Pattern Recognition 110 (2021) 107626

Contents lists available at ScienceDirect

Pattern Recognition

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

A prototype-based SPD matrix network for domain adaptation EEG emotion recognition

Yixin Wang[a,b], Shuang Qiu[a], Xuelin Ma[a,b], Huiguang He[a,b,c,*]

[a] Research Center for Brain-inspired Intelligence and National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Science, Beijing, China
[b] University of Chinese Academy of Sciences, Beijing, China
[c] Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Science, Beijing, China

ARTICLE INFO

Article history:
Received 3 December 2019
Revised 5 August 2020
Accepted 29 August 2020
Available online 30 August 2020

Keywords:
EEG
Emotion recognition
Domain adaptation
SPD matrix
Riemannian manifold
Prototype learning

ABSTRACT

Emotion plays a vital role in human daily life, and EEG signals are widely used in emotion recognition. Due to individual variability, training a generic emotion recognition model across different subjects is difficult. The conventional method involves the collection of a large amount of calibration data to build subject-specific models. Recently, developing an effective brain-computer interface with a short calibration time has become a challenge. To solve this problem, we propose a domain adaptation SPD matrix network (daSPDnet) that can successfully capture an intrinsic emotional representation shared between different subjects. Our method jointly exploits feature adaptation with distribution confusion and sample adaptation with centroid alignment. We compute the SPD matrix based on the covariance as a feature and make a novel attempt to combine prototype learning with the Riemannian metric. Extensive experiments are conducted on the DREAMER and DEAP datasets, and the results show the superiority of our proposed method.

1. Introduction

With the continuous increase in human and machine communication requirements, emotion recognition has attracted more and more interest. Emotion recognition [1] is intended to detect and model human emotions during human-computer interfacing. Many signals can be used to indicate emotional states, including facial expressions [2], vocal expressions [3] and some physiological signals [4]. Among these signals, electroencephalogram (EEG) signals are an effective tool for reflecting emotional states with high temporal resolution [5].

Recently, emotion recognition based on EEG signals has been a topic of great research. The conventional method for EEG-based emotion recognition is to collect a substantial amount of calibration data from a new subject and then train a subject-specific model [6,7]. However, calibration data collection is a time-consuming task that seriously hinders the application of brain-computer interfaces (BCIs) in practice [8]. Developing an effective BCI after a short calibration has become an important trend.

Some researchers have attempted to explore the data from other existing subjects to reduce the requirement for calibration data from a new subject. Since EEG signals have individual differences, the distributions of data from different subjects greatly vary. Therefore, directly using the data from a certain subject to expand the training set of a new subject leads to poor performance for this new subject. To address the above problem, one promising method is domain adaptation (DA) [9], which can minimize the distribution discrepancy between the source domain (source subject) and the target domain (target subject) and thus improve the performance for the target subject. In this study, we design a novel DA method, which is aimed at reducing the time cost for calibration data collection and performing well on subsequent unlabelled target data.

In previous studies, many traditional DA methods were applied in the BCI field. Zheng [10] used four algorithms, transductive component analysis (TCA) [11], kernel principal component analysis (KPCA) [12], transductive support vector machine (TSVM) [13], and transductive parameter transfer (TPT) [14], on the EEG-based emotional SEED dataset [15] to construct a general model for a new target subject. Additionally, Chai [16] proposed adaptive subspace feature matching (ASFM), in which a linear transformation function was developed that matched the marginal distributions of the source and target subspaces. These methods are relatively shallow,

* Corresponding author at: Research Center for Brain-inspired Intelligence, Zhong Guan Cun East Road 95#, Beijing 100190, China.
E-mail addresses: wangyixin2016@ia.ac.cn (Y. Wang), shuang.qiu@ia.ac.cn (S. Qiu), maxuelin2015@ia.ac.cn (X. Ma), huiguang.he@ia.ac.cn (H. He).

https://doi.org/10.1016/j.patcog.2020.107626
0031-3203/

and few studies have taken advantage of deep networks to solve the DA problem in the BCI field.

Recently, a series of methods based on Riemannian geometry has been dedicated to solving transfer learning problems [17–20]. In these works, the symmetric and positive definite (SPD) matrix was computed as a feature, and the geometric properties of the SPD matrix on the Riemannian manifold was used to solve the DA problems. The minimum distance to mean (MDM) Riemannian classifier combined with the SPD matrix has already achieved considerable improvements in the BCI field [21]. Recently, Zanini [18] applied a congruence invariance transformation to recentre all the data points on the identity matrices. Yair [19] proposed a parallel transport on the SPD manifold, which was a mathematical recasting of the affine transformation proposed in [18]. Rodrigues [20] not only recentred matrices but also equalized the dispersions on each domain and rotated them around the geometric mean. The Riemannian metric for the SPD matrix has proven to be a valid measurement for measuring distances between samples in the BCI field [17].

Inspired by these Riemannian-based methods, we propose a domain adaptation SPD matrix network (daSPDnet). Our daSPDnet can extract an intrinsic emotional representation shared between different subjects while maintaining the SPD structure. We combine prototype learning with the Riemannian metric and design a new prototype loss, which aims to calculate the geometric mean of the SPD matrix set in the low-dimensional representation layer. The geometric means can be regarded as prototypes, which can be used not only to classify different emotional states but also to align the semantic information from different subjects at the same time. We attempt to imitate the geometry-aware DA algorithms in our deep learning framework and consider the knowledge transfer at both the feature level and sample level. At the feature level, our method can reduce the marginal distribution divergence between two domains in the high-layer representation space, which can be seen as a more effective operation to replace affine transformation. At the sample level, we match the conditional distribution for each category with the help of prototypes to achieve recentring operations similar to geometry-aware algorithms. All these components are trained jointly and in an end-to-end manner. In summary, this paper mainly contributes three innovations:

- Our work is the first attempt to use the SPD matrix as a feature in EEG emotion recognition. Compared with Riemannian-based methods, our model takes full advantage of deep learning to solve the DA problem of the SPD matrix.
- Our work novelly combines prototype learning with the Riemannian metric. Using the prototype loss we propose, it is easy to calculate the geometric mean in the low-dimensional layer of the neural network.
- Our work transfers knowledge from two levels to match two probability distributions. At the feature level, we confuse the marginal distributions from the source and target domains. At the sample level, we adapt the conditional distributions with alignment of prototypes in each category.

The performance of our model is validated on two EEG-based emotion recognition datasets: DREAMER [22] and DEAP [23]. The results show that our method can effectively reduce the calibration time and provide a significant improvement in the EEG emotion DA problem.

2. Preliminaries

To clarify the problem we will solve, we first illustrate the notations in this section. We also provide relevant knowledge about Riemannian geometry for SPD matrices and introduce the concepts of some metrics and the geometric mean.

2.1. Problem formulation

There are two domains, the *source* ($\mathcal{S}$) and the *target* ($\mathcal{T}$), which can be expressed as follows:

$$\mathcal{S} = \{(\mathbf{C}_i^S, y_i^S)|i \in \{1, \ldots, n^S\}\},$$
$$\mathcal{T} = \{(\mathbf{C}_i^T, y_i^T)|i \in \{1, \ldots, n^T\}\} \quad (1)$$

where $\mathbf{C}_i^S, \mathbf{C}_i^T \in \mathbb{R}^{N\times N}$ are EEG samples with covariance matrices regarded as features, and N indicates the number of electrodes. $y_i^S, y_i^T \in \{1, \ldots, K\}$ are their labels, and K is the number of emotional state classes. n^S and n^T are the numbers of divided sessions for the *source* and *target* subjects, respectively.

In our paper, $\mathcal{S}$ is a set of recordings acquired from a single subject, and $\mathcal{T}$ is a set from another subject. $\mathcal{T}_l$ is composed of labelled target samples from some calibration sessions, and $\mathcal{T}_u$ is a set of unlabelled target samples to be classified in the following sessions, where $\mathcal{T} = \mathcal{T}_l \cup \mathcal{T}_u$ and $\mathcal{T}_l \cap \mathcal{T}_u = \oslash$.

We assume that the label information of all of $\mathcal{S}$ and a part of $\mathcal{T}$ (i.e., $\mathcal{T}_l$) are known. Our goal is to train a classifier that uses the given knowledge of $\mathcal{S}$ and $\mathcal{T}_l$ and to obtain good performance on the samples from $\mathcal{T}_u$. This problem is a good application of the supervised transfer learning paradigm in the EEG emotion DA literature.

2.2. The SPD manifold and its metric

The set of SPD matrices is defined as

$$\mathcal{P}(n) = \{\mathbf{P} \in \mathbb{R}^{n\times n}|\mathbf{P}^T = \mathbf{P}, \mathbf{x}^T\mathbf{P}\mathbf{x} > 0, \forall \mathbf{x} \in \mathbb{R}^n\} \quad (2)$$

and $n \times n$ symmetric matrices form a differentiable Riemannian manifold. In this case, we define some foundational geometric concepts, such as geodesics and the geometric mean.

The distance between two points $\mathbf{P}_1, \mathbf{P}_2 \in \mathcal{P}(n)$ cannot be accurately measured by the Euclidean distance since it does not consider the inner curvature of the manifold. Here, we introduce the *geodesic distances* [24] to describe the length of the unique shortest path between two points on the manifold. Two popular metrics are widely used for the geodesic distances of the SPD matrices: the affine-invariant Riemannian metric (AIRM) [25] and the log-Euclidean metric (LEM) [26,27]. The AIRM is defined as

$$d_a(\mathbf{P}_1, \mathbf{P}_2) = \|log(\mathbf{P}_1^{-\frac{1}{2}}\mathbf{P}_2\mathbf{P}_1^{-\frac{1}{2}})\|_F \quad (3)$$

This metric is often used in some geometry-aware algorithms to solve the DA problem [18–20,28]. However, the inverse operation in the AIRM will lead to a high computational expense; therefore, we have chosen the more convenient LEM as the metric in our paper.

