Research Article

Supervised and Semisupervised Manifold Embedded Knowledge Transfer in Motor Imagery-Based BCI

Yilu Xu [ID],[1] **Hua Yin** [ID],[1] **Wenlong Yi** [ID],[1] **Xin Huang** [ID],[2] **Wenjuan Jian** [ID],[3] **Canhua Wang** [ID],[4] **and Ronghua Hu** [ID][5]

[1]*School of Software, Jiangxi Agricultural University, Nanchang 330045, China*
[2]*Software College, Jiangxi Normal University, Nanchang 330027, China*
[3]*School of Information Engineering, Nanchang University, Nanchang 330031, China*
[4]*School of Computer, Jiangxi University of Chinese Medicine, Nanchang 330004, China*
[5]*School of Advanced Manufacturing, Nanchang University, Nanchang 330031, China*

Correspondence should be addressed to Hua Yin; yinhua@jxau.edu.cn

A long calibration procedure limits the use in practice for a motor imagery (MI)-based brain-computer interface (BCI) system. To tackle this problem, we consider supervised and semisupervised transfer learning. However, it is a challenge for them to cope with high intersession/subject variability in the MI electroencephalographic (EEG) signals. Based on the framework of unsupervised manifold embedded knowledge transfer (MEKT), we propose a supervised MEKT algorithm (sMEKT) and a semisupervised MEKT algorithm (ssMEKT), respectively. sMEKT only has limited labelled samples from a target subject and abundant labelled samples from multiple source subjects. Compared to sMEKT, ssMEKT adds comparably more unlabelled samples from the target subject. After performing Riemannian alignment (RA) and tangent space mapping (TSM), both sMEKT and ssMEKT execute domain adaptation to shorten the differences among subjects. During domain adaptation, to make use of the available samples, two algorithms preserve the source domain discriminability, and ssMEKT preserves the geometric structure embedded in the labelled and unlabelled target domains. Moreover, to obtain a subject-specific classifier, sMEKT minimizes the joint probability distribution shift between the labelled target and source domains, whereas ssMEKT performs the joint probability distribution shift minimization between the unlabelled target domain and all labelled domains. Experimental results on two publicly available MI datasets demonstrate that our algorithms outperform the six competing algorithms, where the sizes of labelled and unlabelled target domains are variable. Especially for the target subjects with 10 labelled samples and 270/190 unlabelled samples, ssMEKT shows 5.27% and 2.69% increase in average accuracy on the two abovementioned datasets compared to the previous best semisupervised transfer learning algorithm (RA-regularized common spatial patterns-weighted adaptation regularization, RA-RCSP-wAR), respectively. Therefore, our algorithms can effectively reduce the need of labelled samples for the target subject, which is of importance for the MI-based BCI application.

1. Introduction

A brain-computer interface (BCI) system can build a direct communication pathway for a subject between his brain and an electrical device without the participation of his peripheral nerves and muscles [1, 2]. Due to safety and convenience, noninvasive electroencephalogram (EEG)-based BCIs have attracted great attention for decades [3]. Diverse EEG paradigms have been widely studied, such as motor imagery (MI), event-related potentials (ERP), steady-state visual evoked potentials (SSVEP), emotion state recognition, and driver drowsiness estimation [4–10].

In this paper, we focus on MI-based BCI, where people with severe neuromuscular disabilities can recover their neurological disorders by spontaneously performing imagined movements of the feet, hands, or tongue without external stimuli [8]. Furthermore, MI-based BCI gives able-bodied people a novel way to control an electrical

equipment [2]. Therefore, it is meaningful to study MI-based BCI.

However, EEG data analysis is challenging due to low signal-to-noise ratio and high artifacts [11, 12]. Moreover, a long calibration procedure hinders the development of MI-based BCI. Each subject usually spends a tedious calibration time in training a subject-specific classifier before performing the real-time MI tasks. Since the MI EEG signals are evoked by spontaneous movement imagination without external stimuli, they are of high intersession/subject variability. Thus, it is difficult to build a generic classifier that fits all sessions/subjects. Instead, it is realistic to train a subject-specific classifier that usually requires sufficient labelled data from the subject. Nevertheless, a long calibration procedure may unfortunately lead to high intersession differences and user frustration.

To cope with this problem, it is crucial for a target subject to reduce the need of amounts of labelled samples and effectively utilize the available samples. Rapid progress in machine learning motivates a variety of studies on how to make full use of the available samples [13–18]. EEG dataset reduction can reduce the feature dimensionality of the available EEG signals and improve the system learning speed. However, it cannot reduce the calibration time for the target subject [19, 20]. Likewise, different polynomials-based recurrence algorithms are promising techniques for signal processing due to their special capabilities in feature extraction. Nevertheless, they also just shorten the computational cost instead of calibration time [21–23]. Deep learning has been widely used in computer vision, natural language processing, and physiological signal analysis [24–28]. Nevertheless, deep learning needs lots of labelled samples from the target subject to show its superiority. The artificial data generation method can generate numerous artificial labelled data by recombining the few original labelled data in the time and frequency domains [29]. However, this method highly relies on the quantity and quality of the labelled samples. Transfer learning is a popular machine learning technique, which usually transfers labelled samples from different source sessions/subjects for a new target session/subject with no or few labelled samples [30]. Semisupervised learning can use the limited labelled set and comparably large unlabelled set from the same subject simultaneously [31]. Therefore, we pay more attention to transfer learning and its combination with semisupervised learning.

In general, transfer learning can be divided into three categories: supervised transfer learning, unsupervised transfer learning, and semisupervised transfer learning, depending on whether the samples from the target domain are all labelled, all unlabelled, or partially labelled and unlabelled. It is noted that all samples from the source domains are labelled no matter whether transfer learning is supervised, unsupervised, or semisupervised. Here, a domain means a subject or a session. Then, a labelled domain consists of the labelled samples from a subject/session, while an unlabelled domain includes the unlabelled samples from a subject/session. To our best knowledge, most studies focus on the supervised transfer learning since it can effectively use the discriminative labelled samples from the target domain

to select and adjust the labelled samples from the source domains. In fact, it is of great importance for unsupervised and semisupervised transfer learning algorithms to explore the geometric information embedded in the amounts of unlabelled samples from the target domain. As an unsupervised transfer learning algorithm, manifold embedded knowledge transfer (MEKT) performed Riemannian alignment (RA), tangent space mapping (TSM), and domain adaptation to continually minimize the differences among different domains in ERP-based and MI-based BCIs [32]. In our opinion, it is difficult to apply unsupervised transfer learning to MI-based BCI due to the existence of BCI illiteracy. It is better to collect initial labelled samples from the target domain.

Therefore, inspired by MEKT, we develop a supervised MEKT algorithm (sMEKT) and a semisupervised MEKT algorithm (ssMEKT) to explore more possibilities of MEKT in all cases and improve the efficiency of transfer learning by using all available samples. The main contributions of our work can be summarized as follows:

(1) We extend MEKT in the supervised and semi-supervised versions to further testify the effectiveness of transfer learning on the Riemannian manifold and its tangent space for MI-based BCI.

(2) We propose a supervised transfer learning algorithm (sMEKT) which transfers a large labelled source domain to a small labelled target domain by performing domain adaptation between the labelled domains.

(3) We present a semisupervised transfer learning algorithm (ssMEKT) which transfers a large labelled source domain into the labelled and unlabelled target domains by performing domain adaptation between the unlabelled target domain and all labelled domains.

The rest of this paper is structured as follows. In Section 2, we introduce the related work on supervised, unsupervised, and semisupervised transfer learning. In Section 3, we present the applied MI datasets and the detailed methods of EEG processing, including preprocessing, RA, TSM, all MEKT based algorithms, and a shrinkage linear discriminant analysis (sLDA) classifier. The results of classification accuracy and computation time are shown in Section 4. The experimental results are discussed in Section 5. Finally, our conclusions are drawn in Section 6.

2. Related Work

Besides MI, the other EEG paradigms mentioned above, such as SSVEP and ERP, also use transfer learning to reduce or suppress the calibration time. Thus, different EEG-based BCIs can learn from each other in terms of transfer learning.

Most transfer learning algorithms are supervised, which are designed to address the shortage of labelled samples from the target domain. In MI-based BCI, the common spatial patterns (CSP) method is a classical feature extraction algorithm for only one subject. However, it performs poorly in

the small labelled set scenario [33, 34]. Originated from CSP, regularized CSP (RCSP) calculated the regularized average spatial covariance matrix for each class by giving the labelled samples from the source and target domains' different regularization parameters [35]. Based on the framework of RCSP, different distance metrics, such as Frobenius norm and cosine distance, were used to measure the similarity between the labelled source and target domains [36, 37]. Combined CSP (CCSP) simply concatenated the labelled samples from the target domain and multiple source domains with equal weight to compute the spatial filters [38]. A dynamic time warping RCSP (DTW-RCSP) method performed domain adaptation by aligning the labelled source domains to the labelled target domain from the same class using an optimal warping path [39]. These RCSP algorithms inherited the advantage of the CSP in MI-based BCI. However, the regularization parameters, which were used to evaluate the differences between the labelled source and target domains, were often manually set or were obtained by means of cross-validation. Recently, since affine transformation can make the covariance matrices of EEG data from different domains close, RA-based supervised transfer learning algorithms have received widespread attention in different EEG-based BCIs [40–43]. Zanini first performed RA for each domain using the Riemannian mean of its resting trials as the reference matrix and then concatenated all aligned matrices from the labelled domains to train a minimum distance to mean (MDM) classifier based on Riemannian Gaussian distributions [44]. A Riemannian procrustes analysis (RPA) algorithm executed the following transformations (translation, scaling, and rotations) for the covariance matrices from the source and target domains to shorten their differences and then constructed an MDM classifier using the transformed matrices from the labelled domains [45]. Due to good performance of RA, our proposed sMEKT also belongs to the RA-based supervised transfer learning algorithms.

Many unsupervised transfer learning algorithms utilize the unlabelled samples from the target domain, as well as abundant labelled samples from the source domains to realize the zero-training for the target domain. In ERP-based BCI, Waytowich presented an unsupervised transfer learning method, where independent models were first trained using labelled samples from different source subjects, then the classification decisions of independent models were combined to classify unlabelled samples from the target subject, and finally each model's decision was weighted based on the inferred accuracy of direct classification [46]. Such method did not utilize the inherent information embedded in the unlabelled samples. In MI-based BCI, Xu proposed an unsupervised cross-dataset transfer learning, where EEGNet and ShallowConvNet were trained with the labelled source dataset, then an unsupervised domain adaptation was performed between the labelled source dataset and the unlabelled target dataset, and finally the pretrained model was validated on the unlabelled target dataset [47]. However, this algorithm did not realize the domain adaptation between the subjects. To minimize the differences among subjects, in ERP-based and MI-based BCIs, MEKT

first executed RA for the covariance matrices from different subjects, then extracted the tangent feature vectors in the TSM module, and finally performed joint probability distribution shift minimization, labelled source domain discriminability preservation, and unlabelled target domain locality preservation [32]. Our proposed algorithms are based on the framework of MEKT, since MEKT can not only shorten the differences between domains but also preserve the characteristics of the labelled source domain and unlabelled target domain as much as possible.

Theoretically, semisupervised transfer learning can provide more information than unsupervised transfer learning due to the existence of a labelled target domain. To achieve epileptic seizure classification from EEG signals, Jiang integrated transfer learning, semisupervised learning, and a Takagi-Sugeno-Kang fuzzy system [48]. This method achieved good performance at the cost of the computation time. In ERP-based BCI, Wu proposed online and offline weighted adaptation regularization (wAR) algorithms which performed domain adaptation between the labelled source domain and the entire target domain by integrating the loss function minimization, structural risk minimization, marginal conditional probability distribution adaptation, and conditional probability distribution adaptation [49]. Although wAR is a semisupervised transfer learning algorithm, its architect is similar to that of unsupervised MEKT.

In summarize, these approaches mentioned above inspired the design of our supervised and semisupervised transfer learning algorithms in MI-based BCI.

