EEG Mental Recognition Based on RKHS Learning and Source Dictionary Regularized RKHS Subspace Learning

WENJIE LEI, ZHENGMING MA, SHUYU LIU, AND YUANPING LIN

1School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China
2Public Experimental Teaching Center, Sun Yat-sen University, Guangzhou 510006, China
Corresponding author: Zhengming Ma (issmzm@mail.sysu.edu.cn)

This work was supported in part by the Science and Technology Program of Guangzhou, China, under Grant 68000-42050001, and in part by the Natural Science Foundation of China under Grant 61773022.

ABSTRACT This article mainly studies Electroencephalogram (EEG) mental recognition. Because the human brain is very complex and the EEG signal is greatly affected by the environment, EEG mental recognition can be attributed to domain adaptative problems. Our main work is as follows: (1) At present, most domain adaptation learning only learns the linear subspace of Reproducing Kernel Hilbert Space (RKHS), and RKHS itself does not. Given the complexity and nonlinearity of EEG mental recognition, we propose an EEG mental recognition algorithm based on two learning. The two learning is RKHS learning and RKHS subspace learning. The source dictionary regularized RKHS subspace learning we proposed applies to EEG mental recognition and is better than pure RKHS subspace learning, but not enough. To get satisfactory results, we learn RKHS before RKHS subspace. (2) According to Moore–Aronszajn theorem, RKHS can be uniquely generated by a kernel function. The existing RKHS is rarely learnable. It is difficult to find a kernel function that can be learned and optimized. In RKHS learning, this paper uses a learnable kernel function that we published, and the kernel function is easy to optimize. (3) In RKHS subspace learning, most of the existing methods adopt the Maximum Mean Discrepancy (MMD) criterion, but it cannot make the spatial distribution of the same category of source and target domain data overlap as much as possible, and the label of the target domain data is unknown. To solve this problem, this paper proposes a framework of RKHS subspace learning based on source domain dictionary regularization. The experimental results on the brain-computer interface international competition data set (BCI competition IV 2a) show that the effect of the algorithm proposed in this paper is better than that of the other five advanced domain adaptation learning algorithms.

INDEX TERMS Brain computer interface, electroencephalogram, domain adaptation learning, subspace learning.

I. INTRODUCTION Transfer Learning is a machine learning method that uses existing knowledge to solve different but related domains or tasks [2], [3]. Domain and task [1] are the two most critical points of transfer learning. We can treat data samples with the same feature space and marginal distribution as the same domain, and the same task represents the label space and posterior of the two tasks. The conditional probability distribution is consistent. Domain adaptive learning is a kind of transfer learning, which transfers data features. Its research on how to use labeled source domain data and prior knowledge of the target domain to reliably learn and complete tasks in the target domain when the probability distributions of the source and target domain are different but related.

Domain adaptation can be divided: semi-supervised [7]–[9] and unsupervised [10]–[14]. To identify the subspace structure of noisy data, Liu et al. [16], [17] proposed low-rank representation (LRR). Jhuo et al. [18] proposed robust domain adaptation based on LRR. Shekhar et al. [23], [24] represent the source domain and the target domain data respectively, through the shared dictionary in the latent subspace. Domain-specific dictionary learning [25], [26] is to learn a dictionary for each domain, and then use...
domain-specific or domain-common representation coefficients to represent each domain. However, none of the above methods makes the source domain data and target domain data close to each other by category, which is different from the method proposed in this article. MMD is one of the most commonly used criteria for domain adaptation. MMD essentially uses the first-order origin moment of two random variables to measure the similarity between two random variables, which brings great computational convenience. But MMD also has some flaws. Therefore, many researchers add various regularization on the basis of MMD to meet the needs of different application scenarios, such as literature [15], [33], and [34]. In addition, the other domain adaptation criterion has been proposed, but this type of algorithm is more complex. Li and Zhang [9] proposed a covariance criterion, which minimizes the difference between the variance of the source domain and the variance of the target, and achieves the same distribution of the two domains. Wang et al. [5] proposed that the distribution of the target domain projected to the subspace can be approximated to the source through a transformation matrix.