The LEM is simply defined as the Frobenius distance of the logarithms

$$d_l(\mathbf{P}_1, \mathbf{P}_2) = \|log(\mathbf{P}_1) - log(\mathbf{P}_2)\|_F \quad (4)$$

The LEM can reduce the manifold to a vectorial space, in which the classical Euclidean distance can be directly applied. In our paper, we take advantage of this metric to construct the deep learning framework for the SPD matrix and use it to calculate the geometric mean.

2.3. The geometric mean (the centre of mass)

We give the concept of the geometric mean [24] used to solve the classification problem. Let $\{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_N\}$ be a set of SPD matrices,

$$\mathcal{G}(\{\mathbf{P}\}_{i=1}^N) = \arg\min_{\mathbf{P}\in\mathcal{P}(n)} \sum_{i=1}^N \delta^2(\mathbf{P}_i, \mathbf{P}) \quad (5)$$

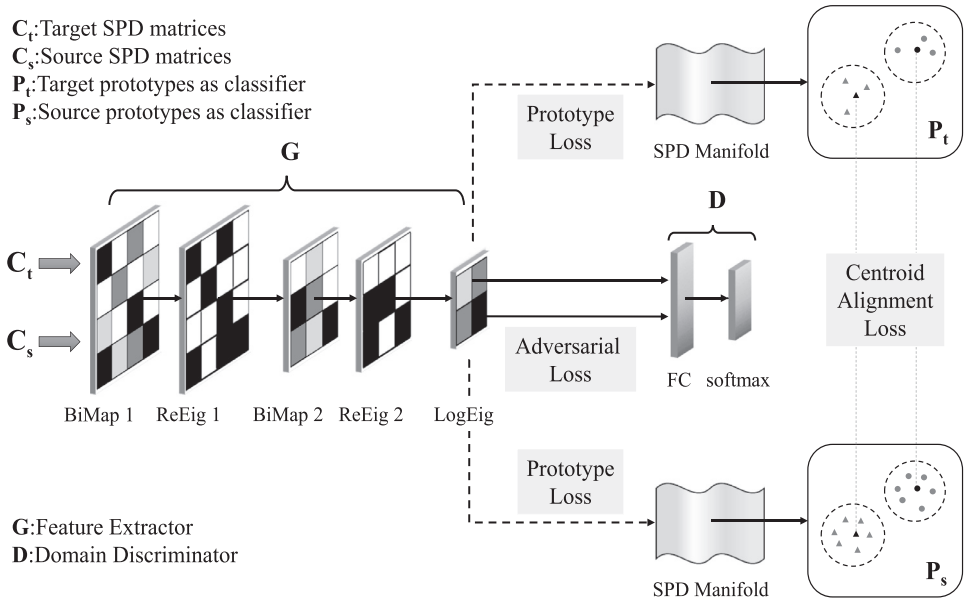

Fig. 1. Architecture of daSPDnet. During forward propagation, source data $\mathbf{C_s}$ and target data $\mathbf{C_t}$ are fed into the weight-shared feature extractor **G** to obtain the low-dimensional representation, from which the source prototypes $\mathbf{P_s}$ and the target prototypes $\mathbf{P_t}$ are calculated as classifiers. Both source and target representations are used to reduce the marginal distribution discrepancy via adversarial training implemented by the gradient reverse and domain discriminator **D**. In addition, the conditional distributions from different domains are matched via prototype alignment.

where $\delta^2(\mathbf{P}_i, \mathbf{P})$ is the Riemannian distance between $\mathbf{P}_i$ and $\mathbf{P}$. $\mathcal{G}(\{\mathbf{P}\}_{i=1}^N)$ can appropriately describe the central tendency (expected value) for the variance [29]. One useful classifier for the SPD matrix based on this property is MDM [30]. The training process of this classifier involves calculating the geometric means $\widehat{\mathbf{P}}(k)$ for each class $k = \{1, \ldots, K\}$ in the training set. Then, a new testing sample $\mathbf{P}_i$ belongs to the class with the closest geometric mean, i.e.,

$$\widehat{k} = \arg\min_{k\in\{1,\ldots,K\}} \delta^2(\mathbf{P}_i, \widehat{\mathbf{P}}(k)) \tag{6}$$

Many studies have attempted to find solutions to the value of the geometric mean [31,32]. In our paper, we novelly combine prototype learning [33–35] with Riemannian geometry and try to use mini-batch stochastic gradient descent (SGD) to solve the geometric mean in the deep learning network.

3. Methods

In this section, we first describe the process of calculating the covariance matrix as the input of our proposed network, and then, we use the SPD matrix network as our feature extractor. We also introduce three losses: prototype loss for classification, adversarial loss and centroid alignment loss for DA. The structure of our method is shown in Fig. 1.

3.1. Covariance matrix

Raw EEG signals are captured by a BCI headset with multiple electrodes and recorded as a multivariate time series. After preprocessing, data segments $\mathbf{R} \in \mathbb{R}^{N\times T}$ are acquired via the interception of these signals through a sliding window, where N indicates the number of electrodes and T indicates the number of points in the time window. Then, we calculate the spatial covariance matrices $\mathbf{C} \in \mathbb{R}^{N\times N}$ as features, which are defined as

$$\mathbf{C} = \frac{1}{T-1}\mathbf{R}\mathbf{R}^T \tag{7}$$

These types of features can simply be extracted from the preprocessed signals, which highlight the connectivity information of EEG signals between different channels. Since these matrices are computed from data by averaging over time, they are also robust to noise [19].

3.2. SPD Matrix network (SPDnet)

The SPD matrix network (SPDnet) [36] is analogous to the widely used convolutional network (ConvNet), which can retain the intrinsic geometry information of the SPD matrix. The network consists of three types of layers. The **bilinear mapping (BiMap) layer** is designed to transform the input SPD matrix into a new SPD matrix with a bilinear mapping. It is a fully connected convolution-like layer. The **eigenvalue rectification (ReEig) layer** is a rectified linear unit (ReLU)-like layer. Similar to the ReLU layer in a traditional neural network, the ReEig layer can rectify the resulting SPD matrix through a non-linear function. As described in the previous section, the **log eigenvalue (LogEig) layer** endows elements in Riemannian manifolds with a Lie group structure so that the matrix can be flattened [25]. Let $\mathbf{X}_{k-1}$ be the input SPD matrix and $\mathbf{X}_k$ be the output. The k-th BiMap layer f_b^k, the k-th ReEig layer f_r^k, and the LogEig layer f_l^k can be defined, respectively, as follows:

$$\mathbf{X}_k = f_b^k(\mathbf{X}_{k-1}; \mathbf{W}_k) = \mathbf{W}_k\mathbf{X}_{k-1}\mathbf{W}_k^T \tag{8}$$

$$\mathbf{X}_k = f_r^k(\mathbf{X}_{k-1}) = \mathbf{U}_{k-1}\max(\epsilon\mathbf{I}, \mathbf{\Sigma}_{k-1})\mathbf{U}_{k-1}^T \tag{9}$$

$$\mathbf{X}_k = f_l^k(\mathbf{X}_{k-1}) = \log(\mathbf{X}_{k-1}) = \mathbf{U}_{k-1}\log(\mathbf{\Sigma}_{k-1})\mathbf{U}_{k-1}^T \tag{10}$$

where $\mathbf{U}_{k-1}$ and $\mathbf{\Sigma}_{k-1}$ are provided by eigenvalue decomposition (EIG) and ϵ is a rectification threshold. The max operation is performed element-wise. Moreover, $\log(\mathbf{\Sigma}_{k-1})$ is the diagonal matrix of eigenvalue logarithms.

3.3. Prototype loss

The prototype learning network [33] learns a low-dimensional representation through an embedding function $f_\varphi(\cdot)$ with learnable parameters φ. The prototypes can be regarded as the geomet-

ric means of each class in the embedding space. We use the Riemannian distance $\delta^2(\cdot,\cdot)$ to measure the similarity between the samples and the prototypes. From the perspective of probability, a sample $(\mathbf{C}, y)$ that is close to the prototype $\widehat{\mathbf{P}}(k)$ is represented as

$$p(\mathbf{C} \in \widehat{\mathbf{P}}(k)|\mathbf{C}) \propto -\delta^2(f_\varphi(\mathbf{C}), \widehat{\mathbf{P}}(k)) \tag{11}$$

Based on softmax, the probability of $p(\mathbf{C} \in \widehat{\mathbf{P}}(k)|\mathbf{C})$ can be further defined as

$$p(\mathbf{C} \in \widehat{\mathbf{P}}(k)|\mathbf{C}) = \frac{e^{-\gamma\delta^2(f_\varphi(\mathbf{C}),\widehat{\mathbf{P}}(k))}}{\sum_{i=1}^{K} e^{-\gamma\delta^2(f_\varphi(\mathbf{C}),\widehat{\mathbf{P}}(i))}} \tag{12}$$

where we choose the LEM (seen in (4)) to compute $\delta^2(\cdot,\cdot)$, and γ is a hyper-parameter that controls the hardness of the probability assignment. Here, we set $\gamma = 1$. In addition, the cross entropy loss under our setting is

$$\mathcal{L}_{ce}((\mathbf{C}, y); \varphi) = -\log\left(\frac{e^{-\delta^2(f_\varphi(\mathbf{C}),\widehat{\mathbf{P}}(k))}}{\sum_{i=1}^{K} e^{-\delta^2(f_\varphi(\mathbf{C}),\widehat{\mathbf{P}}(i))}}\right) \tag{13}$$

From Eqs. (11)–(13), we can see that minimizing the loss function means reducing the distance between the samples with the prototypes. We can also see that learning the prototypes of each class is equivalent to learning the geometric means of the classes, as defined in Eq. (5).