3. Materials and Methods

3.1. Description of Datasets. Two publicly available MI datasets were used to assess the effectiveness of our proposed transfer learning algorithms. More details about these datasets are described as follows:

(1) Dataset 1(BCI Competition III dataset IVa): this dataset contained 118-channel EEG signals from five healthy subjects (aa, al, av, aw, and ay). A visual cue on a computer screen was displayed for 3.5 s, during which the subject was instructed to perform one of the following MI tasks: left hand, right hand, and right foot. Only EEG signals evoked by the right hand and right foot MI tasks were provided for competition. A total of 140 trials per MI task were available for each subject. The EEG signals were band-pass filtered between 0.05 and 200 Hz and downsampled from 1000 Hz to 100 Hz [50].

(2) Dataset 2 (BCI Competition IV dataset 1): this dataset consisted of the calibration data and evaluation data from seven healthy subjects (a, b, c, d, e, f, and g). Only calibration data was used for our experiments because of complete marker information. Two classes of MI tasks could be chosen from the three classes (left hand, right hand, and foot) by each subject. Each subject was shown a visual cue for a period of 4 s and performed the cued MI task. 100 trials per MI task were collected for each subject. 59-

channel EEG signals were recorded at a sampling rate of 100 Hz and band-pass filtered between 0.05 and 200 Hz [51].

3.2. The Frameworks of Different MEKT Based Algorithms.

Our proposed supervised MEKT (sMEKT) and semi-supervised MEKT (ssMEKT) are based on the unsupervised MEKT. To better understand them, the frameworks of MEKT, sMEKT, and ssMEKT are shown in Figure 1.

The detailed steps of the different algorithms are outlined below:

(1) Preprocess the original EEG trials from different subjects. Note that the original EEG trials from all source subjects are labelled and used for all algorithms. For MEKT, all original EEG trials from the target subject are unlabelled. However, for sMEKT and ssMEKT, they are partitioned into the labelled and unlabelled EEG trials. The labelled ones are used for sMEKT and ssMEKT, whereas the unlabelled ones are only used for ssMEKT.

(2) Convert the filtered labelled/unlabelled EEG trials from each subject into the corresponding labelled/unlabelled covariance matrices. Then, we perform RA for these covariance matrices using their Riemannian mean as the reference matrix to obtain the corresponding aligned matrices in different MEKT based algorithms.

(3) In the TSM module, the aligned matrices are transformed into the Euclidean tangent feature vectors. The tangent feature vectors from different source subjects are concatenated into the labelled source tangent feature vector set, which is inputted into ssMEKT, MEKT, and sMEKT along with all, unlabelled, and labelled target tangent feature vectors, respectively. It is noted that the target tangent feature vectors are the feature vectors from the target subject.

(4) All MEKT based algorithms utilize the available tangent feature vector sets to yield different projection matrices which can be used to generate new lower dimensional feature sets.

(5) All new labelled feature sets from different subjects are fed into the sLDA classifier [52] to train a subject-specific model which is then used to classify the new unlabelled target feature set.

Next, we introduce the above procedures in detail.

3.3. Preprocessing of EEG Data.

For dataset 1, all original EEG trials from each subject were band-pass filtered between 8 and 30 Hz using a fifth-order Butterworth filter. Then, the filtered EEG trials were extracted from the time interval between 0.5 and 2.5 s after the visual cue signalling the start of imagery.

For dataset 2, all original EEG trials from each subject were spectrally filtered by a fiftieth order finite impulse response filter with cut-off frequencies of 8 and 30 Hz and temporally segmented from 0.5 to 3.5 s after the visual cue onset.

We only used channels in the central area of the brain where the sensorimotor rhythms (SMR) of MI are active. 25 and 29 channels were selected separately for dataset 1 and dataset 2. In Figure 2, the selected channels for the two datasets are marked in red.

3.4. Riemannian Alignment.

Although there are high intersubject variances in MI-based BCI, RA can make the marginal distributions of EEG trials from different subjects closer.

Let $X_i \in \mathbb{R}^{n \times t}$ be the ith filtered trial from a subject, where n is the number of channels and t is the number of sample points of the selected time window. The spatial covariance matrix of X_i can be defined as [40]

$$C_i = \frac{X_i X_i^{\mathrm{T}}}{t-1}. \tag{1}$$

Since the covariance matrices belong to a smooth Riemannian manifold of symmetric positive definite (SPD) matrices, they can be viewed as points on the manifold [44]. To show the benefits of RA, we first present some basic concepts of Riemannian geometry.

3.4.1. The Riemannian Distance.

Suppose that C_i and C_j are the points of the Riemannian manifold. The Riemannian distance $\delta(C_i, C_j)$ between these two points can be defined as the length of the shortest curve (named geodesic) connecting them [53]:

$$\delta(C_i, C_j) = \left\| \log \left(C_i^{-1} C_j \right) \right\|_F = \left[\sum_{k=1}^{n} \log^2 \lambda_k \right]^{1/2}, \tag{2}$$

with the Frobenius norm $\|\bullet\|_F$ and the eigenvalues $(\{\lambda_k\}_{k=1}^{n})$ of $C_i^{-1} C_j$.

3.4.2. The Riemannian Mean.

The Riemannian mean is usually used as the statistical descriptor of a set of SPD matrices on the manifold. Given N SPD matrices, their Riemannian mean M_R is defined as below [44]:

$$M_R = \underset{M}{\mathrm{argmin}} \sum_{k=1}^{N} \delta^2 (C_k, M). \tag{3}$$

For $N = 2$, M_R is the middle point of a geodesic connecting the two points. However, M_R can be effectively calculated by an iterative procedure for $N > 2$ [40].

3.4.3. Congruence Invariance.

Congruence invariance is an important property about the Riemannian distance, which means that the distance between the two points remains invariant after affine transformation using an invertible matrix as below [45]:

$$\delta(C_i, C_j) = \delta(\mathscr{W}^{\mathrm{T}} C_i \mathscr{W}, \mathscr{W}^{\mathrm{T}} C_j \mathscr{W}) \forall \mathscr{W} \in GL(n), \tag{4}$$
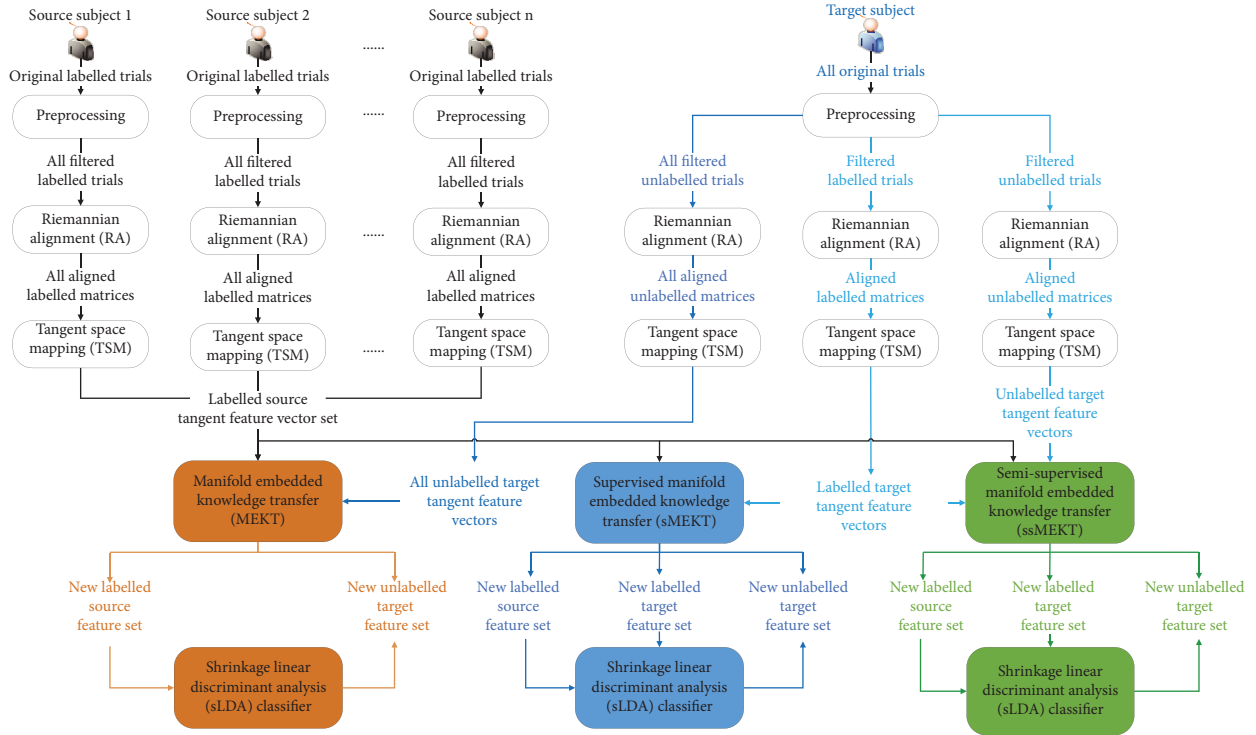
FIGURE 1: The frameworks of MEKT [32], sMEKT, and ssMEKT.

where $GL(n)$ is the set of invertible symmetric $n \times n$ square matrices.

Based on these concepts, RA executes the affine transformation for N points on the manifold using their Riemannian mean as the reference matrix. Then, the aligned matrix of C_i is [44]

$$\widehat{C}_i = M_R^{(-1/2)} C_i M_R^{(-1/2)}. \tag{5}$$

We perform RA for all covariance matrices from each domain using their own Riemannian mean as the reference matrix. Then, all aligned matrices from each domain are centred at the identity matrix. This property can be testified by the following [32]:

$$\mathcal{M}\left(M_R^{-1/2} C_1 M_R^{-1/2}, M_R^{-1/2} C_2 M_R^{-1/2}, \cdots, M_R^{-1/2} C_N M_R^{-1/2}\right)$$
$$= M_R^{-1/2} \mathcal{M}(C_1, C_2, \cdots, C_N) M_R^{-1/2} = M_R^{-1/2} M_R M_R^{-1/2} = I, \tag{6}$$

where $\mathcal{M}(C_1, C_2, \cdots, C_N)$ is the Riemannian mean operation and I is an identity matrix.

Thus, RA can make the aligned matrices from different domains comparable and preliminarily shorten their differences.

Additionally, as in [40], the filtered trial X_i can be spatially whitened by performing RA as follows:

$$\widehat{X}_i = M_R^{(-1/2)} X_i. \tag{7}$$

3.5. Tangent Space Mapping. Most traditional classifiers, such as linear discriminant analysis (LDA) and support vector machine (SVM), are designed for the Euclidean space, instead of the Riemannian space. To inherit the advantage of RA, we transform the aligned matrices into the Euclidean tangent feature vectors.

The SPD matrices lie in a differentiable Riemannian manifold. Their derivatives at a reference point on the manifold compose a tangent space. As mentioned in [54], the choice of Riemannian mean for the reference point leads to a tangent space locally approximate to the manifold. Figure 3 shows a Riemannian manifold and its tangent space at a Riemannian mean point M_R.

As shown in Figure 3, the logarithmic map projects C_i onto the tangent space at a Riemannian mean point M_R by [40]

$$T_i = \log_{M_R}(C_i) = M_R^{1/2} \log\left(M_R^{-1/2} C_i M_R^{-1/2}\right) M_R^{1/2}. \tag{8}$$

As in (6), the identity matrix I is the Riemannian mean of all aligned matrices from each domain. Then, the logarithmic mapping of the aligned matrix $\widehat{C}_i$ onto the normalized tangent space can be calculated as follows [55]:

$$\widehat{T}_i = \log_I(\widehat{C}_i) = \log_I\left(M_R^{-1/2} C_i M_R^{-1/2}\right)$$
$$= \log\left(M_R^{-1/2} C_i M_R^{-1/2}\right) = M_R^{-1/2} \log_{M_R}(C_i) M_R^{-1/2}. \tag{9}$$

FIGURE 2: The selected channels for two datasets. (a) Dataset 1. (b) Dataset 2.

FIGURE 3: A Riemannian manifold and its tangent space (this figure was adopted from Barachant et al. [40]).