BCI is a direct communication and control channel established between the human brain and the computer. Through BCI, you can directly use the human brain to express your intentions or control equipment, without language or body movements [19], [20]. Non-invasive BCI is mainly to record a series of computer graphics activities through the EEG acquisition device [27], the purpose is to detect and classify some specific EEG patterns. The EEG recognition task is a classification task. Basic machine learning classification algorithms include SVM, KNN, etc., but these algorithms do not consider the structure and distribution of the data. If you consider the category information of the data label, you can use the LDA algorithm, and if you consider the structure information of the data, you can consider the LE algorithm. But these are not the most suitable for EEG signal data. In order to solve the problem of EEG signal data classification, the common spatial pattern (CSP) algorithm was proposed [28]. But Kam believes that CSP does not consider the time information of the EEG signal. To solve this problem, they proposed a Time-Dependent Common Spatial Patterns (TDCSP) algorithm [29]. The algorithm can assign weights on each time band to consider the distribution of ERD/ERS in the time and frequency domains, and use the selected function to construct a separate classifier on each time band. Jiao et al. [21] proposed multi-scale optimization of spatial patterns (MSO), which optimizes the filter band through multi-view learning in CSP. Through $L_2,1$-norm regularization, sparse optimization based on multi-view learning is to capture the shared information of multiple related spatial patterns. Zhang et al. [22] added Laplace regularization based on MOS, and proposed subclass relation regularized multi-task learning (srMTL). The algorithm effectively reveals and utilizes the internal distribution structure of the data to obtain a more accurate MI-related EEG classification.

Later, researchers discovered that the probability distribution of EEG signal data are different but related. If the BCI recognition problem is solved from the domain adaptation aspect, the classification accuracy of the data can be improved. In order to solve the transfer learning problem in EEG-based brain-computer interface classification, Zanini et al. [30] proposed a Riemannian geometry framework and a classifier for BCI. Zanini uses the spatial covariance matrix of the EEG signal to represent data, and uses Riemannian geometry based on symmetric positive definite (SPD) manifolds to perform affine transformation on the covariance matrix of each session/object. This makes data from different sessions/objects comparable. Then in order to improve the classification accuracy, the author also proposed a probabilistic classifier to model the class probability distribution, which used the mixture of Riemann Gaussian and Gaussian distribution proposed by Said et al. [31]. Transfer Component Analysis (TCA) proposed by Pan et al. [32] also learns a low-dimensional RKHS subspace for reducing the distribution differences and maintaining the internal data structure. Semi-Supervised Transfer Component Analysis [32] (SSTCA) maximizes the correlation between data and label information, and retains locality, based on TCA. And Integration of Global and Local Metrics for Domain Adaptation Learning [33] (IGLDA) introduces category information for minimizing the intra-class distance of the projected data. The Transfer Independently Together (TIT) algorithm [34] is that the distance between the target domain data on RKHS and the target domain data formed by the subspace orthogonal basis after dimensionality reduction is as close as possible. Guide Subspace Learning (GSL) algorithm [35] use two projection matrices to map the two domain data to different subspace for approaching the projected data.

The main contributions of this paper are as follows:

1) At present, most domain adaptation learning only learns the linear subspace of RKHS, and RKHS itself does not. Given the complexity and nonlinearity of EEG mental recognition, this paper proposes an EEG mental recognition algorithm based on two learning. The two learning is RKHS learning and RKHS subspace learning. The source dictionary regularized RKHS subspace learning proposed in our previous paper [43] applies to EEG mental recognition and is better than pure RKHS subspace learning, but not enough. To get satisfactory results, we learn RKHS itself first before learning RKHS subspace. Compared with single learning, two learning makes the marginal distribution of source domain data and target domain data closer.

2) RKHS can be uniquely produced by a kernel function. The existing RKHS is rarely learnable. The reason is that it is difficult to find a kernel function that is learnable and easy to optimize. In RKHS learning, this paper uses a learnable kernel function that we have published, and the kernel function is easy to optimize. The kernel function can be used to generate a learnable RKHS
In RKHS subspace learning, most of the existing methods use the MMD criterion, but it cannot make the spatial distribution of the same category of source and target domain data overlap. And the label of the target domain data is unknown. So we adopt the source domain dictionary regularization proposed by ourselves [43] and propose a RKHS subspace learning algorithm based on the source domain dictionary regularization. On the basis of MMD, the algorithm requires the target domain data to be distributed around the source with the strongest linear correlation.

The algorithm proposed has not been reported in similar literature. RKHS learning can make the marginal distribution of the source and target domain minimize, while RKHS subspace learning further brings the marginal distribution of the source and target domain closer to each other on the basis of RKHS learning. Moreover, in RKHS subspace learning, the source domain and target domain are close to each other according to the spatial distribution of the same category, rather than the overall spatial distribution.

The following chapters of the article are arranged as follows. In the second chapter, we briefly introduce the basic mathematical theories related to this article, including Reproducing Kernel Hilbert Space (RKHS) and dictionary learning (DL). Then we introduce in detail the EEG mental recognition algorithm based on RKHS learning and source domain dictionary regularized RKHS subspace learning in Chapter 3. In Chapter 4, we briefly introduced the five comparison algorithm models, and the theoretical difference between the proposed algorithm and the comparison algorithm. In Chapter 5, we conducted a series of experiments on the BCI Competition IV 2a dataset and compared it with five state-of-the-art algorithms to verify the effectiveness and practicability of our proposed algorithm. Finally, we summarized our work in Chapter 6.