3.4. DA Loss

In this paper, we resort to both adversarial loss and alignment loss based on prototype learning to achieve DA.

Adversarial loss has been widely used in DA [37], which is usually based on a generative adversarial network (GAN) [38]. The concrete operation is to add one discriminator to the embedding space to determine whether the sample comes from the source or target domain, while the feature extractor is learned to fool the discriminator. The formal expression is

$$\begin{aligned}\mathcal{L}_{adv}(\mathcal{S}, \mathcal{T}_l) = &\mathbb{E}_{\mathbf{C}\sim D_\mathcal{S}}[\log(1 - D \circ G(\mathbf{C}))] \\ &+ \mathbb{E}_{\mathbf{C}\sim D_{\mathcal{T}_l}}[\log(D \circ G(\mathbf{C}))]\end{aligned} \tag{14}$$

where D is the domain classifier, and G is the feature extractor. When the features from the feature extractor become domain invariant, this process reaches an equilibrium.

Centroid alignment loss is an intuitive idea of semantic information transfer. We directly close the distance between prototypes of the same category from different domains [39]. This loss can match the conditional distribution for each category. We still measure the distance by the Riemannian metric.

$$\mathcal{L}_{ca}(\mathcal{S}, \mathcal{T}_l) = \Sigma_{k=1}^{K} \delta^2(\widehat{\mathbf{P}}_\mathcal{S}(k), \widehat{\mathbf{P}}_{\mathcal{T}_l}(k)) \tag{15}$$

where K is the number of emotion state classes, $\widehat{\mathbf{P}}_\mathcal{S}(k)$ and $\widehat{\mathbf{P}}_{\mathcal{T}_l}(k)$ are prototypes for class k in feature space.

3.5. Optimization

Training procedure In our framework, the *source* $(\mathcal{S})$ set and the *target* $(\mathcal{T})$ set share the basic feature extractor; they are used to learn the prototypes for each domain. Combined with the above loss functions, the final objective function is defined as

$$\begin{aligned}\min_{\varphi,\widehat{\mathbf{P}}_\mathcal{S},\widehat{\mathbf{P}}_{\mathcal{T}_l}} \ &\mathcal{L} = \mathcal{L}_{ce}(\mathcal{S} \cup \mathcal{T}_l) \\ &+ \lambda_1\mathcal{L}_{adv}(\mathcal{S}, \mathcal{T}_l) + \lambda_2\mathcal{L}_{ac}(\mathcal{S}, \mathcal{T}_l)\end{aligned} \tag{16}$$

where λ_1 and λ_2 are the hyper-parameters that aim to balance these loss function terms. Note that the parameters of the feature extractor (i.e., φ in $f_\varphi(\cdot)$) will take the reverse direction of the gradients during back-propagation so that our model minimizes the prototype learning loss while maximizing the domain discriminator loss. To show the optimization details, Algorithm 1 outlines the entire learning algorithm of daSPDnet.

Algorithm 1 Learning algorithm for our daSPDnet.

Require: source dataset $\mathcal{S} = \left\{(\mathbf{C}_i^S, y_i^S)\right\}_{i=1}^N$,
target dataset $\mathcal{T}_l = \left\{(\mathbf{C}_i^T, y_i^T)\right\}_{i=1}^M$,
source domain label set $\mathcal{D}_\mathcal{S} = \{0\}$,
target domain label set $\mathcal{D}_\mathcal{T} = \{1\}$
Ensure: model parameters φ,
source prototypes $\widehat{\mathbf{P}}_\mathcal{S}$, target prototypes $\widehat{\mathbf{P}}_\mathcal{T}$
Initialization: $\widehat{\mathbf{P}}_\mathcal{S} = 0$, $\widehat{\mathbf{P}}_\mathcal{T} = 0$
model parameters φ, learning rate α
Use mini-batch N update φ by the Adam optimizer
repeat
Update the parameters of SPDnet:
$\varphi_{SPD} \leftarrow \varphi_{SPD} - \alpha(\frac{\partial\mathcal{L}_{ce}(\mathcal{S})}{\partial\varphi_{SPD}} + \frac{\partial\mathcal{L}_{ce}(\mathcal{T}_l)}{\partial\varphi_{SPD}})$;
Update the prototypes of the source and target domains:
$\widehat{\mathbf{P}}_\mathcal{S} \leftarrow \widehat{\mathbf{P}}_\mathcal{S} - \alpha(\frac{\partial\mathcal{L}_{ce}(\mathcal{S})}{\partial\widehat{\mathbf{P}}_\mathcal{S}} + \frac{\partial\mathcal{L}_{ca}(\mathcal{S},\mathcal{T}_l)}{\partial\widehat{\mathbf{P}}_\mathcal{S}})$,
$\widehat{\mathbf{P}}_\mathcal{T} \leftarrow \widehat{\mathbf{P}}_\mathcal{T} - \alpha(\frac{\partial\mathcal{L}_{ce}(\mathcal{T}_l)}{\partial\widehat{\mathbf{P}}_\mathcal{T}} + \frac{\partial\mathcal{L}_{ca}(\mathcal{S},\mathcal{T}_l)}{\partial\widehat{\mathbf{P}}_\mathcal{T}})$;
Update the parameters of the domain classifier:
$\varphi_{domain} \leftarrow \varphi_{domain} + \alpha\frac{\partial\mathcal{L}_{adv}(\mathcal{S},\mathcal{T}_l)}{\partial\varphi_{domain}}$;
If the model has been trained for m iterations, then $\alpha \leftarrow 0.1 \times \alpha$.
until convergence

$$\begin{aligned}\widehat{k} &= \arg\min_{k\in\{1,\ldots,K\}} g_k(\mathbf{C}) \\ g_k(\mathbf{C}) &= \min_{j\in\{1,2\}} \delta^2(f_\varphi(\mathbf{C}), \widehat{\mathbf{P}}_j(k))\end{aligned} \tag{17}$$

Inference Given a test sample $\mathbf{C}$, we obtain the lower representation on a new SPD manifold $f_\varphi(\mathbf{C})$. Then, we compute the distance between the new representation and the learned prototype $\widehat{\mathbf{P}}$, and the prediction procedures can be defined as follows:

$$\widehat{k} = \arg\min_{k\in\{1,\ldots,K\}} \delta^2(f_\varphi(\mathbf{C}), \widehat{\mathbf{P}}(k)), \tag{18}$$

where $\widehat{k}$ is the predicted category. When using the basic model for simple classification, we can directly obtain the results according to Eq. (18). Regarding the DA problem, we combine prototypes learned from the source domain and the target domain, i.e., $\widehat{\mathbf{P}}_j(k) = \widehat{\mathbf{P}}_\mathcal{S} \bigcup \widehat{\mathbf{P}}_{\mathcal{T}_l}$, where $j \in \{1, 2\}$ represents the domain index. The formula is improved as follows:

$$\begin{aligned}\widehat{k} &= \arg\min_{k\in\{1,\ldots,K\}} g_k(\mathbf{C}) \\ g_k(\mathbf{C}) &= \min_{j\in\{1,2\}} \delta^2(f_\varphi(\mathbf{C}), \widehat{\mathbf{P}}_j(k))\end{aligned} \tag{19}$$

3.6. Source selection strategy

The distributions of different domains greatly vary, and the sources that are more related to the target may lead to a better performance. In EEG emotion recognition fields, the number of sources is relatively large, and the diversity among subjects is also large. Selecting more appropriate sources before transfer is necessary.

To clearly clarify our strategy, we first simplify our method for subject-dependent emotion recognition, which is called prototype-based SPDnet (pSPDnet), as shown in Fig. 2. Let the training set be $\{(\mathbf{C}_i, y_i)\}_{i=1}^N$, and the objective function is as follows:

$$\min_{\varphi,\widehat{\mathbf{P}}} \mathcal{L} = \mathcal{L}_{ce}((\mathbf{C}_i, y_i)) \tag{20}$$

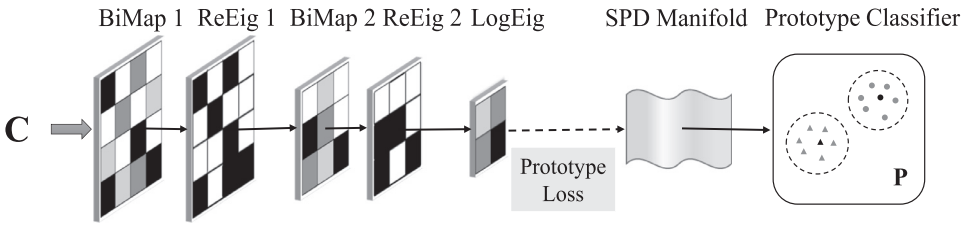

Fig. 2. Architecture of our basic prototype SPDnet (pSPDnet). pSPD is a subject-dependent method, and the samples **C** that come from one subject are fed into the network to learn the prototypes **P** as the classifier.