To obtain a minimal representation, we vectorize the above logarithmic mapping $\widehat{T}_i$ by keeping its upper triangular part and applying unity weight for its diagonal elements and $\sqrt{2}$ weight for its nondiagonal elements [40]:

$$F_i = \text{vect}(\widehat{T}_i) = \left[\widehat{T}_{i1,1}; \sqrt{2}\,\widehat{T}_{i1,2}; \widehat{T}_{i2,2}; \sqrt{2}\,\widehat{T}_{i1,3}; \right.$$
$$\left. \sqrt{2}\,\widehat{T}_{i2,3}; \widehat{T}_{i3,3}; \cdots; \widehat{T}_{in,n} \right], \tag{10}$$

where $\widehat{T}_{i_{j,k}} \in \widehat{T}_i$. Then, the aligned matrix $\widehat{C}_i$ is transformed into the tangent feature vector F_i.

3.6. Manifold Embedded Knowledge Transfer.

The tangent feature vectors from different domains have similar marginal probability distributions inherited from the corresponding aligned matrices. However, their dimensionality $d = n \times (n + 1)/2$ is very high. To further reduce their differences and dimensionality, MEKT based algorithms aim to find optimal projection matrices for these tangent feature vectors.

As shown in Figure 1, all labelled tangent feature vectors from multiple source subjects are concatenated into a labelled source tangent feature vector set. For convenience, this set is called a labelled source domain $F_S = \left\{F_{S,i}\right\}_{i=1}^{n_S}$. Let

$F_{S,i}$ and n_S be the ith tangent feature vector and the size of F_S, respectively. Likewise, all tangent feature vectors from the target subject can be called a target domain $F_T = \left\{F_{T,i}\right\}_{i=1}^{n_T}$. Assume $F_{T,i}$ and n_T are the ith tangent feature vector and the size of F_T, respectively. For MEKT, F_T is unlabelled.

Since MEKT transfers a labelled source domain to an unlabelled target domain, it is an unsupervised transfer learning algorithm.

MEKT seeks the optimal projection matrix $P_S \in \mathbb{R}^{d \times q}$ for F_S and the optimal projection matrix $P_T \in \mathbb{R}^{d \times p}$ for F_T, which can not only make the lower dimensional features $P_S^T F_S$ and $P_T^T F_T$ close, but also preserve the labelled source domain discriminability and the unlabelled target domain locality. Note that $q \ll d$ is the dimensionality of a shared subspace for MEKT. MEKT sets $q = 10$. The following four properties are designed:

3.6.1. Joint Probability Distribution Shift Minimization.

The traditional maximum mean discrepancy (MMD) is usually used to shorten the marginal and conditional probability distribution discrepancies between different domains [56]. For simplicity, in MEKT, the joint probability

MMD is proposed to measure and minimize the joint probability distribution shift between the source and target domains as follows [32]:

$$\min_{P_S, P_T} \mathscr{D}_{S,T} = \min_{P_S, P_T} \mathscr{D}\left(Q(F_S, y_S), Q(F_T, \hat{y}_T)\right) = \min_{P_S, P_T} \mathscr{D}\left(Q(F_S|y_S)Q(y_S), Q(F_T|\hat{y}_T) \mid Q(\hat{y}_T)\right)$$

$$\approx \min_{P_S, P_T} \left(\sum_{k=1}^{2} \left\| \frac{1}{n_S} \sum_{i=1}^{n_S^k} P_S^T F_{S,i}^k - \frac{1}{n_T} \sum_{i=1}^{n_T^k} P_T^T F_{T,i}^k \right\|_F^2 \right) = \min_{P_S, P_T} \left\| N_S^T F_S^T P_S - N_T^T F_T^T P_T \right\|_F^2,$$

(11)

where $\mathscr{D}(\bullet, \bullet)$ is the joint probability distribution shift operation. $Q(\bullet)$, $Q(\bullet \mid \bullet)$, and $Q(\bullet, \bullet)$ are the marginal, conditional, and joint probability distributions, respectively. Assume y_S and $\hat{y}_T$ are the label vector of the labelled source domain and the pseudolabel vector of the unlabelled target domain, respectively. Let n_S^k and $F_{S,i}^k$ be the size and the ith tangent feature vector of the labelled source domain belonging to the kth class, respectively. Likewise, n_T^k and $F_{T,i}^k$ are the size and the ith tangent feature vector of the unlabelled target domain predicted to be the kth class, respectively. Here, only binary classification is considered. Then, $N_S = Y_S/n_S$ and $N_T = \hat{Y}_T/n_T$, where Y_S is the one-hot encoding matrix of y_S, and $\hat{Y}_T$ is the one-hot encoding matrix of $\hat{y}_T$. The one-hot encoding matrix will be $[1, 0; 0, 1; 0, 1]$, if its corresponding true/pseudolabel vector is [class 1; class 2; class 2].

3.6.2. Source Domain Discriminability Preservation.
The source domain discriminability can be defined by the within-class and between-class scatter matrices. Thus, after projection, it can be maintained by [32]

$$\min_{P_S} \text{tr}\left(P_S^T S_w^s P_S\right), \text{subject to: } P_S^T S_b^s P_S = I, \quad (12)$$

where $\text{tr}(\bullet)$ is the trace computation, $S_w^s = \sum_{k=1}^{2} F_S^k H_{n_S^k} F_S^k{}^T$ is the within-class scatter matrix of the labelled source domain, and $S_b^s = \sum_{k=1}^{2} n_S^k (\overline{m}_S^k - \overline{m}_S)(\overline{m}_S^k - \overline{m}_S)^T$ is the between-class scatter matrix of the labelled source domain, in which F_S^k, $H_{n_S^k} = I - \mathbf{1}_{n_S^k}/n_S^k$, $\overline{m}_S^k$ represent the labelled source domain belonging to class k, its centring matrix, and its mean, respectively, and $\overline{m}_S$ is the mean of the labelled source domain [57]. Note that $\mathbf{1}_{n_S^k} \in \mathbb{R}^{n_S^k \times n_S^k}$ is an all-one matrix.

3.6.3. Target Domain Locality Preservation.
Although the target domain is unlabelled, its local manifold structure can be formulated by the graph Laplacian matrix. MEKT constructs the normalized Laplacian graph $L = I - D^{-1/2} W D^{-1/2}$ [32], where

$$W_{ij} = \begin{cases} \exp\left(\dfrac{-\left\|F_{T,i} - F_{T,j}\right\|_2^2}{2\sigma^2} \right) & \text{if } F_{T,i} \in \text{Near}_K(F_{T,j}), \\ 0, & \text{otherwise} \end{cases}$$

$$D_{ij} = \begin{cases} \sum_{j=1}^{n_T} W_{ij}, & \text{if } i = j, \\ 0, & \text{otherwise} \end{cases}$$

(13)

in which σ is a scaling parameter and $\text{Near}_K(F_{T,j})$ is a set of K nearest neighbours of $F_{T,j}$ using the Euclidean metric. MEKT sets $\sigma = 1$ and $K = 10$.

To maintain the target domain locality after projection and remove the scaling effect, a graph regularization is minimized under the following constraints [32]:

$$\min_{P_T} \sum_{i,j=1}^{n_T} \left\| P_T^T F_{T,i} - P_T^T F_{T,j} \right\|_2^2 W_{ij} = \min_{P_T} \text{ tr}\left(P_T^T F_T L F_T^T P_T\right),$$

$$\text{subject to: } P_T^T F_T H_{n_T} F_T^T P_T = I,$$

(14)

where $H_{n_T} = I - \mathbf{1}_{n_T}/n_T$ is also a centring matrix.

3.6.4. Parameter Transfer and Regularization.
The following constraints are imposed on the projection matrices for better similarity and generalization performance [32]:

$$\min_{P_S, P_T} \left(\left\| P_S - P_T \right\|_F^2 + \left\| P_T \right\|_F^2 \right). \quad (15)$$

Then, the four properties above are integrated into an overall loss function of MEKT using different weights α, β, γ, and θ [32]:

$$\min_{P_S, P_T} \left(\alpha \mathscr{D}_{S,T} + \beta \text{tr}\left(P_S^T S_w^s P_S\right) + \gamma \text{tr}\left(P_T^T F_T L F_T^T P_T\right) + \theta \left(\left\| P_S - P_T \right\|_F^2 + \left\| P_T \right\|_F^2 \right) \right),$$

$$\text{subject to: } P_S^T S_b^s P_S = I, P_T^T F_T H_{n_T} F_T^T P_T = I,$$

(16)

where α, β, γ, and θ are manually set to be 1, 0.01, 0.1, and 20, respectively. For convenience, assume $P = [P_S; P_T]$ is an overall projection matrix ($P \in \mathbb{R}^{2d \times q}$). The Lagrange function is designed as below [32]:

$$\mathscr{T} = \mathrm{tr}\left(P^{\mathrm{T}}(\alpha A + \beta B + \gamma E + \theta G)P + \mu(I - P^{\mathrm{T}}JP)\right), \quad (17)$$

where

$$A = \begin{bmatrix} F_S N_S N_S^{\mathrm{T}} F_S^{\mathrm{T}} & -F_S N_S N_T^{\mathrm{T}} F_T^{\mathrm{T}} \\ -F_T N_T N_S^{\mathrm{T}} F_S^{\mathrm{T}} & F_T N_T N_T^{\mathrm{T}} F_T^{\mathrm{T}} \end{bmatrix}, B = \begin{bmatrix} S_w^s & 0 \\ 0 & 0 \end{bmatrix}, E = \begin{bmatrix} 0 & 0 \\ 0 & F_T L F_T^{\mathrm{T}} \end{bmatrix}$$

$$G = \begin{bmatrix} I & -I \\ -I & 2I \end{bmatrix}, J = \begin{bmatrix} S_b^s & 0 \\ 0 & F_T H_{n_T} F_T^{\mathrm{T}} \end{bmatrix}. \qquad (18)$$

To obtain the optimal P, MEKT sets the derivative of $\mathscr{T}$ to be 0 and then has

$$(\alpha A + \beta B + \gamma E + \theta G)P = \mu JP. \qquad (19)$$

Note that $\mu = 10^{-3}$ is also a weight. After generalized eigen-decomposition, P is comprised of q trailing eigen-vectors, where $q = 10$ is the dimensionality of the new feature. Consequently, P_S and P_T can be obtained from P. For the matrix A, N_T relates to the pseudolabel vector $\hat{y}_T$. Since $\hat{y}_T$ is unknown initially, it is set to be an all-zero vector first. At the next iteration, $\hat{y}_T$ is updated using sLDA as the classifier. MEKT performs five iterations in total.

3.7. The Proposed Supervised Manifold Embedded Knowledge Transfer.

Supervised manifold embedded knowledge transfer (sMEKT) is an extension of MEKT in the supervised version. For sMEKT, all tangent feature vectors from the target subject are divided into the labelled and unlabelled target domains. Only the labelled source and target domains are used to train a subject-specific classifier. The remaining unlabelled target domain is used to evaluate the performance of sMEKT.

To make full use of the available data, the following two problems should be taken into consideration for sMEKT:

(1) How to choose the appropriate regularization terms and constraints in the formulation of sMEKT?

(2) How to map the unlabelled target domain into the projected subspace since only the projection

matrices for the labelled source and target domains are obtained for sMEKT?

This section presents the corresponding solutions to the above questions. Let P_{TL} be the projection matrix of the labelled target domain.

First, we exclude the target domain locality preservation because the limited labelled target domain does not effectively show the entire geometric structure of the target domain with the absence of the unlabelled target domain.

Secondly, we keep the source domain discriminability preservation as in (12) and do not add the labelled target domain discriminability preservation as

$$\min_{P_{TL}} \mathrm{tr}\left(P_{TL}^{\mathrm{T}} S_w^{TL} P_{TL}\right),$$

$$\text{subject to: } P_{TL}^{\mathrm{T}} S_b^{TL} P_{TL} = I, \qquad (20)$$

where S_w^{TL} and S_b^{TL} are the within-class scatter matrix and between-class scatter matrix of the labelled target domain, respectively. The reason is that sMEKT aims at transferring the labelled source domain for the target domain. Therefore, the source domain discriminability should be taken into consideration, instead of the target domain discriminability.

