., Meng, J., Suma, D., Zurn, C., Nagarajan, E., Baxter, B., et al. (2019). Noninvasive neuroimaging enhances continuous neural tracking for robotic device control. *Science Robotics*, *4*.

van Erp, J., Lotte, F., & Tangermann, M. (2012). Brain-computer interfaces: Beyond medical applications. *Computer*, *45*, 26–34.

Fletcher, P. T., & Joshi, S. (2004). Principal geodesic analysis on symmetric spaces: Statistics of diffusion tensors. *Lecture Notes in Computer Science*, *3117*, 87–98.

General Data Protection Regulation (2016). Regulation (EU) 2016/679 of the European parliament and of the council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46. *Official Journal of the European Union*, *59*, 294.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. Boston, MA: MIT Press, http://www.deeplearningbook.org.

Gu, X., Cao, Z., Jolfaei, A., Xu, P., Wu, D., Jung, T.-P., et al. (2021). Eeg-based brain-computer interfaces (BCI): A survey of recent studies on signal sensing technologies and computational intelligence approaches and its applications. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *18*, 1645–1666.

He, B., Baxter, B., Edelman, B. J., Cline, C. C., & Ye, W. W. (2015). Noninvasive brain-computer interfaces based on sensorimotor rhythms. *Proceedings of the IEEE*, *103*, 907–925.

He, H., & Wu, D. (2020a). Different set domain adaptation for brain-computer interfaces: A label alignment approach. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *28*, 1091–1108.

He, H., & Wu, D. (2020b). Transfer learning for brain-computer interfaces: A Euclidean space data alignment approach. *IEEE Transactions on Biomedical Engineering*, *67*, 399–410.

Jayaram, V., Alamgir, M., Altun, Y., Scholkopf, B., & Grosse-Wentrup, M. (2016). Transfer learning in brain-computer interfaces. *IEEE Computational Intelligence Magazine*, *11*, 20–31.

Jiao, Y., Zhang, Y., Chen, X., Yin, E., Jin, J., Wang, X., et al. (2019). Sparse group representation model for motor imagery EEG classification. *IEEE Journal of Biomedical and Health Informatics*, *23*, 631–641.

Jolliffe, I. (2002). *Principal component analysis*. Wiley Online Library.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *Proc. 3rd int'l conf. on learning representations*. San Diego, CA.

Kononenko, I. (1994). Estimating attributes: Analysis and extensions of RELIEF. In *Proc. European Conf. on machine learning* (pp. 171–182). Catania, Italy.

Kostas, D., & Rudzicz, F. (2020). Thinker invariance: enabling deep neural networks for BCI across more people. *Journal of Neural Engineering*, *17*, Article 056008.

Krucoff, M. O., Rahimpour, S., Slutzky, M. W., Edgerton, V. R., & Turner, D. A. (2016). Enhancing nervous system recovery through neurobiologics, neural interface training, and neurorehabilitation. *Frontiers in Neuroscience*, *10*, 584.

Lagerlund, T. D., Sharbrough, F. W., & Busacker, N. E. (1997). Spatial filtering of multichannel electroencephalographic recordings through principal component analysis by singular value decomposition. *Journal of Clinical Neurophysiology*, *14*, 73–82.

Lance, B. J., Kerick, S. E., Ries, A. J., Oie, K. S., & McDowell, K. (2012). Brain-computer interface technologies in the coming decades. *Proceedings of the IEEE*, *100*, 1585–1599.

Lawhern, V. J., Solon, A. J., Waytowich, N. R., Gordon, S. M., Hung, C. P., & Lance, B. J. (2018). EEGNet: a compact convolutional neural network for EEG-based brain-computer interfaces. *Journal of Neural Engineering*, *15*, Article 056013.

Lin, C.-T., Chuang, C., Hung, Y., Fang, C., Wu, D., & Wang, Y.-K. (2021). A driving performance forecasting system based on brain dynamic state analysis using 4-D convolutional neural networks. *IEEE Transactions on Cybernetics*, *51*, 4959–4967.

Liyanage, S. R., Guan, C., Zhang, H., Ang, K. K., Xu, J., & Lee, T. H. (2013). Dynamically weighted ensemble classification for non-stationary EEG processing. *Journal of Neural Engineering*, *10*, Article 036007.

Long, M., Wang, J., Ding, G., Pan, S. J., & Yu, P. S. (2014). Adaptation regularization: A general framework for transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *26*, 1076–1089.

Long, M., Wang, J., Sun, J., & Philip, S. Y. (2015). Domain invariant transfer kernel learning. *IEEE Transactions on Knowledge and Data Engineering*, *27*, 1519–1532.

Lotte, F. (2015). Signal processing approaches to minimize or suppress calibration time in oscillatory activity-based brain-computer interfaces. *Proceedings of the IEEE*, *103*, 871–890.

Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., et al. (2018). A review of classification algorithms for EEG-based brain-computer interfaces: a 10 year update. *Journal of Neural Engineering*, *15*, Article 031005.

Lu, H., Eng, H.-L., Guan, C., Plataniotis, K. N., & Venetsanopoulos, A. N. (2010). Regularized common spatial pattern with aggregation for EEG classification in small-sample setting. *IEEE Transactions on Biomedical Engineering*, *57*, 2936–2946.

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579–2605.

Marathe, A. R., Lawhern, V. J., Wu, D., Slayback, D., & Lance, B. J. (2016). Improved neural signal classification in a rapid serial visual presentation task using active learning. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *24*, 333–343.

Martini, M. L., Oermann, E. K., Opie, N. L., Panov, F., Oxley, T., & Yaeger, K. (2020). Sensor modalities for brain-computer interface technology: a comprehensive literature review. *Neurosurgery*, *86*, E108–E117.

Mellinger, J., Schalk, G., Braun, C., Preissl, H., Rosenstiel, W., Birbaumer, N., et al. (2007). An MEG-based brain-computer interface (BCI). *Neuroimage*, *36*, 581–593.

Meng, L., Huang, J., Zeng, Z., Jiang, X., Yu, S., Jung, T.-P., et al. (2022). EEG-based brain-computer interfaces are vulnerable to backdoor attacks. *Engineering*, URL: https://www.researchsquare.com/article/rs-108085/v1. submitted for publication.

Moakher, M. (2005). A differential geometric approach to the geometric mean of symmetric positive-definite matrices. *SIAM Journal on Matrix Analysis and Applications*, *26*, 735–747.

Naseer, N., & Hong, K.-S. (2015). fNIRS-based brain-computer interfaces: a review. *Frontiers in Human Neuroscience*, *9*, 3.

Nicolas-Alonso, L. F., & Gomez-Gil, J. (2012). Brain computer interfaces, a review. *Sensors*, *12*, 1211–1279.

Oikonomou, V. P., Nikolopoulos, S., & Kompatsiaris, I. (2020). Robust motor imagery classification using sparse representations and grouping structures. *IEEE Access*, *8*, 98572–98583.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*, 1345–1359.

Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*, 1226–1238.

Pennec, X., Fillard, P., & Ayache, N. (2006). A Riemannian framework for tensor computing. *International Journal of Computer Vision*, *66*, 41–66.

Pfurtscheller, G., Müller-Putz, G. R., Scherer, R., & Neuper, C. (2008). Rehabilitation with brain-computer interface systems. *Computer*, *41*, 58–65.

Pfurtscheller, G., & Neuper, C. (2001). Motor imagery and direct brain-computer communication. *Proceedings of the IEEE*, *89*, 1123–1134.

Ramoser, H., Muller-Gerking, J., & Pfurtscheller, G. (2000). Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Transactions on Rehabilitation Engineering*, *8*, 441–446.

Rao, R. P. (2013). *Brain-computer interfacing: An introduction*. New York, NY: Cambridge University Press.

Rao, R. P., & Scherer, R. (2010). Chapter 10 - statistical pattern recognition and machine learning in brain-computer interfaces. In K. G. Oweiss (Ed.), *Statistical signal processing for neuroscience and neurotechnology* (pp. 335–367). Oxford: Academic Press.

Rivet, B., Souloumiac, A., Attina, V., & Gibert, G. (2009). xDAWN algorithm to enhance evoked potentials: application to brain-computer interface. *IEEE Transactions on Biomedical Engineering*, *56*, 2035–2043.

Rodrigues, P. L. C., Jutten, C., & Congedo, M. (2019). Riemannian procrustes analysis: Transfer learning for brain–computer interfaces. *IEEE Transactions on Biomedical Engineering*, *66*, 2390–2401.

Roy, R. N., Bonnet, S., Charbonnier, S., Jallon, P., & Campagne, A. (2015). A comparison of ERP spatial filtering methods for optimal mental workload estimation. In *Proc. 37th annual int'l conf. of the ieee engineering in medicine and biology society (EMBC)* (pp. 7254–7257).

Saha, S., Ahmed, K. I. U., Mostafa, R., Hadjileontiadis, L., & Khandoker, A. (2018). Evidence of variabilities in EEG dynamics during motor imagery-based multiclass brain–computer interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *26*, 371–382.

Schirrmeister, R. T., Springenberg, J. T., Fiederer, L. D. J., Glasstetter, M., Eggensperger, K., Tangermann, M., et al. (2017). Deep learning with convolutional neural networks for EEG decoding and visualization. *Human Brain Mapping*, *38*, 5391–5420.

Sellers, E. W., & Donchin, E. (2006). A P300-based brain-computer interface: initial tests by ALS patients. *Clinical Neurophysiology*, *117*, 538–548.

Shin, Y., Lee, S., Lee, J., & Lee, H.-N. (2012). Sparse representation-based classification scheme for motor imagery-based brain-computer interface systems. *Journal of Neural Engineering*, *9*, Article 056002.