The first step of our source selection strategy takes advantage of the existing multi-source to train N basic pSPD models. When there comes a new subject as the target domain, we collect a small amount of labelled calibration data. We enumerate the N classifiers to classify the labelled target samples and rank these sources in the order of accuracy from high to low. Then, we locate the top K classifiers and regard these K training datasets as the appropriate sources. The selected source samples and the labelled target samples co-train K daSPD models, and the results of these K daSPD models are integrated to obtain the final prediction of the following test samples.

4. Experiments and results

4.1. Categorization of emotion

Because of the subjectivity and uncertainty of emotion, labelling the categories of emotion in emotion recognition is a special problem. The most common model used in the categorization of emotion is the circumplex model of affect, which maps emotions into the valence and arousal dimensions [40]. A third dimension, dominance [41], is usually used in combination with these two dimensions to form the valence arousal and dominance (VAD) emotional model. Valence goes from very positive feelings to very negative; arousal goes from states such as sleepy to excited; and finally, dominance corresponds to the strength of the emotion. These three assessments are independent and separately used to evaluate emotions in our paper.

4.2. Datasets

Here, we briefly introduce the experimental settings, data preprocessing and feature extraction for two benchmark emotion recognition datasets, the DREAMER dataset[1] and the DEAP dataset.[2]

DREAMER dataset [22]: Eighteen film clips were chosen to evoke emotions, and the length of the film clips ranged from 65 to 393 s. Twenty-three volunteers were asked to watch these films. EEG data were recorded by an Emotiv EPOC at a sampling rate of 128 Hz with 14 channels. After watching a film clip, self-assessment manikins (SAMs) were used to obtain each subject's assessments of arousal, valence and dominance. The evaluations of each index had a scale from 1 to 5.

The EEG data were recorded at a sampling rate of 128 Hz, without down-sampling. Then, the data were common average referenced. A band-pass frequency filter from 4 to 30 Hz was applied to remove most ocular artefacts and muscle movement-produced artefacts.

We regarded a score of 1–2 as 'low' and a score of 4–5 as 'high'. According to this criterion, there were three kinds of different binary classifications: low/high arousal, low/high valence and low/high dominance. To ensure that the target emotions were evoked, only data during the last 60 s of each trial were used for further analysis. The 60-s EEG signals were segmented by a sliding window of 2 s with an overlap of 1 s. Thus, we obtained 59 samples in one session. For each sample, the covariance matrix was calculated as a feature, the size of which was 14*14.

DEAP dataset [23]: The stimuli were 120 one-minute music videos. Thirty-two subjects were asked to watch 40 of these music videos. The EEG data were recorded at a sampling rate of 512 Hz using 32 electrodes. After each trial, valence, arousal and dominance were rated directly on a continuous 9-point scale by each subject.

The EEG data were common average referenced and down-sampled to 128 Hz. The electrooculography (EOG) signal was recorded in the experiment and later used to identify blink artefacts in the EEG data. To further filter the noise and remove the artefacts, the signals were then processed with a band-pass filter between 4 Hz and 45 Hz.

We regarded a score of 1–4 as 'low' and a score of 6–9 as 'high'. According to this criterion, we considered three kinds of different binary classifications: low/high arousal, low/high valence and low/high dominance. The data were segmented into 60-s trials, and a 3-s pretrial baseline was removed. We used a sliding window of 2 s with an overlap of 1 s. We obtained 59 samples in one session, and the feature of each sample was a 32*32 SPD matrix.

4.3. Source selection

To locate the number of sources we selected, we conducted large-scale experiments on the two datasets, as shown in Fig. 3. We used all the sessions from source subjects and 4 sessions from the target subjects (two high and two low) to train multiple models. The remaining sessions of the target subject were predicted based on the voting by these models.

As the source number grew from 1 to 7, the results fluctuated sharply for the three assessments on each dataset. When the number of sources continued to increase, the resulting curve gradually flattened. To balance the computational burden and the classification performance, we used 7 sources in the following experiments. In addition, we also applied our source selection strategy to the compared methods based on other common classifiers.

4.4. Compared methods and parameter settings

SVM is the most common method implemented for EEG emotion recognition, and three traditional methods based on the SVM classifier were selected as baseline methods for comparison experiments. Here, we used a linear SVM and let C = 1.0.

- **TCA** [11]: learns some transfer components across domains using the maximum mean discrepancy; here, we set the transfer component dimension equal to 50.

[1] Data are available at https://zenodo.org/record/546113#.X1GXWHkzaUk.
[2] Data are available at http://www.eecs.qmul.ac.uk/mmv/datasets/deap/.

Table 1
Mean accuracy in DA emotion recognition on the DREAMER dataset.

Method	Feature	Accuracy(%)		
		Valence	Arousal	Dominance
TCA [11]	PSD	59.31 ± 8.77	66.54 ± 10.79†	62.51 ± 8.67
KPCA [12]	PSD	62.52 ± 7.79	72.39 ± 12.22	72.05 ± 12.48
TSVM [13]	PSD	60.98 ± 9.90	67.85 ± 15.06	69.51 ± 17.54
TCA [11]	SPD vector	56.53 ± 10.23	58.35 ± 12.60	59.49 ± 9.85
KPCA [12]	SPD vector	61.10 ± 6.89	69.31 ± 10.47	65.20 ± 9.37
TSVM [13]	SPD vector	57.63 ± 10.56	65.61 ± 23.49	72.12 ± 23.07
RCT [18]	SPD matrix	62.98 ± 6.89	72.64 ± 12.22	71.50 ± 11.57
PRL [19]	SPD matrix	60.65 ± 7.02	59.83 ± 18.99	60.26 ± 23.82
RPA [20]	SPD matrix	61.31 ± 11.10	69.90 ± 13.14	67.29 ± 10.01
pSPDnet (calibration)	SPD matrix	50.47 ± 6.72***	57.18 ± 14.05***	59.52 ± 16.20***
pSPDnet (source)	SPD matrix	64.16 ± 9.06**	71.64 ± 16.22**	76.86 ± 15.56***
Ours-daSPDnet	SPD matrix	**67.99 ± 6.34**	**76.57 ± 14.04**	**81.77 ± 14.24**

*There are significant differences between our method and other non-transfer methods. (***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$). † There are significant differences between the SPD vector feature and the PSD feature in SVM-based methods. (†: $p < 0.05$)

Table 2
Mean accuracy in DA emotion recognition on the DEAP dataset.

Method	Feature	Accuracy(%)		
		Valence	Arousal	Dominance
TCA [11]	PSD	55.32 ± 5.16	55.05 ± 9.60	55.01 ± 10.01
KPCA [12]	PSD	54.89 ± 7.85	60.18 ± 10.34	57.81 ± 8.00
TSVM [13]	PSD	60.98 ± 9.90†	56.59 ± 11.98	56.81 ± 12.51
TCA [11]	SPD vector	53.15 ± 9.85	53.49 ± 8.32	53.77 ± 9.29
KPCA [12]	SPD vector	54.11 ± 10.03	56.68 ± 11.19	55.67 ± 11.05
TSVM [13]	SPD vector	62.53 ± 14.67	55.38 ± 24.91	58.42 ± 21.05
RCT [18]	SPD matrix	58.88 ± 5.77	55.82 ± 5.13	59.98 ± 9.67
PRL [19]	SPD matrix	38.57 ± 15.08	44.61 ± 23.99	51.73 ± 26.48
RPA [20]	SPD matrix	59.30 ± 6.24	56.53 ± 4.65	57.93 ± 6.09
pSPDnet (calibration)	SPD matrix	54.52 ± 11.86***	53.11 ± 13.52***	53.66 ± 20.13***
pSPDnet (source)	SPD matrix	62.53 ± 9.05**	67.19 ± 13.35*	63.30 ± 9.80*
Ours-daSPDnet	SPD matrix	**66.47 ± 8.75**	**69.79 ± 11.93**	**71.12 ± 14.68**

*There are significant differences between our method and other non-transfer methods. (***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$) † There are significant differences between the SPD vector feature and the PSD feature in SVM-based methods. (†: $p < 0.05$)

- **KPCA** [12]: projects the original high dimensional feature space into a low dimensional feature space using kernel methods; here, we used a linear kernel, and the dimension was equal to 50.
- **TSVM** [13]: learns the decision boundary in a semi-supervised manner and weights all the training instances equally. Here, we used the radial basis function kernel, with C = 1e−4 and gamma = 0.5.

Three geometry-aware methods based on the MDM classifier were also implemented for comparison experiments.

- **RCT** [18]: only considers the samples of each domain recentred about one reference identity matrix.
- **PRL** [19]: is analogous to **RCT**, but the reference point is halfway along the geodesic path linking the geometric means of each dataset.
- **RPA** [20]: is an improvement of **RCT**, which not only recentres the geometric means but also performs other geometric transformations.

For our basic pSPD-based model, we designed two comparison methods, **pSPD (calibration)** and **pSPD (source)**.

- **pSPD (calibration)**: directly uses the calibration sample training models without the help of the source domain.
- **pSPD (source)**: uses the data from the selected source domains to classify the unlabelled samples on the target domain.

The feature extractor consists of two BiMap layers, two ReEig layers and one LogEig layer. For DREAMER, we set the BiMap matrices to 14*10 and 10*5. The DEAP values were 32*20 and 20*10. For the daSPD model, we added a two-layer domain discriminator, and the numbers of nodes were 20 and 2. The epochs were set to 15 and 30, and the mini-batch size was 59 for each dataset. We used two hyper-parameters, λ_1 and λ_2. λ_1 is an adaptive parameter, which decreases as the iteration step increases. We conducted cross-validation on the training set to determine the hyper-parameters. We selected λ_2 from $10^{[-3:-1]}$. We used L2 regularization and selected the parameter in the range of $10^{[-3:-1]}$ with a step of one. The Adam optimizer with a learning rate of 0.1 was utilized to train all of the models. We applied exponential decay to the learning rate, which decayed every 180 steps with a base of 0.1.