In addition, the crucial work of MEKT is to minimize the joint probability distribution shift between the source and target domains. In sMEKT, since we only have the labelled source and target domains, the joint probability distribution shift minimization is updated as below:

$$\min_{P_S,P_{TL}} \mathscr{D}_{S,TL} = \min_{P_S,P_{TL}} \mathscr{D}\left(Q(F_S, y_S), Q(F_{TL}, y_{TL})\right) = \min_{P_S,P_{TL}} \mathscr{D}\left(Q(F_S \mid y_S)Q(y_S), Q(F_{TL} \mid y_{TL}) \mid Q(y_{TL})\right)$$

$$\approx \min_{P_S,P_{TL}} \left(\sum_{k=1}^{2} \left\| \frac{1}{n_S} \sum_{i=1}^{n_S^k} P_S^{\mathrm{T}} F_{S,i}^k - \frac{1}{n_{TL}} \sum_{i=1}^{n_{TL}^k} P_{TL}^{\mathrm{T}} F_{TL,i}^k \right\|_F^2 \right) = \min_{P_S,P_{TL}} \left\| N_S^{\mathrm{T}} F_S^{\mathrm{T}} P_S - N_{TL}^{\mathrm{T}} F_{TL}^{\mathrm{T}} P_{TL} \right\|_F^2, \qquad (21)$$

where y_{TL} and n_{TL} are the label vector and the size of the labelled target domain F_{TL}, respectively, n_{TL}^k and $F_{TL,i}^k$ are the size and the ith tangent feature vector of F_{TL} belonging to the kth class, and $N_{TL} = Y_{TL}/n_{TL}$, in which Y_{TL} is the one-hot encoding matrix of y_{TL}.

Finally, we remove the parameter transfer and regularization. In (15) and (16), MEKT pays more attention to minimization of the differences between P_S and P_T because

it sets $\theta = 20$. In our opinion, this minimization can further shorten the gaps between the labelled source domain and the unlabelled target domain, which will benefit the classification of the latter. If we have the similar constraint, $\min_{P_S,P_{TL}}(\|P_S - P_{TL}\|_F^2)$, and use the same weight on it, it may not play the same role as the constraint in (15) since the limited labelled target domain may not represent the remaining unlabelled target domain well. Calculating the

optimal θ with cross-validation can make up this limitation. However, it will yield computational burden.

Thus, the overall loss function of sMEKT can be formulated by

$$\min_{P_S, P_{TL}} \left(\alpha \mathscr{D}_{S,TL} + \beta \mathrm{tr}\left(P_S^\mathrm{T} S_w^s P_S \right) \right), \text{subject to: } P_S^\mathrm{T} S_b^s P_S = I. \quad (22)$$

Then, (23) can be solved by the same means as in (17). Since Y_{TL} in N_{TL} is the one-hot encoding matrix of the label vector y_{TL}, rather than that of the pseudolabel vector, we can obtain the optimal P without multiple iterations.

As for the second question presented above, we assume that the labelled and unlabelled target domains have similar joint probability distributions. Moreover, we minimize the joint probability distribution shift between the labelled source and target domains in (21). Thus, we define the average of the projection matrices of the labelled source and target domains as the projection matrix of unlabelled target domain:

$$P_{TU} = \frac{(P_S + P_{TL})}{2}. \quad (25)$$

Therefore, the new unlabelled target feature set is $P_{TU}^\mathrm{T} F_{TU}$, where F_{TU} is the unlabelled target domain.

3.8. The Proposed Semisupervised Manifold Embedded Knowledge Transfer.
Semisupervised manifold embedded knowledge transfer (ssMEKT) utilizes the labelled source and target domains, as well as the unlabelled target domain.

Let $P = [P_S; P_{TL}] (P \in \mathbb{R}^{2d \times q})$. Then, the corresponding Lagrange function is

$$\mathscr{T} = \mathrm{tr}\left(P^\mathrm{T} (\alpha A + \beta B) P + \mu \left(I - P^\mathrm{T} J P \right) \right), \quad (23)$$

where

$$A = \begin{bmatrix} F_S N_S N_S^\mathrm{T} F_S^\mathrm{T} & -F_S N_S N_{TL}^\mathrm{T} F_{TL}^\mathrm{T} \\ -F_{TL} N_{TL} N_S^T F_S^\mathrm{T} & F_{TL} N_{TL} N_{TL}^\mathrm{T} F_{TL}^\mathrm{T} \end{bmatrix}, B = \begin{bmatrix} S_w^s & 0 \\ 0 & 0 \end{bmatrix}, J = \begin{bmatrix} S_b^s & 0 \\ 0 & 0 \end{bmatrix}. \quad (24)$$

We construct the following regularization terms and constraints for ssMEKT.

First, we keep the target domain locality preservation because of the existence of the labelled and unlabelled target domains. Let $F_T = F_{TL} \cup F_{TU}$. It is noted that F_{TL} and F_{TU} are separately obtained after performing RA and TSM for their original domains. Then, we can minimize the graph regularization using all data from the target domain as in Section 3.6. Accordingly, we retain the projection matrix for the entire target domain, denoted as P_T.

Then, like sMEKT, the source domain discriminability preservation is considered.

Additionally, to benefit the classification of the unlabelled target domain, we reduce the joint probability distribution discrepancies between the labelled and unlabelled domains, instead of those between the source and target domains. Actually, MEKT also minimizes the joint probability distribution shift between the labelled and unlabelled domains since the source domain is labelled and the target domain is unlabelled. For ssMEKT, the joint probability distribution shift minimization can be rewritten as

$$\begin{aligned} \min_{P_S, P_T, P_{TL}} \mathscr{D}_{L,U} &= \min_{P_S, P_T, P_{TL}} \mathscr{D}\left(Q(F_S \cup F_{TL}, y_S \cup y_{TL}), Q(F_{TU}, \widehat{y}_{TU}) \right) \\ &= \min_{P_S, P_T, P_{TL}} \mathscr{D}\left(Q(F_S \cup F_{TL} \mid y_S \cup y_{TL}) Q(y_S \cup y_{TL}), Q(F_{TU} \mid \widehat{y}_{TU}) Q(\widehat{y}_{TU}) \right) \\ &\approx \min_{P_S, P_T, P_{TL}} \left(\sum_{k=1}^{2} \left\| \frac{1}{n_S} \sum_{i=1}^{n_S^k} P_S^\mathrm{T} F_{S,i}^k + \frac{1}{n_{TL}} \sum_{i=1}^{n_{TL}^k} P_{TL}^\mathrm{T} F_{TL,i}^k - \frac{1}{n_{TU}} \sum_{i=1}^{n_{TU}^k} P_T^\mathrm{T} F_{TU,i}^k \right\|_F^2 \right) \\ &= \min_{P_S, P_T, P_{TL}} \left\| N_S^\mathrm{T} F_S^\mathrm{T} P_S + N_{TL}^\mathrm{T} F_{TL}^\mathrm{T} P_{TL} - N_{TU}^\mathrm{T} F_{TU}^\mathrm{T} P_T \right\|_F^2, \end{aligned} \quad (26)$$

where $\widehat{y}_{TU}$ and n_{TU} are the pseudolabel vector and the size of the unlabelled target domain F_{TU}, respectively, $F_{TU,i}^k$ and n_{TU}^k are separately the ith tangent feature vector and the size of the unlabelled target domain predicted to be the kth class. Let $N_{TU} = \widehat{Y}_{TU}/n_{TU}$, in which $\widehat{Y}_{TU}$ is the one-hot encoding matrix of $\widehat{y}_{TU}$. For simplicity, P_T is temporally used as the projection matrix of the unlabelled

target domain since it relates to the unlabelled target domain.

Finally, we keep and update parameter transfer and regularization since there are abundant tangent feature vectors in the target domain. Then, we want the projection matrix P_T learned in the entire target domain to be similar to the projection matrix P_S learned in the source domain and to

be similar to the projection matrix P_{TL} learned in the labelled target domain. For better generalization performance, we avoid extreme values for these projection matrices. Therefore, we redefine the following constraints:

$$\min_{P_S, P_T, P_{TL}} \left(\|P_S - P_T\|_F^2 + \|P_{TL} - P_T\|_F^2 + \|P_T\|_F^2 \right). \tag{27}$$

$$\min_{P_S, P_T, P_{TL}} \left(\alpha \mathscr{D}_{L,U} + \beta \text{tr}\left(P_S^T S_w^s P_S\right) + \gamma \text{tr}\left(P_T^T F_T L F_T^T P_T\right) + \theta \left(\|P_S - P_T\|_F^2 + \|P_{TL} - P_T\|_F^2 + \|P_T\|_F^2 \right) \right), \tag{28}$$

$$\text{subject to: } P_S^T S_b^s P_S = I, P_T^T F_T H_{n_T} F_T^T P_T = I.$$

Given an overall projection matrix $P = [P_S; P_T; P_{TL}](P \in \mathbb{R}^{3d \times p})$, the corresponding Lagrange function can be reformulated as

After integrating the regularization terms and the constraints above, the overall loss function of ssMEKT can be formulated as follows:

$$\mathscr{T} = \text{tr}\left(P^T (\alpha A + \beta B + \gamma E + \theta G)P + \mu\left(I - P^T JP\right)\right), \tag{29}$$

where

$$A = \begin{bmatrix} F_S N_S N_S^T F_S^T & -F_S N_S N_{TU}^T F_{TU}^T & F_S N_S N_{TL}^T F_{TL}^T \\ -F_{TU} N_{TU} N_S^T F_S^T & F_{TU} N_{TU} N_{TU}^T F_{TU}^T & -F_{TU} N_{TU} N_{TL}^T F_{TL}^T \\ F_{TL} N_{TL} N_S^T F_S^T & -F_{TL} N_{TL} N_{TU}^T F_{TU}^T & F_{TL} N_{TL} N_{TL}^T F_{TL}^T \end{bmatrix}, B = \begin{bmatrix} S_w^s & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$E = \begin{bmatrix} 0 & 0 & 0 \\ 0 & F_T L F_T^T & 0 \\ 0 & 0 & 0 \end{bmatrix}, G = \begin{bmatrix} I & -I & 0 \\ -I & 3I & -I \\ 0 & -I & I \end{bmatrix}, J = \begin{bmatrix} S_b^s & 0 & 0 \\ 0 & F_T H_{n_T} F_T^T & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{30}$$

Then, we can obtain the optimal P, P_S, P_T and P_{TL} in the same way as MEKT and sMEKT. Like MEKT, N_{TU} is updated along with the change of $\widehat{Y}_{TU}$ at each iteration.

Finally, we choose and average the most relevant projection matrices P_T and P_{TL} for the projection matrix of unlabelled target domain:

$$P_{TU} = \frac{(P_T + P_{TL})}{2}. \tag{31}$$

3.9. Classification. As depicted in Figure 1, the tangent feature vector sets from different domains can be transformed into new feature sets by different MEKT based algorithms. However, only labelled feature sets are inputted to the supervised sLDA classifier to build a subject-specific model. For MEKT and ssMEKT, the pseudolabels of the new unlabelled target features marked by the model can be used to iteratively update the projection matrices and the model.

Additionally, the goal of LDA is to find an optimal hyperplane that can simultaneously maximize the between-class variances and minimize the within-class variances of the two-class projection data. To cope with high-dimensional data, sLDA uses a shrinkage estimate for the average

covariance matrix of each class in the LDA algorithm. More details can be seen in [52].

4. Experimental Results

4.1. Baseline Algorithms. We compared the following six baseline algorithms with various properties of transfer learning:

(1) CSP-LDA is the classical combination of feature extraction and classifier for MI. No source domain is used at all. Only the labelled target domain is used to design the CSP spatial filters and then to train the LDA classifier [33].

(2) RA-CCSP-LDA separately performs RA for the labelled source and target domains as in (7), then concatenates them with equal weight to calculate the CSP spatial filters, and finally inputs them into the LDA classifier [38].

(3) RA-RCSP-LDA is similar to RA-CCSP-LDA except for the way of generating the spatial filters. RCSP weights the labelled source and target domains using different regularization parameters [35]. To relieve the computational cost and give bigger weight to the labelled target domain, we manually set the two regularization parameters to be 0.1.

(4) RA-CCSP-wAR successively executes RA and CCSP before wAR. wAR is also a semisupervised transfer learning algorithm, which performs weighted domain adaptation between the labelled source domain and the entire target domain using SVM as the base classifier [49].

(5) RA-RCSP-wAR sequentially performs RA, RCSP, and wAR. For RA-CCSP-wAR and RA-RCSP-wAR, the hyperparameters of wAR were set according to its corresponding publication [49]. Note that three pairs of spatial filters were used for all spatial filtering-based algorithms in our experiments.