**II. NOTATIONS AND PRELIMINARIES**

**A. NOTATIONS**

In this paper, \(X_s\) and \(X_t\) are the set of source domain data and the set of target domain data, respectively. \(X = \{X_s, X_t\}\) represents the set of the source domain and target domain data. \(y^s\) and \(y^t\) represent respectively the source domain data and the target domain data projected to the subspace (shown in Table 1).

**B. RKHS AND KERNEL FUNCTIONS**

The Hilbert space is a complete inner product space, and Reproducing Kernel Hilbert Space (RKHS) is a special Hilbert space. Let \(H\) be a Hilbert space composed of functions defined on the set \(\Omega\) and satisfying certain conditions (such as square integrability), that is, \(f \in \Omega, f : \Omega \rightarrow R\). If there is a function \(k : \Omega \times \Omega \rightarrow R\), it satisfies

1. For any \(x \in \Omega\), \(k(x, \cdot) \in H\);
2. For any \(x \in \Omega\) and any \(f \in H\), \(f(x) = \langle f, k(x, \cdot) \rangle\), where \(\langle \cdot, \cdot \rangle\) is the inner product of \(H\).

Then \(H\) is called RKHS, and \(k\) is the reproducing kernel of \(H\) [36].

The reproducing kernel has properties such as symmetry, positive semi-definiteness and uniqueness. Using the reproducing kernel \(k\), the transformation can be defined:

\[
\varphi : \Omega \rightarrow H, \text{ for any } x \in \Omega, \varphi(x) = k(\cdot, x) \in H.
\]

Using the properties of reproducing kernel, it can be proved that for any \(x, y \in \Omega\),

\[
\langle \varphi(x), \varphi(y) \rangle = k(x, y)
\]

### Table 1. The basic notations.

| Notations           | Description                        |
|---------------------|------------------------------------|
| \(X_s\)            | the set of source domain data      |
| \(X_t\)            | the set of target domain data      |
| \(X\)              | the set of source and target domain data |
| \(N_s\)            | the number of source domain data   |
| \(N_t\)            | the number of target domain data   |
| \(N\)              | the number of datapoints           |
| \(W\)              | the projected matrix               |
| \(y^s\)            | source domain data projected to subspace |
| \(y^t\)            | target domain data projected to subspace |
| \(\alpha\)         | the dictionary encoding matrix     |
| \(\mathbb{R}\)     | real number field                  |
| \(k\)              | kernel function                    |
| \(\delta\)         | an optimizable kernel function     |
| \(K\)              | kernel matrix                      |
| \(K^\ast\)         | an optimizable kernel matrix       |
| \(X^T\)            | transpose                          |
| \(X^{-1}\)         | inverse                            |
| \(\langle \cdot, \cdot \rangle\) | inner product                      |
According to the Moore-Aronszajn theorem, RKHS can be uniquely generated by the kernel function. The definition of the kernel function is as follows [37]: \( k : \Omega \times \Omega \rightarrow R \), it satisfies

1. Symmetry: For any \( x, y \in \Omega \), \( k(x, y) = k(y, x) \);
2. Positive definiteness: For any finite number of elements \( \{x_1, \cdots, x_N\} \subseteq \Omega \), the following matrix is positive definite:

\[
K = \begin{bmatrix}
k(x_1, x_1) & \cdots & k(x_1, x_N) \\
\vdots & \ddots & \vdots \\
k(x_N, x_1) & \cdots & k(x_N, x_N)
\end{bmatrix}
\]

Note: The kernel function is not the same concept as the reproducing kernel. The reproducing kernel relies on the Hilbert space definition, and the kernel function is a separately defined function.

The process of kernel function to generate RKHS is as follows:

1. Generate a linear space by using the kernel function:

\[
H_k = \text{span}\{k(\bullet, x)|x \in \Omega\}
\]

\[
= \left\{ \sum_{i=1}^{N} \lambda_i k(\bullet, x_i)|x_i \in \Omega, \lambda_i \in R, i \in Z^+ \right\} \quad (2)
\]

2. Define an inner product on \( H_k \) : for all \( f, g \in H_k \), since

\[
f(\bullet) = \sum_{i=1}^{N} \lambda_i k(\bullet, x_i), \quad \text{and} \quad g(\bullet) = \sum_{j=1}^{M} \beta_j k(\bullet, y_j),
\]

\[
(f, g) = [\lambda_1, \cdots, \lambda_N] \times \begin{bmatrix}
k(x_1, y_1) & \cdots & k(x_1, y_M) \\
\vdots & \ddots & \vdots \\
k(x_N, y_1) & \cdots & k(x_N, y_M)
\end{bmatrix} \begin{bmatrix}
\beta_1 \\
\vdots \\
\beta_M
\end{bmatrix} \quad (3)
\]

3. Complete \( H_k \) to obtain the complete space of \( \hat{H}_k \), denoted as \( \hat{H}_k \), then \( \hat{H}_k \) is an RKHS and the kernel function \( k \) is exactly the reproducing kernel of \( \hat{H}_k \).