4.5. Comparison of common methods

We first compared the result with some existing methods that have been used in EEG-based DA problems. For all the methods, source selection was first applied, and the training set included all sessions from the source subjects and four sessions from the target subjects (two high valence/arousal/dominance and two low valence/arousal/dominance). The remaining sessions of the target subjects were regarded as testing sets. We randomly selected three times from all of the target sessions to obtain the mean accuracy. The results are summarized in Tables 1 and 2. The best results are emphasized in boldface, We repeated these SVM-based methods using one widely used feature in EEG emotion recognition, the power spectral density (PSD). To apply the SPD matrix to

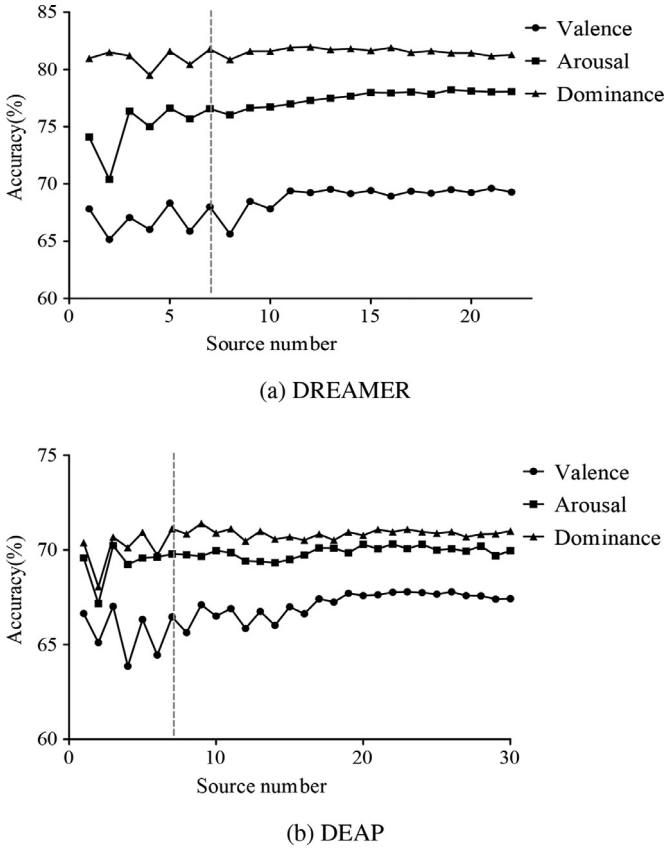

Fig. 3. Mean accuracy achieved with different source numbers on (a) the DREAMER dataset and (b) the DEAP dataset. We chose 7 as the number of sources, which is marked by the grey dashed line.

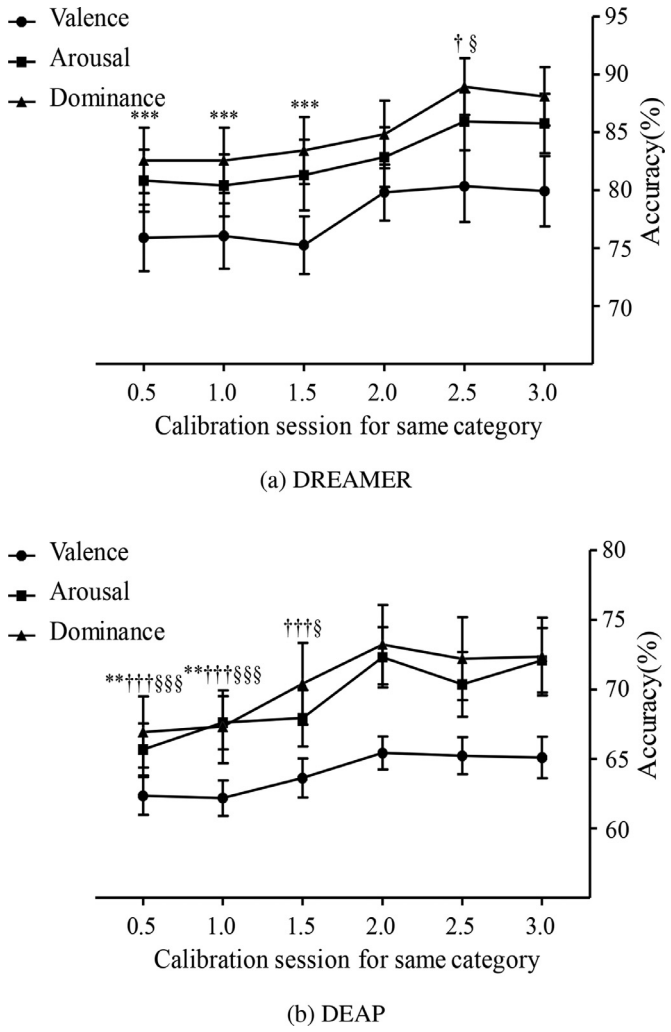

Fig. 4. Performance comparison for different calibration data quantities on (a) the DREAMER dataset and (b) the DEAP dataset. Here, we calculated the *t*-test between 2.0 and other conditions. We carried out the significance test between 2.0 and other sessions. (*:'valence'; †: 'arousal'; §: 'dominance').

SVM-based transfer methods directly, we extracted the upper triangular matrix of SPD and reshaped it into a vector as a feature for TCA, KPCA and TSVM, which is named the 'SPD vector'. On the two benchmark datasets, there were few significant differences between the PSD feature and SPD vector, only 'arousal, TCA' on the DREAMER dataset and 'valence, TSVM' on the DEAP dataset, with $p < 0.05$. This demonstrates that our SPD vector can achieve a similar performance on emotion recognition as the PSD feature.

On the DREAMER dataset, our method achieved a high accuracy of 67.99% for 'valence', 76.57% for 'arousal' and 81.77% for 'dominance', significantly outperforming the traditional transfer learning methods using the PSD or SPD vector (all: $p < 0.05$) and the geometry-aware methods using the SPD matrix (all: $p < 0.01$). On the DEAP dataset, our method also achieved the best classification performance, with accuracies of 66.47%, 69.79%, and 71.12%. Similarly, on the DEAP dataset, our method achieved a significant improvement compared with these two kinds of methods on emotion classification (all traditional methods: $p < 0.05$, all geometry-aware methods: $p < 0.001$).

Moreover, we compared our daSPDnet with its basic network pSPDnet using two non-transfer methods. For the pSPDnet (calibration) method, our daSPDnet achieved significantly higher values on the DREAMER dataset, with increases of 17.52%, 19.39% and 22.25% (all: $p < 0.001$). On the DEAP dataset, our daSPDnet achieved significantly higher classification accuracies, with increases of 11.95%, 16.68% and 17.46% (all: $p < 0.001$). For pSPDnet (source), our daSPDnet achieved 3.83% ($p < 0.01$), 4.93% ($p < 0.01$) and 4.91% ($p < 0.001$) higher values than the simple integration of pSPDnet for the three assessments on the DREAMER dataset. On the DEAP dataset, our daSPDnet also achieved 3.94% ($p < 0.01$), 2.60% ($p < 0.05$) and 7.82% ($p < 0.05$) higher values. The results show that our method can effectively outperform the simple integration from selected sources and the subject-dependent model using a small number of calibration sessions.

4.6. Comparison of calibration data quantities

To investigate the effect of the calibration data (labelled target samples) quantity, different numbers of calibration samples were selected to train our model. For the DREAMER dataset, the number of test sessions was 3, and for the DEAP dataset, it was 12. Here, 0.5, 1.0, 1.5, 2.0, 2.5, and 3.0 sessions for each category were selected as the target domain, and 0.5 indicates that the first half of the sessions were used for training.

Fig. 4 shows that the accuracy becomes high with increasing number of calibration sessions. A one-way analysis of variance (ANOVA) shows that the number of calibration sessions has a significant influence on the classification performance of daSPDnet on the DREAMER dataset (for 'valence', $\chi^2(5) = 15.462$, $p < 0.01$; for 'arousal', $\chi^2(5) = 24.563$, $p < 0.001$; and for 'dominance', $\chi^2(5) = 13.875$, $p < 0.05$) and the DEAP dataset (for 'valence', $\chi^2(5) = 12.047$, $p < 0.05$; for 'arousal', $\chi^2(5) = 30.385$, $p < 0.001$; and for 'dominance', $\chi^2(5) = 29.108$, $p < 0.001$).

Table 3
Mean accuracy for the ablation study on the DREAMER dataset.

Method	Accuracy(%)		
	Valence	Arousal	Dominance
Only adversarial loss	65.16 ± 5.01***	74.12 ± 14.96***	78.93 ± 14.70***
Only centroid alignment loss	65.25 ± 4.93***	74.83 ± 14.87***	79.02 ± 15.98***
Ours-daSPDnet	**67.99 ± 6.34**	**76.57 ± 14.04**	**81.77 ± 14.24**

*There are significant differences between our method and ablated methods. (***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$)

Table 4
Mean accuracy for the ablation study on the DEAP dataset.