(6) MEKT-sLDA first performs unsupervised MEKT and then feeds the new labelled features from the source domain into the supervised sLDA classifier [32].

As mentioned above, our proposed sMEKT and ssMEKT also use the sLDA classifier. Moreover, we set the same values for the same weights and parameters in the MEKT based algorithms, such as $\alpha = 1$, $\beta = 0.01$, $\gamma = 0.1$, $\theta = 20$, $\mu = 10^{-3}$, $\sigma = 1$, $K = 10$, and $q = 10$. Additionally, both MEKT and ssMEKT perform five iterations to update their overall loss functions.

A summary of the six baseline algorithms and the two proposed algorithms is shown in Table 1.

4.2. Experimental Design. For each dataset, all trials from each target subject were randomly partitioned into two portions over twenty repetitions. The first portion was the labelled set to train a subject-specific classifier for the supervised and semisupervised algorithms, while the second portion was the unlabelled set to build the classifier for the unsupervised and semisupervised algorithms and to evaluate the effectiveness of different algorithms. All trials from the remaining source subjects were labelled and concatenated into a source domain to be transferred into a target domain. For each target subject, we varied the number of labelled trials from 10 to 50 with a step of 10 to investigate the robustness of all algorithms. For simplicity, denote the trial from the target subject and the trial from the source subject as the target trial and the source trial, respectively.

4.3. Classification Accuracy with Few Labelled and/or More Unlabelled Target Trials. The goal of our proposed algorithms is to achieve good classification performance even using few labelled target trials. Thus, we first conducted the experiments with few labelled and/or more unlabelled target trials. For the two MI datasets, ten labelled target trials, with equal number per class, were randomly selected over twenty repetitions. Then, 270 and 190 unlabelled target trials were separately available for dataset 1 and dataset 2. For each target subject, the classification accuracy of the unlabelled target trials was taken as an average of twenty repetitions. Detailed results of the two MI datasets are given in Tables 2 and 3. The bold-faced and italic numbers show the best and second-best classification accuracies, respectively.

Defining BCI illiteracy has been challenging because different researchers use different criteria to distinguish between good and bad subjects. Early work stated that 70% accuracy was effective for binary classification [58, 59]. In Table 2, the nontransfer learning algorithm CSP-LDA reaches or approaches a benchmark accuracy of 70% for subjects al, aw, and ay. Therefore, in our paper, subjects al, aw, and ay are grouped into good subjects, whereas subjects aa and av are grouped into bad ones. The transfer learning algorithms obviously improve the classification performance for bad subjects aa and av. Only MEKT-sLDA and ssMEKT-sLDA obtain satisfactory accuracy for bad subject av. Furthermore, sMEKT-sLDA outperforms the other supervised transfer learning algorithms RA-CCSP-LDA and RA-RCSP-LDA on average when the number of labelled target trials is as low as 10. RA-CCSP-wAR and RA-RCSP-wAR perform slightly higher than their corresponding supervised transfer learning algorithms. Our proposed ssMEKT-sLDA stands out itself among all semisupervised transfer learning algorithms.

In Table 3, CSP-LDA provides the benchmark accuracy of about 70% for subjects e, f, and g, leading to the following categorization in our paper: good subjects e, f, and g; bad subjects a, b, c, and d. The improvement in the classification performance of the transfer learning algorithms is substantial compared to CSP-LDA for bad subjects a, b, c, and d. However, sMEKT-sLDA yields worse classification performance than RA-CCSP-LDA and RA-RCSP-LDA on average. A possible explanation is that more than half of the subjects on dataset 2 perform MI tasks poorly. For sMEKT-sLDA, the target domain can be seriously affected by the bad source domain since the target and source domains are simultaneously adapted to make them close during the minimization of the joint probability distribution shift. However, for RA-CCSP-LDA and RA-RCSP-LDA, both the target and source domains are used to design the spatial filters without directly influencing each other. In addition, RA-CCSP-wAR and RA-RCSP-wAR still perform better than their corresponding supervised transfer learning algorithms.

As shown in Tables 2 and 3, ssMEKT-sLDA exhibits slightly higher classification performance than unsupervised MEKT-sLDA due to the existence of ten labelled target trials. In addition, for the target subjects with 10 labelled samples and 270/190 unlabelled samples, ssMEKT-sLDA shows a 5.27% and 2.69% increase in average accuracy on the two datasets compared to the best semisupervised algorithm RA-RCSP-wAR, respectively.

Then, we performed the paired-sample t-tests between the six baseline approaches and our proposed algorithms on the two datasets to further check if the performance differences among all algorithms were significant. The p-values are shown in Table 4.

The paired-sample t-tests show that the results of our proposed algorithms are statistically higher than those of CSP-LDA ($p < 0.005$). Although our proposed algorithms are based on MEKT-sLDA, their performance differences are very big. In most cases, ssMEKT-sLDA shows its superiority in terms of the p-values, especially on dataset 1. The

TABLE 1: A summary of the eight algorithms with various properties.

	Spatial filtering	RA	Transfer learning	Supervised	Semisupervised	Unsupervised
CSP-LDA [33]	√	—	—	√	—	—
RA-CCSP-LDA [38]	√	√	√	√	—	—
RA-RCSP-LDA [35]	√	√	√	√	—	—
RA-CCSP-wAR [49]	√	√	√	—	√	—
RA-RCSP-wAR [49]	√	√	√	—	√	–
MEKT-sLDA [32]	—	√	√	—	—	√
sMEKT-sLDA	—	√	√	√	—	—
ssMEKT-sLDA	—	√	√	—	√	—

TABLE 2: Classification accuracy (%) with 10 labelled target trials and/or 270 unlabelled target trials on dataset 1, for CSP-LDA [33], RA-CCSP-LDA [38], RA-RCSP-LDA [35], RA-CCSP-wAR [49], RA-RCSP-wAR [49], MEKT-sLDA [32], and our proposed algorithms.

	aa	al	av	aw	ay	Mean ± std.
CSP-LDA	63.39	82.89	60.06	68.61	72.54	69.50 ± 8.89
RA-CCSP-LDA	72.17	93.06	64.31	64.76	87.41	76.34 ± 13.21
RA-RCSP-LDA	70.93	95.24	65.04	68.50	88.91	77.72 ± 13.45
RA-CCSP-wAR	72.89	92.70	62.72	68.56	88.35	77.04 ± 12.92
RA-RCSP-wAR	70.76	95.06	65.41	71.24	89.37	78.37 ± 13.00
MEKT-sLDA	*75.94*	*96.35*	*71.94*	*75.26*	*90.48*	*82.00* ± 10.74
sMEKT-sLDA	72.52	93.72	67.94	73.69	87.22	79.02 ± 10.92
ssMEKT-sLDA	**76.06**	**96.52**	**72.59**	**76.85**	**90.50**	**82.50** ± 10.39

TABLE 3: Classification accuracy (%) with 10 labelled target trials and/or 190 unlabelled target trials on dataset 2, for CSP-LDA [33], RA-CCSP-LDA [38], RA-RCSP-LDA [35], RA-CCSP-wAR [49], RA-RCSP-wAR [49], MEKT-sLDA [32], and our proposed algorithms.

	a	b	c	d	e	f	g	Mean ± std.
CSP-LDA	65.50	52.37	56.05	61.92	70.34	69.97	70.68	63.83 ± 7.36
RA-CCSP-LDA	64.79	77.03	75.13	**78.50**	**92.45**	76.82	88.42	79.02 ± 9.09
RA-RCSP-LDA	69.37	76.87	74.68	*77.84*	92.21	77.47	87.61	79.44 ± 7.82
RA-CCSP-wAR	67.11	**77.58**	75.63	77.71	*92.32*	76.95	89.47	79.54 ± 8.62
RA-RCSP-wAR	69.79	*77.42*	74.79	*77.84*	92.24	77.53	88.95	79.79 ± 7.94
MEKT-sLDA	*83.24*	72.63	*76.08*	71.71	91.55	*85.50*	*91.61*	*81.76* ± 8.42
sMEKT-sLDA	71.89	64.16	60.47	68.89	84.13	74.74	88.26	73.22 ± 10.11
ssMEKT-sLDA	**83.76**	72.66	**76.11**	71.92	91.47	**85.74**	**91.95**	**81.94** ± 8.46

TABLE 4: The *p*-values in paired-sample *t*-tests on the two datasets between our proposed algorithms and CSP-LDA [33], RA-CCSP-LDA [38], RA-RCSP-LDA [35], RA-CCSP-wAR [49], RA-RCSP-wAR [49], and MEKT-sLDA [32].

	Dataset 1		Dataset 2	
	sMEKT-sLDA	ssMEKT-sLDA	sMEKT-sLDA	ssMEKT-sLDA
CSP-LDA	0.0040	0.0011	0.0027	0.00002
RA-CCSP-LDA	0.1903	0.0246	0.0954	0.4091
RA-RCSP-LDA	0.3813	0.0310	0.0444	0.3883
RA-CCSP-wAR	0.2179	0.0230	0.0573	0.4536
RA-RCSP-wAR	0.5493	0.0264	0.0325	0.4513
MEKT-sLDA	0.0020	0.1546	0.0025	0.0570

performance differences between sMEKT-sLDA and other transfer learning algorithms are small when sMEKT-sLDA performs well.

4.4. Classification Accuracy with Varying Numbers of Labelled and/or Unlabelled Target Trials.

To show the effect of the size of the target domain, with varying numbers of labelled and/or unlabelled target trials, the classification accuracies of

different subjects from different datasets and their means are shown in Figures 4 and 5.