Sitaram, R., Caria, A., Veit, R., Gaber, T., Rota, G., Kuebler, A., et al. (2007). fMRI brain-computer interface: a tool for neuroscientific research and treatment. *Computational Intelligence and Neuroscience*.

Steyrl, D., Scherer, R., Förstner, O., & Müller-Putz, G. R. (2014). Motor imagery brain-computer interfaces: random forests vs regularized LDA-non-linear beats linear. In *Proc. 6th int'l brain-computer interface conf.* (pp. 241–244). Graz, Austria.

Teplan, M. (2002). Fundamentals of EEG measurement. *Measurement Science Review*, *2*, 1–11.

Wang, P., Lu, J., Zhang, B., & Tang, Z. (2015). A review on transfer learning for brain-computer interface classification. In *Proc. 5th int'l conf. on information science and technology*. Changsha, China.

Wang, Y., & Wu, D. (2017). Real-time fMRI based brain computer interface: A review. In *Proc. 24th int'l. conf. on neural information processing*. Guangzhou, China.

Willett, F. R., Avansino, D. T., Hochberg, L. R., Henderson, J. M., & Shenoy, K. V. (2021). High-performance brain-to-text communication via handwriting. *Nature*, *593*, 249–254.

Wu, D. (2017). Online and offline domain adaptation for reducing BCI calibration effort. *IEEE Transactions on Human–Machine Systems*, *47*, 550–563.

Wu, D., King, J.-T., Chuang, C.-H., Lin, C.-T., & Jung, T.-P. (2018). Spatial filtering for EEG-based regression problems in brain-computer interface (BCI). *IEEE Transactions on Fuzzy Systems*, *26*, 771–781.

Wu, D., Lance, B. J., & Lawhern, V. J. (2014). Transfer learning and active transfer learning for reducing calibration data in single-trial classification of visually-evoked potentials. In *Proc. IEEE int'l conf. on systems, man, and cybernetics*. San Diego, CA.

Wu, D., Lawhern, V. J., Gordon, S., Lance, B. J., & Lin, C.-T. (2017). Driver drowsiness estimation from EEG signals using online weighted adaptation regularization for regression (OwARR). *IEEE Transactions on Fuzzy Systems*, *25*, 1522–1535.

Wu, D., Lawhern, V. J., Hairston, W. D., & Lance, B. J. (2016). Switching EEG headsets made easy: Reducing offline calibration effort using active wighted adaptation regularization. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *24*, 1125–1137.

Wu, D., Lawhern, V. J., Lance, B. J., Gordon, S., Jung, T.-P., & Lin, C.-T. (2017). EEG-based user reaction time estimation using Riemannian geometry features. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *25*, 2157–2168.

Wu, D., Xu, J., Fang, W., Zhang, Y., Yang, L., Luo, H., et al. (2022). Adversarial attacks and defenses in physiological computing: A systematic review. *National Science Open*, in press.

Wu, D., Xu, Y., & Lu, B.-L. (2022). Transfer learning for EEG-based brain-computer interfaces: A review of progress made since 2016. *IEEE Transactions on Cognitive and Developmental Systems*, *14*, 4–19.

Xia, K., Deng, L., Duch, W., & Wu, D. (2022). Privacy-preserving domain adaptation for motor imagery-based brain-computer interfaces. *IEEE Transactions on Biomedical Engineering*, in press.

Xia, K., Duch, W., Sun, Y., Xu, K., Fang, W., Luo, H., et al. (2022). Privacy-preserving brain-computer interfaces: A systematic review. *IEEE Transactions on Computational Social Systems*, in press.

Xu, L., Xu, M., Ke, Y., An, X., Liu, S., & Ming, D. (2020). Cross-dataset variability problem in EEG decoding with deep learning. *Frontiers in Human Neuroscience*, *14*, 103.

Yger, F., Berar, M., & Lotte, F. (2017). Riemannian approaches in brain-computer interfaces: a review. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *25*, 1753–1762.

Zanini, P., Congedo, M., Jutten, C., Said, S., & Berthoumieu, Y. (2018). Transfer learning: a Riemannian geometry framework with applications to brain-computer interfaces. *IEEE Transactions on Biomedical Engineering*, *65*, 1107–1116.

Zhang, W., Deng, L., Zhang, L., & Wu, D. (2022). A survey on negative transfer. *IEEE/CAA Journal of Automatica Sinica*, URL: https://arxiv.org/abs/2009.00909. Accepted.

Zhang, W., & Wu, D. (2020). Manifold embedded knowledge transfer for brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *28*, 1117–1127.

Zimmermann-Schlatter, A., Schuster, C., Puhan, M. A., Siekierka, E., & Steurer, J. (2008). Efficacy of motor imagery in post-stroke rehabilitation: a systematic review. *Journal of Neuroengineering and Rehabilitation*, *5*, 1–10.