Method	Accuracy(%)		
	Valence	Arousal	Dominance
Only adversarial loss	65.84 ± 8.43	69.60 ± 12.13	70.93 ± 15.22
Only centroid alignment loss	64.80 ± 7.88**	68.62 ± 11.92**	70.81 ± 15.27
Ours-daSPDnet	**66.47 ± 8.75**	**69.79 ± 11.93**	**71.11 ± 14.68**

*There are significant differences between our method and ablated methods. (***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$)

On the DREAMER dataset, from 0.5 to 2.0 sessions for each category, the results gradually increased ($p < 0.05$) and stabilized (not significant) from 2.0 for 'valence'. The *t*-test analysis showed the significance of these classification results for 'valence': 2.0 vs. 0.5, $p < 0.05$; 2.0 vs. 1.0, $p < 0.05$; and 2.0 vs. 1.5, $p < 0.05$. On DEAP for the three assessments, the classification results of 2.0 sessions for each category were significantly higher than those with fewer sessions. (For all: 2.0 vs. 0.5, $p < 0.01$; 2.0 vs. 1.0, $p < 0.01$; and 2.0 vs. 1.5, $p < 0.05$.) In addition, no significant difference was observed between using 2.0 sessions and more sessions. To balance the calibration time and the average accuracy, we chose 4 sessions (2.0 sessions for each category) as the calibration sessions.

4.7. Ablation study

We conducted ablation studies to evaluate the effects of our two main components in our framework on the classification performance. For each experiment, we only maintain one loss to solve the DA problem.

Tables 3 and 4 show the results on the two datasets. The best results are emphasized in boldface. The adversarial loss can significantly improve the classification performance for the three assessments on the DREAMER dataset. (all: $p < 0.001$) On the DEAP dataset, this loss also showed significant improvement for 'valence' ($p < 0.01$) and 'arousal' ($p < 0.01$). The centroid alignment loss had a significant impact on the classification performance on the DREAMER dataset for the three assessments (all: $p < 0.001$), and the loss only increased the accuracy, but not significantly, for all three assessments on the DEAP dataset. Above all, our method combining the two DA constraints obtained the best accuracy on both datasets.

4.8. Comparison of EEG features

The SPD matrix based on the covariance was calculated as a feature for our proposed method. The correlation and coherence are also in the form of SPD matrices, which were also used in our model. The correlation is similar to the covariance, which is normalized to the range of (0,1). The coherence has generally been assessed based on the similarity of the frequency content across EEG sensors.

First, the SPD matrix based on the covariance shows better performance than any other SPD features on these two datasets, as shown in Tables 5 and 6. The best results are emphasized in boldface. Our covariance feature achieved significantly higher classification than the coherence of certain bands on the DREAMER dataset for 'arousal' (θ: $p < 0.05$, α: $p < 0.01$, β: $p < 0.01$) and 'dominance' (θ: $p < 0.01$, α: $p < 0.01$, β: $p < 0.01$). On the DEAP dataset, our covariance showed significant superiority over the coherence of certain bands on the DEAP dataset for 'valence' (θ: $p < 0.001$, α: $p < 0.001$), 'arousal' (θ: $p < 0.01$, α: $p < 0.01$, β: $p < 0.01$) and 'dominance' (θ: $p < 0.01$, α: $p < 0.01$, β: $p < 0.05$). Our covariance feature also exceeded the coherence of all the bands on the DREAMER dataset for 'arousal' ($p < 0.05$) and 'dominance' ($p < 0.05$) and on the DEAP dataset for the three assessments (all: $p < 0.05$). Compared to the correlation, the covariance showed significant improvements on the DREAMER dataset for 'arousal' ($p < 0.01$) and 'dominance' ($p < 0.05$) and on the DEAP dataset for 'valence' ($p < 0.05$). We conclude that the SPD matrix based on the covariance feature can capture some important information about EEG DA emotion recognition.

Table 5
Mean accuracy for different SPD features on the DREAMER dataset.

Method	Accuracy(%)		
	Valence	Arousal	Dominance
Coherence on θ (4–8 Hz)	66.97 ± 5.31	73.56 ± 16.88*	78.97 ± 17.02**
Coherence on α (8–13 Hz)	67.75 ± 6.81	74.38 ± 16.82*	79.34 ± 16.57**
Coherence on β (13–30 Hz)	67.40 ± 6.03	73.96 ± 16.37**	78.74 ± 17.29**
Coherence on all	67.35 ± 6.28	73.39 ± 17.79*	79.02 ± 16.05*
Correlation	67.42 ± 5.37	74.96 ± 15.65**	80.80 ± 15.03*
Ours-Covariance	**67.99 ± 6.34**	**76.57 ± 14.04**	**81.77 ± 14.24**

*There are significant differences between our covariance features and other SPD features. (***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$)

Table 6
Mean accuracy for different SPD features on the DEAP dataset.

Method	Accuracy(%)		
	Valence	Arousal	Dominance
Coherence on θ (4–8 Hz)	61.09 ± 9.10***	66.70 ± 13.96**	67.02 ± 17.08**
Coherence on α (8–13 Hz)	61.69 ± 9.82***	67.26 ± 12.70**	67.49 ± 17.44**
Coherence on β (13–30 Hz)	65.04 ± 9.09	67.40 ± 13.30**	68.76 ± 15.75*
Coherence on all	63.46 ± 8.95**	66.72 ± 11.47*	67.93 ± 15.43*
Correlation	65.57 ± 8.67*	69.17 ± 11.22	70.68 ± 15.40
Ours-Covariance	**66.47 ± 8.75**	**69.79 ± 11.93**	**71.11 ± 14.68**

*There are significant differences between our covariance features and other SPD features. (***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$)

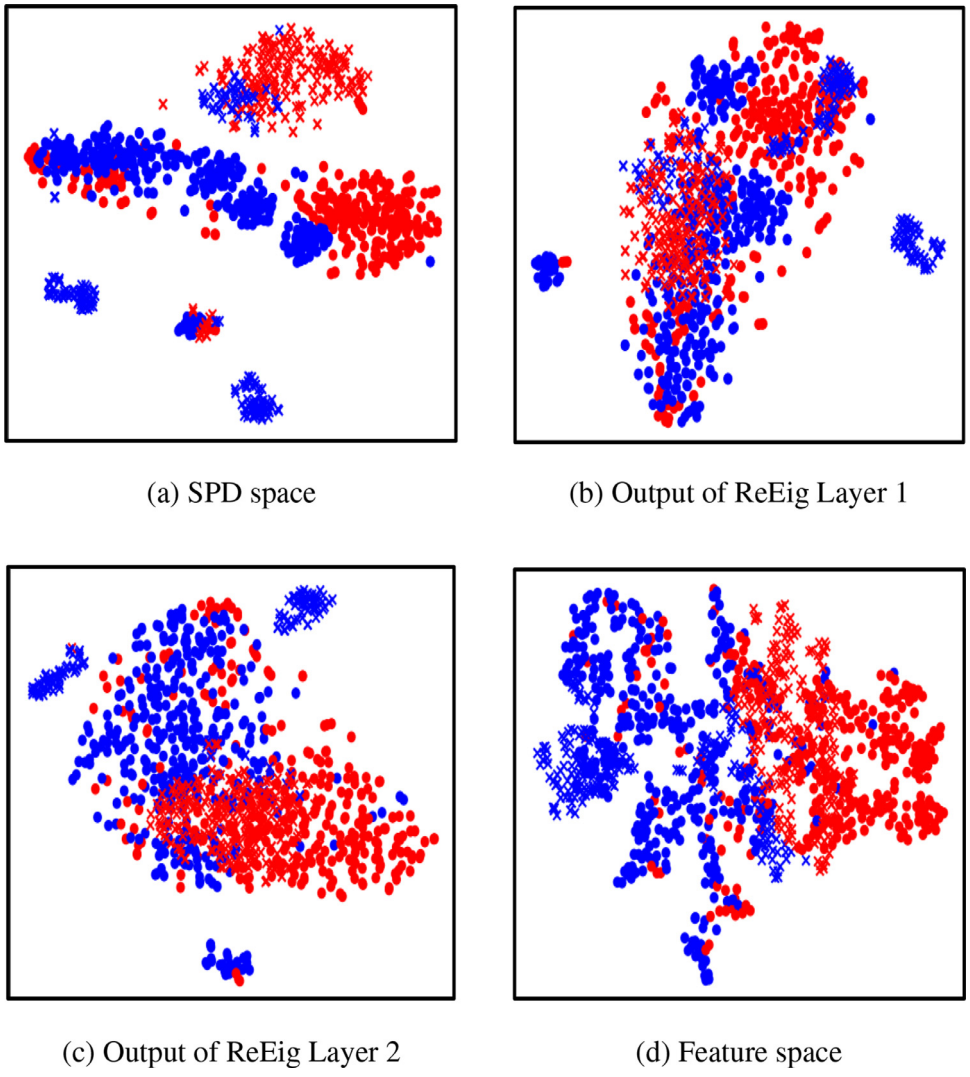

Fig. 5. Example of our model's hierarchical visualization for the (a) SPD space, (b) output of the first ReEig layer (c) output of the second ReEig layer and (d) feature space. The dots and crosses represent labelled source samples and unlabelled target samples, respectively. The blue and red represent high and low emotion states. (Please see the electronic version for a better look.) (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.9. Visualization

T-distributed stochastic neighbour embedding (t-SNE) is a technique for dimensionality reduction, which is particularly suited for the visualization of high-dimensional datasets. In this paper, we used t-SNE as the dimensionality reduction approach. For simplicity, we randomly chose one target subject and its one source in the DREAMER dataset for visualization. Here, we only show its "arousal" results in Fig. 5.