As depicted in Figures 4(b), 4(d), and 4(e), for good subjects al, aw, and ay, CSP-LDA shows better classification performance than the supervised and semisupervised transfer learning algorithms with the increase of the labelled target trials. As shown in Table 2, these subjects perform well even using few labelled target trials. Thus, as their labelled target trials increase, their increasing between-class
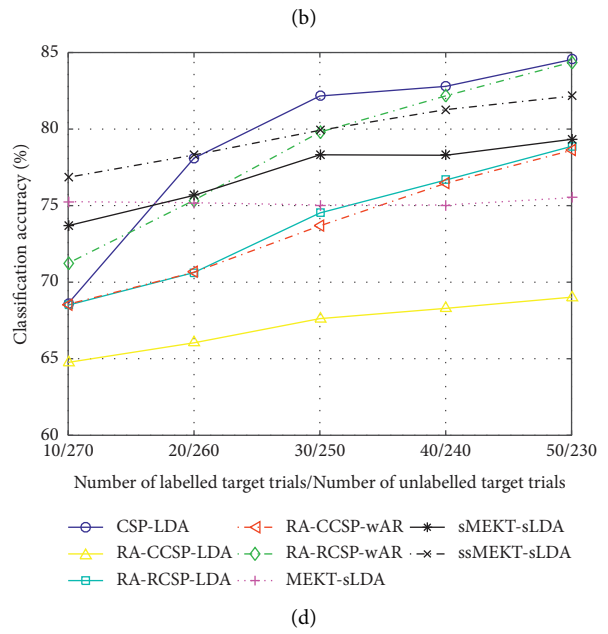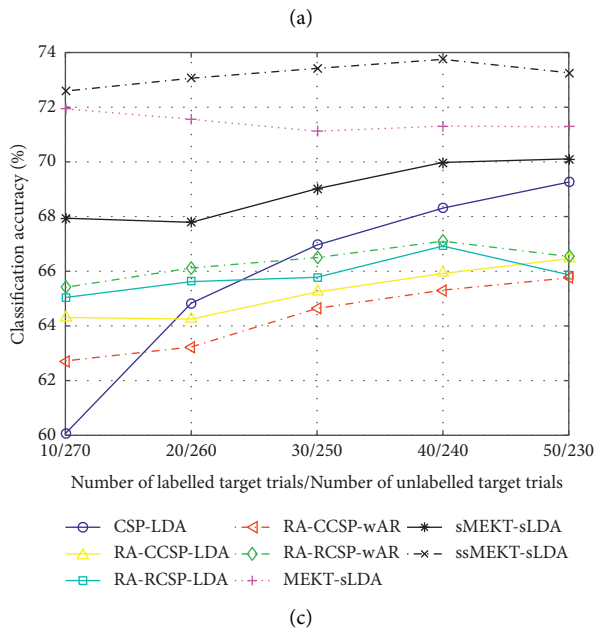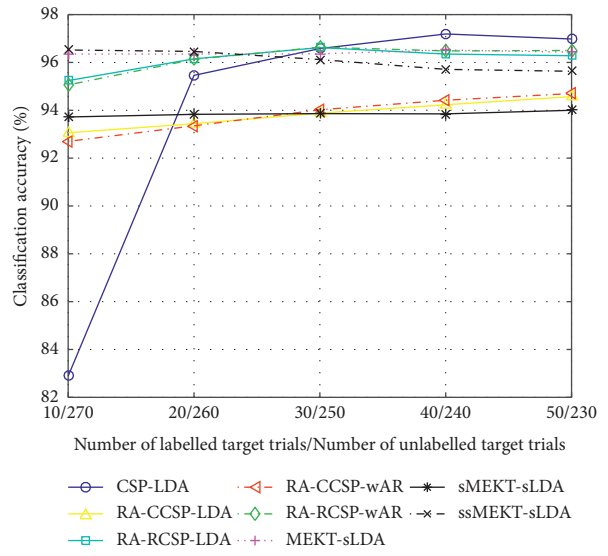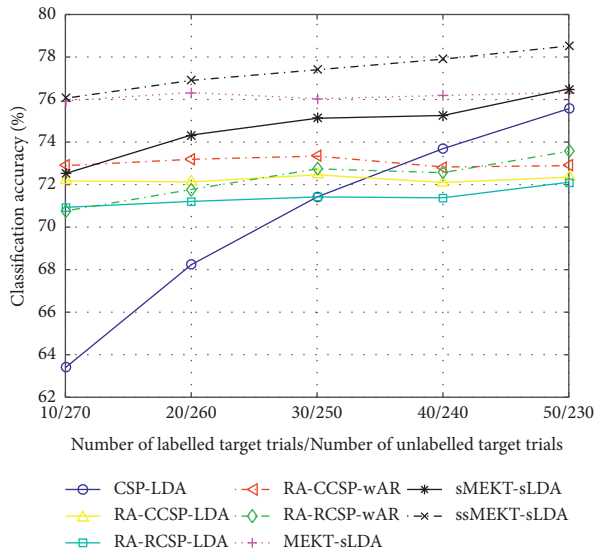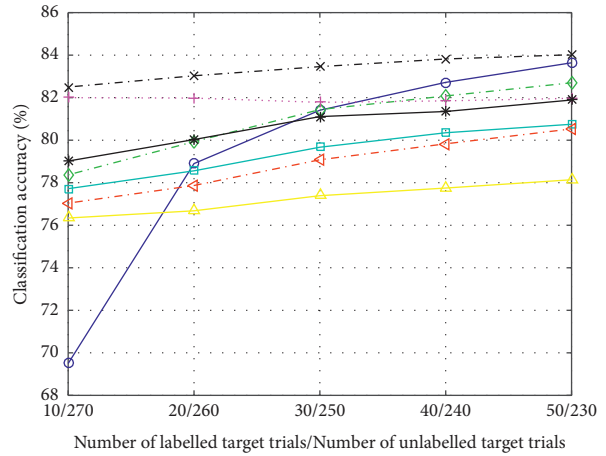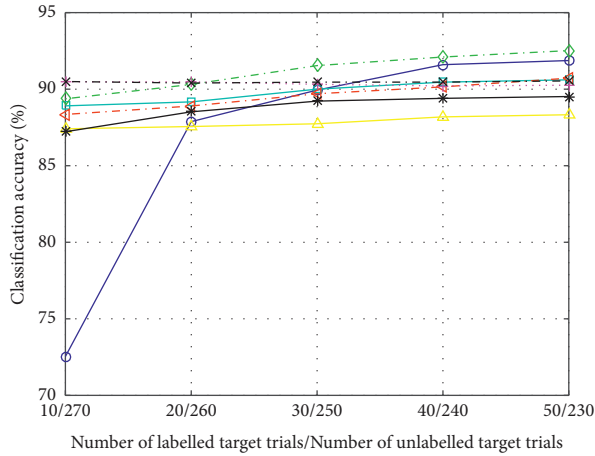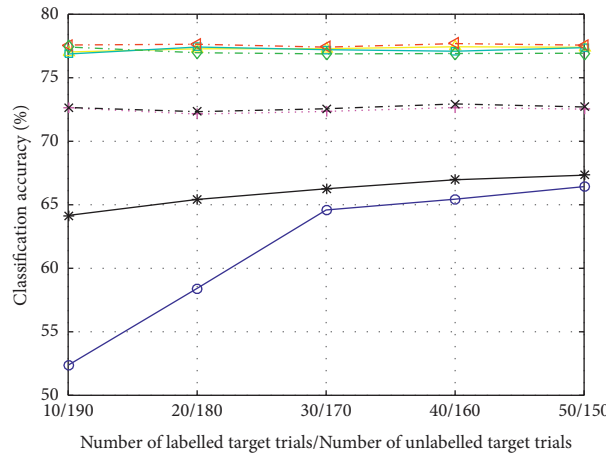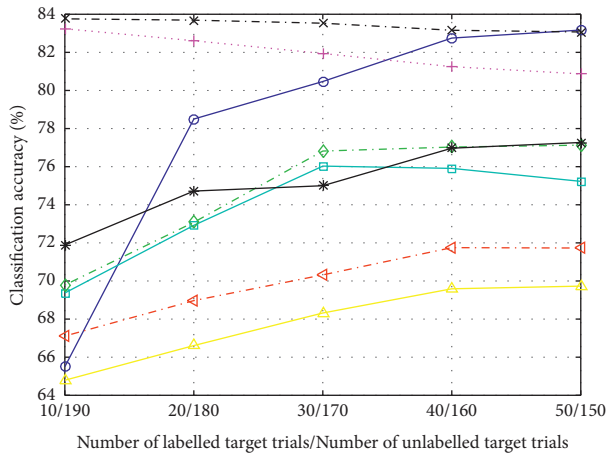
(a)

(b)

(c)
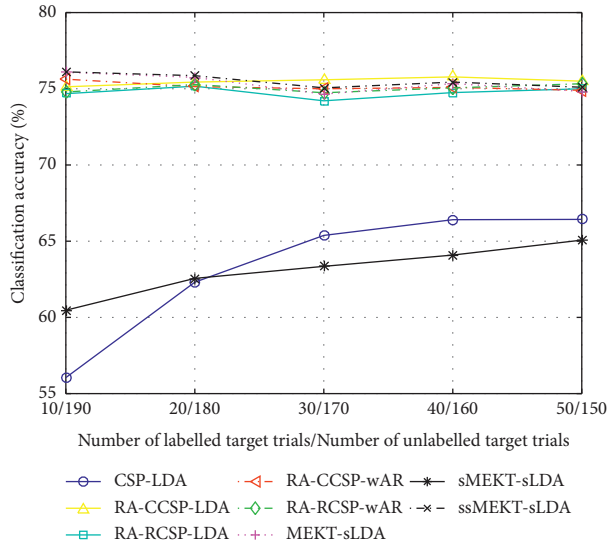
(d)

Figure 4: Continued.

(e)

(f)

FIGURE 4: Classification accuracy (%), with varying numbers of labelled and/or unlabelled target trials on dataset 1, for CSP-LDA [33], RA-CCSP-LDA [38], RA-RCSP-LDA [35], RA-CCSP-wAR [49], RA-RCSP-wAR [49], MEKT-sLDA [32], and our proposed algorithms. (a) Subject aa. (b) Subject al. (c) Subject av. (d) Subject aw. (e) Subject ay. (f) All subjects.
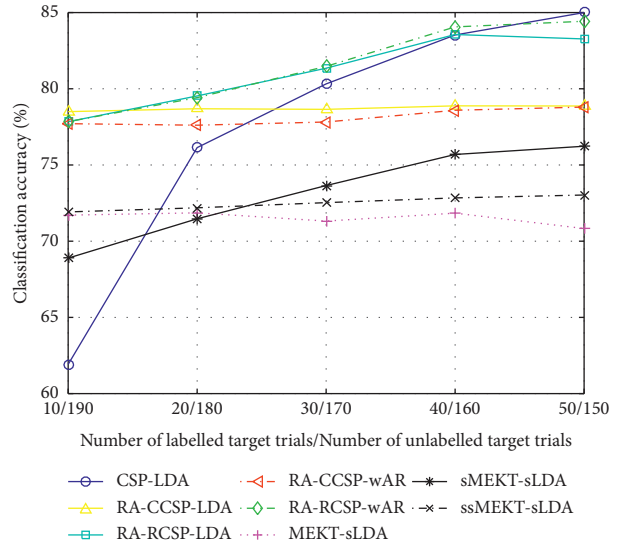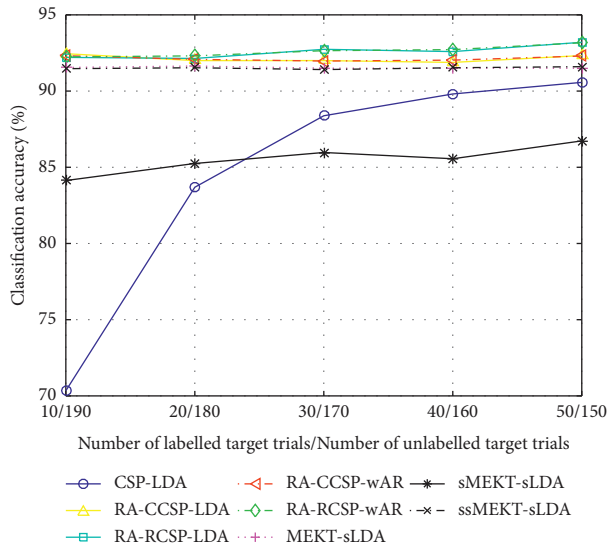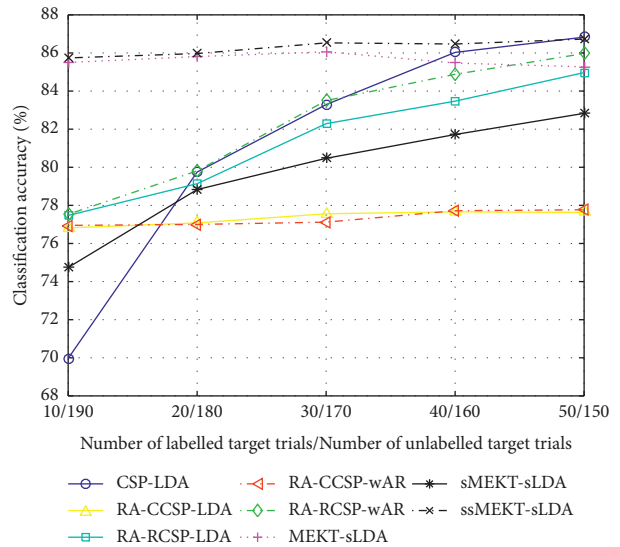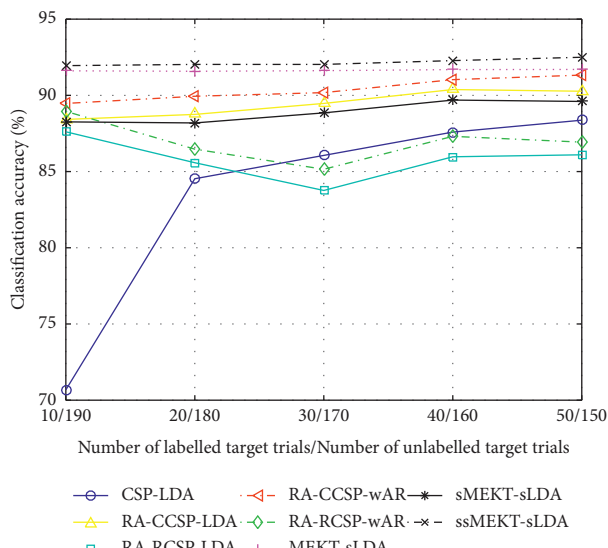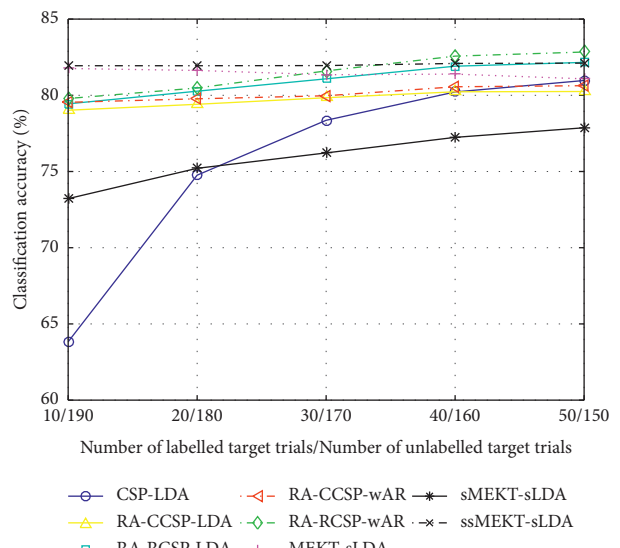

(a)

(b)

FIGURE 5: Continued.