We first drew the raw SPD matrices in the original input space (Fig. 5(a)); the source samples and target samples were far from each other, and the high and low arousal states were not easy to separate. After a BiMap computation followed by a ReEig layer (Fig. 5(b)), the separation between samples from the source and target domains was no longer as evident. However, all the samples were combined, and it was still difficult to distinguish the categories. Then, the features went through another BiMap layer and ReEig layer (Fig. 5(c)); the distributions of the source and target domains were closer to each other, and the representations be-

longing to different categories were more linearly separable. We also visualized the final feature space after the LogEig operation, as shown in Fig. 5(d), and we observed that the source and target distributions became much more similar and that the unlabelled target domain (represented by crosses) could be linearly separated. From the point of view of classification, our method obtained an accuracy of 82.67% for this subject.

In summary, benefiting from the extractor, the target and source distributions were confused using our constraint, which implies that the source samples will provide good assistance for inferring labels of the target samples via our daSPDnet.

5. Discussion

In this paper, we develop a domain adaptation SPD matrix network to reduce the calibration time by using data from different subjects. The experimental results offer some inspiration for further research.

(1) Our model is appropriate for features in the form of SPD matrices, and the covariance matrix is calculated as features. Our model outperforms a series of SVM-based transfer learning methods using the PSD feature, which is one of the most commonly used features in EEG-based emotion recognition. One reason may be that the covariance matrix not only includes information from a single channel but also highlights the relationship information from each channel pair. Our feature can also be successfully applied to some EEG-based DA studies. It can be demonstrated that our model using covariance matrices can identify a more effective emotional representation across subjects. We can also infer that the connectivity of EEG signals between different channels has underlying inter-subject consistency for the same emotional state.

(2) The number of the source domain is an important factor in the EEG DA problem. Using data from all subjects as sources may not improve the accuracy and also bring computational burden. Hence, we exploited a source selection strategy to determine 7 sources rather than using data from all subjects as sources for knowledge transfer on the present datasets. Based on our source selection strategy, when we add a new source subject on the present two datasets, if the new source data is selected into the first 7 sources, it demonstrates that the data distribution of the new subject is closer to that of the target subject. Thus, adding the new subject will improve the performance. Contrarily, if the data distribution of the new source subject is not closer to the target subject, it will not be selected into the first 7 sources. This case will not affect the performance. Therefore, our source selection strategy can avoid performance degradation when the number of subjects increases.

(3) We used the source selection strategy to determine 7 sources for subsequent experiments. Here, we adjusted the quantity of labelled data for the target subjects using 2 sessions (one low valence/arousal/dominance and one high valence/arousal/dominance) to further discuss multi-source DA. With increasing number of source subjects, as shown in Fig. 6, for the three kinds of assessments on the DREAMER dataset and the 'valence' and 'dominance' assessments on the DEAP dataset, the accuracies showed a stable trend after the rising fluctuations. However, for the 'arousal' assessment on the DEAP dataset, there was a small decrease after the 20th subject. This phenomenon may be due to the negative transfer of subsequent subjects. It also reflects the necessity of our source selection strategy.

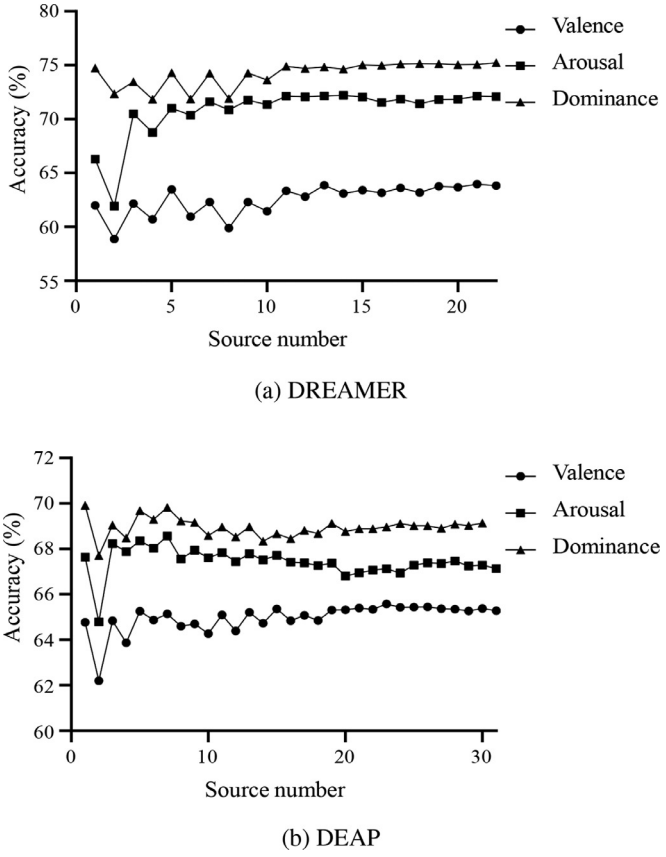

Fig. 6. Mean accuracy achieved with different source numbers on (a) the DREAMER dataset and (b) the DEAP dataset. Here, we chose 2 sessions (one low, one high) from the target subject.

(4) To determine how much the data were reduced, we exploited different numbers of sessions to train subject-dependent SVM classifiers, as shown in Fig. 7. On the DREAMER dataset, our method significantly ($p < 0.05$) outperformed the SVM using 3 calibration sessions in each category, achieving a 4.43% higher value for 'valence', 3.52% higher for 'arousal', and 5.65% higher for 'dominance'. On the DEAP dataset, our method showed a significant ($p < 0.05$) advantage over 20 calibration session trained SVMs. The accuracies were higher by 3.01%, 5.29%, and 4.43%. According to these results, when using 2 calibration sessions for each category and approximately 14 test sessions, our methods performed as well as the conventional method, which used 8 calibration sessions and approximately 10 test sessions on the DREAMER dataset. On the DEAP dataset, we used 2 calibration sessions for each category and approximately 36 test sessions, which led to a better performance than the SVM using 20 calibration sessions and approximately 20 test sessions. Our model saves at least 4 calibration sessions on the DREAMER dataset and at least 16 sessions on the DEAP dataset.

(5) We compared our method with a common subject-independent method, "leave-one-subject-out" (LOSO), based on our basic network pSPDnet. On the DREAMER dataset, our method achieved significantly higher accuracies than LOSO by 11.35%, 16.23% and 13.45% for the three assessments (all: $p < 0.001$). On the DEAP dataset, our method also achieved significantly higher classification performances than LOSO by 10.73%, 11.49% and 14.56% for the three assessments (all $p < 0.001$). This indicates that our daSPDnet can

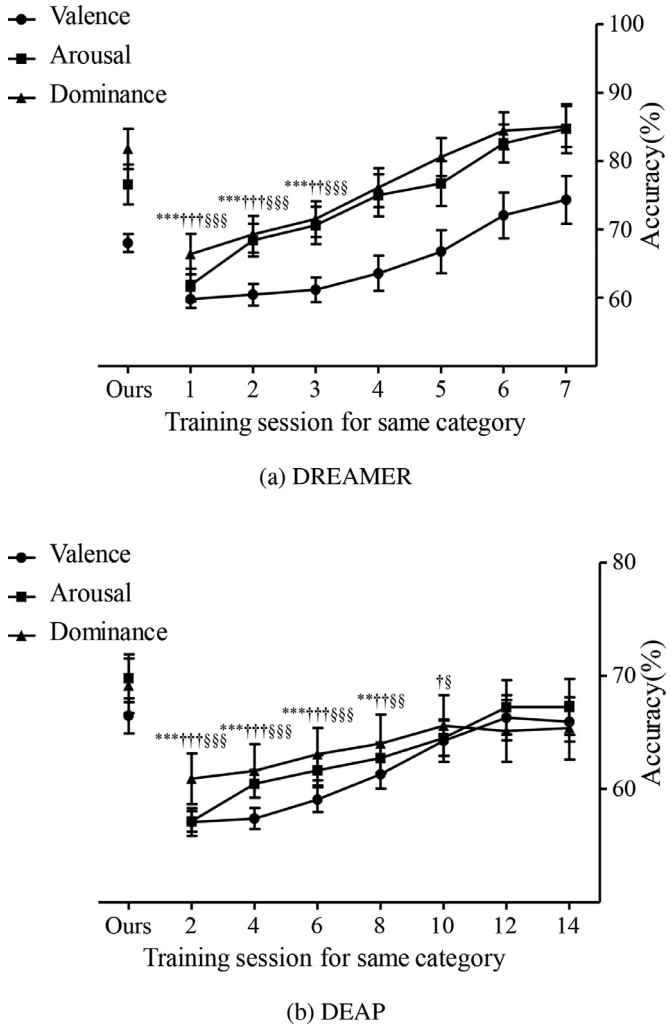

Fig. 7. Performance comparison for fewer training data on (a) the DREAMER dataset and (b) the DEAP dataset. To explore what the exact reduction in the calibration time is, we train a basic subject-dependent SVM. We carry out the significance test between our subject-independent method and a series of different partition approaches of the subject-dependent model. (*:'valence'; †: 'arousal'; §: 'dominance').

improve the performance by reducing the distribution discrepancy between two domains; moreover, achieving good results by simply using the data of the other subjects without considering the diversity between the subjects is difficult.

Concerning future work, we have limitations that can be improved. Currently, our method cannot handle cases under the condition of no calibration time. We can try to use the prototypes of the source domain to obtain pseudo-labels of the unsupervised target samples. By using these pseudo-labels, we can continue to perform subsequent operations.