Figure 5: Classification accuracy (%), with varying numbers of labelled and/or unlabelled target trials on dataset 2, for CSP-LDA [33], RA-CCSP-LDA [38], RA-RCSP-LDA [35], RA-CCSP-wAR [49], RA-RCSP-wAR [49], MEKT-sLDA [32], and our proposed algorithms. (a) Subject a. (b) Subject b. (c) Subject c. (d) Subject d. (e) Subject e. (f) Subject f. (g) Subject g. (h) All subjects.

discriminability can gradually reduce their dependence on transfer learning. As illustrated in Figures 4(a) and 4(c), for bad subjects aa and av, most transfer learning algorithms outperform CSP-LDA in most cases. It is necessary for them to transfer the source labelled trials due to their poor between-class discriminability. In Figures 4(a), 4(b), 4(c), and 4(e), the average performance of sMEKT-sLDA is higher than that of RA-RCSP-LDA for bad subjects aa and av, while RA-RCSP-LDA performs better than sMEKT-sLDA on average for good subjects al and ay. The reason is that much more good source subjects are available for bad target subjects aa and av, compared to good target subjects al and ay. Bad target subjects aa and av benefit from the domain adaptation used in sMEKT-sLDA, while good target subjects al and ay benefit from their bigger weights used in RA-RCSP-LDA. As shown in Figure 4(f), for all subjects, the classification performance of MEKT-sLDA decreases due to the reduction of unlabelled target trials. On average, ssMEKT-sLDA shows its compelling validity.

As shown in Figures 5(e)–5(g), for good subjects e, f, and g, due to their good between-class discriminability, as the labelled target trials increase, the classification performance of CSP-LDA is close to that of the transfer learning algorithms. As depicted in Figures 5(b) and 5(c), for bad subjects b and c, due to their poor between-class discriminability, the transfer learning algorithms maintain obvious advantages over CSP-LDA with the increasing labelled target trials. As illustrated in Figures 5(a)–5(h), on average, sMEKT-sLDA performs worse than most supervised transfer learning algorithms since a poor source domain might affect the discriminability of the target domain during domain adaptation. For subjects a, d, e, and f, RA-RCSP-wAR achieves higher results than RA-CCSP-wAR in most cases. All semisupervised transfer learning algorithms reach better performance than their supervised or unsupervised counterparts. Moreover, ssMEKT-sLDA shows its superiority for four out of seven subjects.

4.5. Computational Cost.

All experimental results were obtained from our operating platform:

(1) Hardware: processor: Intel i5 CPU@1.60 GHz; RAM: 8 GB.

(2) Software: Windows 10 Home Edition; Matlab 2015a.

The computation times of the different algorithms on two datasets using 10 labelled and/or 270/190 unlabelled target trials are shown in Table 5. The best performance was highlighted in bold.

As shown in Table 5, CSP-LDA spends the shortest computation time among all algorithms. The computational cost of sMEKT-sLDA is slightly higher than that of RA-RCSP-LDA due to higher-dimensional features. ssMEKT-sLDA requires more time than other MEKT based algorithms because of more available trials. However, for all semisupervised transfer learning algorithms, even with higher-dimensional features, ssMEKT-sLDA runs faster than RA-CCSP-wAR and RA-RCSP-wAR on average. The possible reason is that wAR works more complicatedly than ssMEKT.

TABLE 5: Computation times (seconds) of CSP-LDA [33], RA-CCSP-LDA [38], RA-RCSP-LDA [35], RA-CCSP-wAR [49], RA-RCSP-wAR [49], MEKT-sLDA [32], and our proposed algorithms on two datasets.

	Dataset 1	Dataset 2	Mean ± std.
CSP-LDA	**0.16**	**0.03**	**0.10** ± 0.09
RA-CCSP-LDA	1.36	0.29	0.83 ± 0.76
RA-RCSP-LDA	0.28	0.31	0.30 ± 0.02
RA-CCSP-wAR	6.56	4.50	5.53 ± 1.46
RA-RCSP-wAR	5.42	4.57	5.00 ± 0.60
MEKT-sLDA	1.26	2.60	1.93 ± 0.95
sMEKT-sLDA	0.45	0.50	0.48 ± 0.04
ssMEKT-sLDA	2.47	4.74	3.61 ± 1.61

5. Discussion

In this section, we discuss the experimental results from various aspects.

5.1. Effectiveness of Riemannian Alignment and Transfer Learning.

In our experiments, all algorithms perform RA and transfer learning, except CSP-LDA. They first execute RA for different domains in an unsupervised way, which can not only make different domains comparable, but also overcome the impact of limited labelled target trials. For all spatial filtering-based transfer learning algorithms, such as RA-CCSP-LDA, RA-RCSP-LDA, RA-CCSP-wAR, and RA-RCSP-wAR, the EEG trials are whitened by performing RA. Likewise, for all MEKT based transfer learning algorithms, the tangent feature vectors from different subjects are close to each other due to RA. Thus, RA can shorten the differences between domains, which is beneficial to successive transfer learning. As mentioned above, different transfer learning algorithms combine different domains in different ways. In Tables 2 and 3, even using few labelled and/or more unlabelled target trials, with the help of abundant labelled source trials, the supervised, unsupervised, and semisupervised transfer learning algorithms demonstrate better average classification accuracies than CSP-LDA. It is implied that both RA and transfer learning contribute to the good classification performance.

However, as shown in Figures 4 and 5, for good target subjects al, aw, ay, e, f, and g, with the increase of their labelled trials, their between-class discriminability can increase; thus the classification accuracies of CSP-LDA are gradually close to, or even higher than those of transfer learning algorithms. In contrast, for bad target subjects aa, av, b, and c, the transfer learning algorithms always outperform CSP-LDA as the labelled target trials increase. Therefore, it is implied that the number of labelled target trials and the extent of between-class discriminability of the source subjects greatly affect the performance of transfer learning algorithms.

5.2. Differences between Spatial Filtering and Manifold Embedded Knowledge Transfer.

The CSP features used in all spatial filtering-based transfer learning algorithms have lower dimensions than the transformed tangent space

vectors used in all MEKT based transfer learning algorithms. To keep or highlight the importance of the target domain, for CCSP based algorithms, the weight of the target domain is the same as that of the source domain, and for RCSP based algorithms, the weight of the target domain is higher than that of the source domain. Consequently, the source and target domains independently play their roles in spatial filtering-based algorithms. However, to further reduce the differences between the source and target domains, all MEKT based transfer learning algorithms pay great attention to the joint probability distribution shift minimization between domains. Therefore, the target domain is easily affected by the source domain during the minimization. The degree of positive transfer depends on the performance of the source domain.

As shown in Tables 2–4, most MEKT based transfer learning algorithms show better performance than the spatial filtering-based ones. However, for dataset 2, the average classification accuracy of sMEKT-sLDA is inferior to that of spatial filtering-based transfer learning algorithms, but still superior to that of CSP-LDA. The possible reason is that only three out of seven subjects perform MI tasks well on dataset 2. Thus, the poor source domain affects the positive transfer of sMEKT-sLDA. Thus, our supervised transfer learning algorithm sMEKT-sLDA should identify the most suitable source subjects instead of all available source subjects. In addition, due to the existence of unlabelled target domain, MEKT-sLDA and ssMEKT-sLDA reduce the negative impact of the poor source domain. As given in Table 5, in terms of computation time, all supervised transfer learning algorithms run efficiently. Furthermore, ssMEKT-sLDA takes shorter time than RA-CCSP-wAR and RA-RCSP-wAR. Overall, our proposed MEKT based algorithms provide comparably good performance with efficient running time.

5.3. Impact of Labelled and/or Unlabelled Target Trials. To investigate the roles of labelled and unlabelled target trials, all algorithms can be divided into supervised, unsupervised, and semisupervised algorithms. As shown in Tables 2 and 3, with abundant unlabelled target trials, MEKT-sLDA outperforms spatial filtering-based algorithms and sMEKT-sLDA on average. Thus, a large unlabelled target domain is beneficial for classification due to its embedded geometric structure. Moreover, with the help of a few labelled target trials, the performance of ssMEKT-sLDA is slightly better than that of MEKT. As illustrated in Figures 4 and 5, with increasing number of the labelled target trials and decreasing number of the unlabelled target trials, the performance improvement of ssMEKT over MEKT increases. As depicted in Figures 4(f) and 5(h), the curve of CSP-LDA grows fast with the increase of the labelled target trials. It is implied that the labelled target trials are crucial for the classification performance. Since the labelled and unlabelled target trials are less than the labelled source trials, the performance improvements of supervised and semisupervised transfer learning are not apparent as the labelled target trials increase.

Additionally, as shown in Figures 4 and 5, for good target subjects al, ay, e, f, and g, their classification accuracies of ssMEKT-sLDA are always similar to those of MEKT-sLDA. The possible reason is that both the unlabelled and labelled target trials of good target subjects can provide important information. However, for bad target subjects aa, av, a, and d, the differences between ssMEKT-sLDA and MEKT-sLDA are comparably obvious. The possible explanation is that the labelled target trials of bad target subjects can provide more valuable information than their unlabelled target trials. Overall, labelled target trials play an important role in the classification. The unlabelled target trials are also beneficial for classification, especially for good target subjects.

It is noted that our proposed ssMEKT-sLDA does not utilize the unlabelled target trials to train the classifier, resulting in the limited performance improvement.

6. Conclusion

To shorten the calibration time of the target subject, we propose a supervised MEKT algorithm (sMEKT) and a semisupervised MEKT algorithm (ssMEKT) in MI-based BCI. Both are combined with the sLDA classifier. Due to high intersubject variability, it is better to build a subject-specific classifier rather than a generic classifier. Both sMEKT and ssMEKT transfer abundant labelled samples from multiple source subjects for a specific target subject. First, they perform RA in an unsupervised way to preliminarily reduce the differences among different subjects. Then, they convert the aligned covariance matrices from different subjects into the corresponding tangent feature vectors for classification in the Euclidean space. Finally, to further cope with variations among different subjects, sMEKT performs domain adaptation between the labelled source and target domains, whereas ssMEKT performs domain adaptation between the labelled source domain and the entire target domain. During adaptation, both sMEKT and ssMEKT not only minimize the joint probability distribution shift among different domains, but also maintain the source domain discriminability as much as possible. Moreover, ssMEKT keeps the entire target domain locality to utilize its geometric structure. In addition, ssMEKT performs the parameter transfer and regularization, to make the projection matrices of different domains closer. For the target subjects with 10 labelled samples and 270/190 unlabelled samples, ssMEKT stands out itself with average accuracies of 82.50% and 81.94% on the dataset 1 and dataset 2, respectively, and sMEKT can also obtain higher classification performance (79.02%) than other spatial filtering-based transfer learning algorithms on dataset 1. Therefore, the experimental results show that our proposed algorithms can reduce the need of labelled target trials. In the future, we will not only choose the most beneficial source subjects for the target subjects, but also make use of the unlabelled target samples in the classification module. In addition, our proposed ssMEKT is designed offline since the unlabelled samples from the target subject are obtained a priori, instead of on-the-fly. Future work will be dedicated to updating ssMEKT in real-time BCI applications, where domain adaptation is performed between the labelled source domain and the increasing target domain.