6. Conclusion

We have proposed a brand-new domain adaptation SPD matrix network (daSPDnet) approach to solve the supervised transfer learning problem in EEG emotion recognition. Our model provides a solution to overcome the variability of subjects' physiological activities and helps reduce the time-wasting data collection, which is a great improvement in the EEG emotion recognition field.

Declaration of Competing Interest

Authors declare that they have no conflict of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under grants 61976209 and 81701785, in part by the CAS International Collaboration Key Project, grant no. 173211KYSB20190024, in part by the Strategic Priority Research Program of CAS, grant no. XDB32040000.

References

[1] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, J.G. Taylor, Emotion recognition in human-computer interaction, IEEE Signal Process. Mag. 18 (1) (2001) 32–80.
[2] P. Ekman, Expression and the nature of emotion, Approaches Emot. 3 (1984) 19–344.
[3] Z. Zeng, M. Pantic, G.I. Roisman, T.S. Huang, A survey of affect recognition methods: audio, visual, and spontaneous expressions, IEEE Trans. Pattern Anal. Mach. Intell. 31 (1) (2009) 39–58.
[4] R.A. Calvo, S. D'Mello, Affect detection: an interdisciplinary review of models, methods, and their applications, IEEE Trans. Affect. Comput. 1 (1) (2010) 18–37.
[5] S.M. Alarcao, M.J. Fonseca, Emotions recognition using EEG signals: a survey, IEEE Trans. Affect. Comput. 10 (3) (2017) 374–393.
[6] K.K. Ang, Z.Y. Chin, H. Zhang, C. Guan, Mutual information-based selection of optimal spatial–temporal patterns for single-trial EEG-based BCIs, Pattern Recognit. 45 (6) (2012) 2137–2144.
[7] B. Boashash, G. Azemi, N.A. Khan, Principles of time–frequency feature extraction for change detection in non-stationary signals: applications to newborn EEGabnormality detection, Pattern Recognit. 48 (3) (2015) 616–627.
[8] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, B. Arnaldi, A review of classification algorithms for EEG-based brain–computer interfaces, J. Neural Eng. 4 (2) (2007) R1.
[9] P. Wang, J. Lu, B. Zhang, Z. Tang, A review on transfer learning for brain-computer interface classification, in: 2015 5th International Conference on Information Science and Technology (ICIST), IEEE, 2015, pp. 315–322.
[10] W.-L. Zheng, B.-L. Lu, Personalizing EEG-based affective models with transfer learning, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, AAAI Press, 2016, pp. 2732–2738.
[11] P. Sinno Jialin, I.W. Tsang, J.T. Kwok, Y. Qiang, Domain adaptation via transfer component analysis, IEEE Trans. Neural Netw. 22 (2) (2011) 199–210.
[12] B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (5) (1998) 1299–1319.
[13] R. Collobert, F. Sinz, J. Weston, L. Bottou, T. Joachims, Large scale transductive SVMs, J. Mach. Learn. Res. 7 (1) (2006) 1687–1712.
[14] E. Sangineto, G. Zen, E. Ricci, N. Sebe, We are not all equal: personalizing models for facial expression analysis with transductive parameter transfer, in: Proceedings of the ACM International Conference on Multimedia, MM, 2014, pp. 357–366.
[15] W.-L. Zheng, B.-L. Lu, Investigating critical frequency bands and channels for EEG-based emotion recognition with deep neural networks, IEEE Trans. Auton. Ment. Dev. 7 (3) (2015) 162–175.
[16] X. Chai, Q. Wang, Y. Zhao, Y. Li, D. Liu, X. Liu, O. Bai, A fast, efficient domain adaptation technique for cross-domain electroencephalography (EEG)-based emotion recognition, Sensors 17 (5) (2017) 1014.
[17] M. Congedo, A. Barachant, R. Bhatia, Riemannian geometry for EEG-based brain-computer interfaces; a primer and a review, Brain-Comput. Interfaces 4 (3) (2017) 155–174.
[18] P. Zanini, M. Congedo, C. Jutten, S. Said, Y. Berthoumieu, Transfer learning: a Riemannian geometry framework with applications to brain–computer interfaces, IEEE Trans. Biomed. Eng. 65 (5) (2018) 1107–1116.
[19] O. Yair, M. Ben-Chen, R. Talmon, Parallel transport on the cone manifold of SPDmatrices for domain adaptation, IEEE Trans. Signal Process. 67 (7) (2019) 1797–1811.
[20] P.L.C. Rodrigues, C. Jutten, M. Congedo, Riemannian procrustes analysis: transfer learning for brain-computer interfaces, IEEE Trans. Biomed. Eng. (2018).
[21] A. Barachant, S. Bonnet, M. Congedo, C. Jutten, Riemannian geometry applied to BCI classification, in: International Conference on Latent Variable Analysis and Signal Separation, Springer, 2010, pp. 629–636.
[22] S. Katsigiannis, N. Ramzan, Dreamer: a database for emotion recognition through EEG and ECG signals from wireless low-cost off-the-shelf devices, IEEE J. Biomed. Health Inform. 22 (1) (2017) 98–107.
[23] S. Koelstra, C. Muhl, M. Soleymani, J.-S. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, I. Patras, Deap: a database for emotion analysis; using physiological signals, IEEE Trans. Affect. Comput. 3 (1) (2011) 18–31.
[24] R. Bhatia, Positive Definite Matrices, 16, Princeton University Press, 2009.
[25] V. Arsigny, P. Fillard, X. Pennec, N. Ayache, Geometric means in a novel vector space structure on symmetric positive-definite matrices, SIAM J. Matrix Anal. Appl. 29 (1) (2007) 328–347.
[26] V. Arsigny, P. Fillard, X. Pennec, N. Ayache, Log-euclidean metrics for fast and simple calculus on diffusion tensors, Magn. Reson. Med. 56 (2) (2006) 411–421.

[27] B. Wang, Y. Hu, J. Gao, M. Ali, D. Tien, Y. Sun, B. Yin, Low rank representation on SPD matrices with log-euclidean metric, Pattern Recognit. 76 (2018) 623–634.
[28] M. Harandi, M. Salzmann, R. Hartley, Dimensionality reduction on SPD manifolds: the emergence of geometry-aware methods, IEEE Trans. Pattern Anal. Mach. Intell. 40 (1) (2018) 48–62.
[29] D.A. Bini, B. Iannazzo, Computing the Karcher mean of symmetric positive definite matrices, Linear Algebra Appl. 438 (4) (2013) 1700–1710.
[30] A. Barachant, S. Bonnet, M. Congedo, C. Jutten, Multiclass brain-computer interface classification by Riemannian geometry, IEEE Trans. Biomed. Eng. 59 (4) (2012) 920–928.
[31] R. Ferreira, J. Xavier, J.P. Costeira, V. Barroso, Newton algorithms for Riemannian distance related problems on connected locally symmetric manifolds, IEEE J. Sel. Top. Signal Process. 7 (4) (2013) 634–645.
[32] S. Bonnabel, Stochastic gradient descent on Riemannian manifolds, IEEE Trans. Automat. Control. 58 (9) (2013) 2217–2229.
[33] H.-M. Yang, X.-Y. Zhang, F. Yin, C.-L. Liu, Robust classification with convolutional prototype learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3474–3482.
[34] C.-L. Liu, M. Nakagawa, Evaluation of prototype learning algorithms for nearest-neighbor classifier in application to handwritten character recognition, Pattern Recognit. 34 (3) (2001) 601–615.
[35] Z. Gao, Y. Wu, X. Bu, T. Yu, J. Yuan, Y. Jia, Learning a robust representation via a deep network on symmetric positive definite manifolds, Pattern Recognit. 92 (2019) 1–12.
[36] Z. Huang, L. Van Gool, A Riemannian network for SPD matrix learning, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.
[37] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, Domain-adversarial training of neural networks, J. Mach. Learn. Res. 17 (1) (2016) 2030–2096.
[38] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.
[39] S. Motiian, Q. Jones, S. Iranmanesh, G. Doretto, Few-shot adversarial domain adaptation, in: Advances in Neural Information Processing Systems, 2017, pp. 6670–6680.
[40] J. Posner, J.A. Russell, B.S. Peterson, The circumplex model of affect: an integrative approach to affective neuroscience, cognitive development, and psychopathology, Dev. Psychopathol. 17 (3) (2005) 715–734.
[41] A. Mehrabian, Pleasure-arousal-dominance: a general framework for describing and measuring individual differences in temperament, Curr. Psychol. 14 (4) (1996) 261–292.

Yixin Wang received the B.S. degree in School of Control Science and Engineering, Shandong University, Jinan, in 2016. She is currently pursuing the Ph.D. degree with the Institute of Automation, Chinese Academy of Sciences (CASIA). Her current research interests include affective computing, machine learning and Brain-Computer Interface (BCI).

Shuang Qiu received Ph.D. degree in Biomedical Engineering from Tianjin University, Tianjin, China. She is currently an assistant professor with CASIA. Her research interests concern biosignal processing, machine learning, and rehabilitation engineering, especially in Brain-Computer Interface (BCI).

Xuelin Ma received the B.S. degree in School of Automation Science and Engineering, South China University of Technology, Guangzhou, in 2015. He is currently pursuing the Ph.D. degree with the Institute of Automation, Chinese Academy of Sciences (CASIA). His current research interests include machine learning and Brain-Computer Interface (BCI).

Huiguang He received the Ph.D. degree (with honor) in pattern recognition and intelligent systems from Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China. He is currently a full professor with CASIA. His research interests include pattern recognition, medical image processing, and Brain-Computer Interface (BCI).