Data Availability

Conflicts of Interest

Acknowledgments

References

[1] G. Pfurtscheller, C. Brunner, A. Schlögl, and F. H. Lopes da Silva, "Mu rhythm (de) synchronization and EEG single-trial classification of different motor imagery tasks," *NeuroImage*, vol. 31, no. 1, pp. 153–159, 2006.

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

[3] B. Blankertz, G. Dornhege, M. Krauledat, K.-R. Müller, and G. Curio, "The non-invasive Berlin Brain–Computer Interface: fast acquisition of effective performance in untrained subjects," *NeuroImage*, vol. 37, no. 2, pp. 539–550, 2007.

[4] G. Pfurtscheller and C. Neuper, "Motor imagery and direct brain computer communication," *Proceedings of the IEEE*, vol. 89, no. 7, pp. 1123–1134, 2001.

[5] Y. Xu, J. Hua, H. Zhang et al., "Improved transductive support vector machine for a small labelled set in motor imagery-based brain-computer interface," *Computational Intelligence and Neuroscience*, vol. 2019, pp. 1–16, Article ID 2087132, 2019.

[6] B. Blankertz, S. Lemm, M. Treder, S. Haufe, and K.-R. Müller, "Single-trial analysis and classification of ERP components–a tutorial," *NeuroImage*, vol. 56, no. 2, pp. 814–825, 2011.

[7] E. Yin, T. Zeyl, R. Saab, T. Chau, D. Hu, and Z. Zhou, "A hybrid brain–computer interface based on the fusion of P300 and SSVEP scores," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 4, pp. 693–701, 2015.

[8] R. Chai, S. H. Ling, G. P. Hunter, Y. Tran, and H. T. Nguyen, "Brain–computer interface classifier for wheelchair commands using neural network with fuzzy particle swarm optimization," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 5, pp. 1614–1624, 2014.

[9] C. Muhl, B. Allison, A. Nijholt, and G. Chanel, "A survey of affective brain computer interfaces: principles, state-of-the-art, and challenges," *Brain Computer Interfaces*, vol. 1, no. 2, pp. 66–84, 2014.

[10] D. Wu, V. J. Lawhern, S. Gordon, B. J. Lance, and C. T. Lin, "Driver drowsiness estimation from EEG signals using online weighted adaptation regularization for regression (OwARR)," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 6, pp. 1522–1535, 2017.

[11] M. Sample, M. Aunos, S. Blain-Moraes et al., "Brain–computer interfaces and personhood: interdisciplinary deliberations on neural technology," *Journal of Neural Engineering*, vol. 16, no. 6, Article ID 063001, 2019.

[12] I. Martisius and R. Damasevicius, "Class-adaptive denoising for EEG data classification," in *Proceedings of the 11th Artificial Intelligence and Soft Computing (ICAISC)*, pp. 302–309, Zakopane, Poland, April 2012.

[13] K. Kumar, D. D. Shrimankar, and N. Singh, "Eratosthenes sieve based key-frame extraction technique for event summarization in videos," *Multimedia Tools and Applications*, vol. 77, no. 6, pp. 7383–7404, 2018.

[14] X. Huang, Y. Xu, J. Hua et al., "A review on signal processing approaches to reduce calibration time in EEG-based brain–computer interface," *Frontiers in Neuroscience*, vol. 1519 pages, Article ID 733546, 2021.

[15] K. Kumar, D. D. Shrimankar, and N. Singh, "Event BAGGING: a novel event summarization approach in multiview surveillance videos," in *IEEE 2017 International Conference on Innovations in Electronics, Signal Processing and Communication*, pp. 106–111, Shillong, India, April 2017.

[16] C. Huang, X. Huang, Y. Fang et al., "Sample imbalance disease classification model based on association rule feature selection," *Pattern Recognition Letters*, vol. 133, pp. 280–286, 2020.

[17] X. Huang, B. Zhong, Y. Cao, Y. Yi, and M. Gu, "Chest X-ray lung Chinese description generation based on semantic labels and hierarchical LSTM," in *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine*, pp. 1020–1023, Seoul, South Korea, December 2020.

[18] K. Kumar and D. D. Shrimankar, "F.-D. E. S.: F-DES: fast and deep event summarization," *IEEE Transactions on Multimedia*, vol. 20, no. 2, pp. 323–334, 2018.

[19] D. Birvinskas, V. Jusas, I. Martisius, and R. Damasevicius, "EEG dataset reduction and feature extraction using discrete cosine transform," in *Proceedings of the 2012 Sixth UKSim/AMSS European Symposium on Computer Modeling and Simulation*, pp. 199–204, Malta, Malta, November 2012.

[20] I. Martisius, D. Birvinskas, R. Damasevicius, and V. Jusas, "EEG dataset reduction and classification using wave atom transform," in *Proceedings of the 23rd International Conference on Artificial Neural Networks and Machine Learning (ICANN 2013)*, pp. 208–215, Sofia, Bulgaria, September 2013.

[21] B. M. Mahmmod, S. H. Abdulhussain, T. Suk, and A. Hussain, "Fast computation of Hahn polynomials for high order moments," *IEEE Access*, vol. 10, pp. 48719–48732, 2022.

[22] K. A. AL-Utaibi, S. H. Abdulhussain, B. M. Mahmmod, M. A. Naser, M. Alsabah, and S. M. Sait, "Reliable recurrence algorithm for high-order Krawtchouk polynomials," *Entropy*, vol. 23, no. 9, 24 pages, Article ID 1162, 2021.

[23] S. H. Abdulhussain and B. M. Mahmmod, "Fast and efficient recursive algorithm of Meixner polynomials," *Journal of Real-Time Image Processing*, vol. 18, no. 6, pp. 2225–2237, 2021.

[24] K. Kumar and D. D. Shrimankar, "Deep event learning boosT-up approach: delta," *Multimedia Tools and Applications*, vol. 77, no. 20, pp. 26635–26655, 2018.

[25] K. Kumar, "Evs-D. K.: EVS-DK: event video skimming using deep keyframe," *Journal of Visual Communication and Image Representation*, vol. 58, pp. 345–352, 2019.

[26] K. Kumar, "Text query based summarized event searching interface system using deep learning over cloud," *Multimedia Tools and Applications*, vol. 80, no. 7, pp. 11079–11094, 2021.

[27] S. Sharma and K. Kumar, "Asl-3Dcnn: American sign language recognition technique using 3-D convolutional neural networks," *Multimedia Tools and Applications*, vol. 80, no. 17, pp. 26319–26331, 2021.

[28] J. Chen, Z. Yu, Z. Gu, and Y. Li, "Deep temporal-spatial feature learning for motor imagery-based brain-computer interfaces," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 11, pp. 2356–2366, 2020.

[29] F. Lotte, "Signal processing approaches to minimize or suppress calibration time in oscillatory activity-based brain–computer interfaces," *Proceedings of the IEEE*, vol. 103, no. 6, pp. 871–890, 2015.

[30] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[31] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020.

[32] W. Zhang and D. Wu, "Manifold embedded knowledge transfer for brain-computer interfaces," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 5, pp. 1117–1127, 2020.

[33] H. Ramoser, J. Müller-Gerking, and G. Pfurtscheller, "Optimal spatial filtering of single trial EEG during imagined hand movement," *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 4, pp. 441–446, 2000.

[34] M. Grosse-Wentrup, C. Liefhold, K. Gramann, and M. Buss, "Beamforming in noninvasive brain computer interfaces," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 4, pp. 1209–1219, 2009.

[35] H. Lu, H. Eng, C. Guan, K. N. Plataniotis, and A. N. Venetsanopoulos, "Regularized common spatial pattern with aggregation for EEG classification in small-sample setting," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 12, pp. 2936–2946, 2010.

[36] M. Cheng, Z. Lu, and H. Wang, "Regularized common spatial patterns with subject-to-subject transfer of EEG signals," *Cognitive Neurodynamics*, vol. 11, no. 2, pp. 173–181, 2017.

[37] Y. Xu, Q. Wei, H. Zhang et al., "Transfer learning based on regularized common spatial patterns using cosine similarities of spatial filters for motor-imagery BCI," *Journal of Circuits, Systems, and Computers*, vol. 28, no. 07, Article ID 1950123, 2019.

[38] D. Wu, X. Jiang, and R. Peng, "Transfer learning for motor imagery based brain-computer interfaces: a tutorial," *Neural Networks*, vol. 153, pp. 235–253, 2022.

[39] A. M. Azab, H. Ahmadi, L. Mihaylova, and M. Arvaneh, "Dynamic time warping-based transfer learning for improving common spatial patterns in brain-computer interface," *Journal of Neural Engineering*, vol. 17, no. 1, p. 0, Article ID 16061.

[40] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Classification of covariance matrices using a Riemannian-based kernel for BCI applications," *Neurocomputing*, vol. 112, pp. 172–178, 2013.

[41] H. Qi, Y. Xue, L. Xu, Y. Cao, and X. Jiao, "A speedy calibration method using Riemannian geometry measurement and other-subject samples on a P300 speller," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 3, pp. 602–608, 2018.

[42] A. Singh, S. Lal, and H. W. Guesgen, "Small sample motor imagery classification using regularized Riemannian features," *IEEE Access*, vol. 7, pp. 46858–46869, 2019.

[43] Y. Xu, X. Huang, and Q. Lan, "Selective cross-subject transfer learning based on Riemannian tangent space for motor imagery brain–computer interface," *Frontiers in Neuroscience*, vol. 15, Article ID 779231, 14 pages, 2021.

[44] P. Zanini, M. Congedo, C. Jutten, S. Said, and Y. Berthoumieu, "Transfer learning: a Riemannian geometry framework with applications to brain-computer interfaces," *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 5, pp. 1107–1116, 2018.

[45] P. L. C. Rodrigues, C. Jutten, and M. Congedo, "Riemannian procrustes analysis: transfer learning for brain–computer interfaces," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 8, pp. 2390–2401, 2019.

[46] N. R. Waytowich, V. J. Lawhern, A. W. Bohannon, K. R. Ball, and B. J. Lance, "Spectral transfer learning using information geometry for a user-independent brain-computer interface," *Frontiers in Neuroscience*, vol. 10, p. 430, 2016.

[47] L. Xu, M. Xu, Z. Ma, K. Wang, T. P. Jung, and D. Ming, "Enhancing transfer performance across datasets for brain-computer interfaces using a combination of alignment strategies and adaptive batch normalization," *Journal of Neural Engineering*, vol. 18, no. 4, Article ID 0460e5, 2021.

[48] Y. Jiang, D. Wu, Z. Deng et al., "Seizure classification from EEG signals using transfer learning, semi-supervised learning and TSK fuzzy system," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 12, pp. 2270–2284, 2017.

[49] D. Wu, "Online and offline domain adaptation for reducing BCI calibration effort," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 4, pp. 550–563, 2017.

[50] Graz University, *BCI Competition III Datasets Iva*, Graz University, Graz, Austria, 2004, http://www.bbci.de/competition/iii/datasetIva.

[51] M. Tangermann, K. R. Müller, A. Aertsen et al., "Review of the BCI competition IV," *Frontiers in Neuroscience*, vol. 6, p. 55, 2012.

[52] R. Peck and J. Van Ness, "The use of shrinkage estimators in linear discriminant analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 4, no. 5, pp. 530–537, 1982.

[53] M. Moakher, "A differential geometric approach to the geometric mean of symmetric positive-definite matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 26, no. 3, pp. 735–747, 2005.

[54] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on riemannian manifolds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1713–1727, 2008.

[55] X. Pennec, P. Fillard, and N. Ayache, "A Riemannian framework for tensor computing," *International Journal of Computer Vision*, vol. 66, no. 1, pp. 41–66, 2006.

[56] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

[57] K.-R. Müller, C. W. Anderson, and G. E. Birch, "Linear and nonlinear methods for brain-computer interfaces," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 11, no. 2, pp. 165–169, 2003.

[58] J. Perelmouter and N. Birbaumer, "A binary spelling interface with random errors," *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 2, pp. 227–232, 2000.

[59] Y. Fang, P. Chen, and T. han, "Hint: harnessing the wisdom of crowds for handling multi-phase tasks," *Neural Computing & Applications*, pp. 1–23, 2022, https://doi.org/10.1007/s00521-021-06825-7.