Since the kernel function generates only one RKHS, learning an RKHS means learning a kernel function.

### C. DICTIONARY LEARNING

Let \( X = \{x_1, \cdots, x_N\} \) be a sample, and find a suitable dictionary \( \{d_1, \cdots, d_L\} \) through the \( N \) samples. The objective function of dictionary learning can be expressed as follows:

\[
\min_{d_1, \cdots, d_L} \sum_{j=1}^{N} \left[ \|x_j - \sum_{j=1}^{L} \alpha_{ij} d_j\|^2 + \lambda \text{sparse}(A_{lRow}) \right] \quad (4)
\]

where \( A = \begin{bmatrix}
\alpha_{11} & \cdots & \alpha_{1L} \\
\vdots & \ddots & \vdots \\
\alpha_{NL} & \cdots & \alpha_{NL}
\end{bmatrix} \) is called the sparse coding matrix.

After the dictionary \( \{d_1, \cdots, d_L\} \) has been learned, each trusted data point \( x \) can be roughly represented by this set of dictionaries. The function of sparse coding is to make the coefficient components of \( x \) linearly represented by this set of dictionaries \( \{d_1, \cdots, d_L\} \) tend to 0 as much as possible. The objective function of sparse coding can be expressed as follows

\[
J(w) = \|x - \sum_{j=1}^{L} \alpha_j d_j\|^2 + \lambda \text{sparse}(\alpha) \quad (5)
\]

Here \( \alpha = [\alpha_1, \cdots, \alpha_L]^T \), \( \text{sparse}(\alpha) \) represents the sparse regular term of the sparse encoding, so that the component tend to zero. We usually use the 1 norm to represent the sparse regular term of sparse coding

\[
\text{sparse}(\alpha) = \|\alpha\|_1 = \sum_{j=1}^{L} |\alpha_j| \quad (6)
\]

And \( \text{sparse}(\alpha) \) is the feature vector after sparse coding. If there is no \( \text{sparse}(\alpha) \), it will become to solve subspace problem. If \( \{d_1, \cdots, d_L\} \) is orthogonal to each other, then

\[
\min_{\alpha} \|x - \sum_{j=1}^{L} \alpha_j d_j\|^2 \rightarrow \alpha_j = x^T d_j, \quad j = 1, \cdots, L \quad (7)
\]

### D. BCI DATA

A typical brain-computer interface (BCI) experiment includes several experiments. In these experiments, the EEG signals are used to infer which tasks the subjects are performing. The typical BCI data set is composed of a set of coupled pairs if the records are \( d \) electrodes and each of the \( N \) tests is composed of \( T \) time samples.

\[
\mathcal{X} = \{X_i, l_i\} \text{ for } i = 1, \cdots, N \in \mathbb{R}^d \times T \times \{1, \cdots, C\} \quad (8)
\]

where \( X_i \) is the \( i \)th EEG test recorded in the experiment and \( l_i \) is its label (there are \( C \) classes). Different BCI examples consist of different cognitive tasks that perform different filters on recorded signals. Generally speaking, the distribution of collected experimental data differs greatly due to the different experimental conditions, such as experimental period and experimental subjects, etc. Therefore, the classifier trained by labeled training data has low performance in test data.

### III. EEG MENTAL RECOGNITION BASED ON RKHS LEARNING AND SOURCE DICTIONARY REGULARIZED RKHS SUBSPACE LEARNING

When the subjects complete the corresponding motor imagination tasks, they may be affected by the environment or their own factors, which may lead to different EEG signals generated by the same motor imagination task imagined on different days. Due to the instability of EEG signals, the problem of EEG mental recognition can be attributed to the problem of domain adaptation. At present, most domain adaptation learning only learns the linear subspace of RKHS, and RKHS itself is not learned. In view of the complexity and nonlinearity of EEG mental recognition, this paper proposes an EEG intention recognition algorithm based on the two learning framework, which is called EEG Mental Recognition.
Based on RKHS Learning and Source Dictionary Regularized RKHS Subspace Learning (EEG-KLSD).

**A. DOMAIN ADAPTATION BASED ON RKHS LEARNING**

1) **SAMPLE-DEPENDENT AND LEARNABLE KERNEL FUNCTIONS**

As mentioned in section II.B, RKHS is uniquely generated by the kernel function. Therefore, RKHS learning can be transformed into kernel function learning. This paper proposes a learning kernel function framework based on sample dependence.

Let $k(x, y)$ be a symmetric positive definite kernel function defined on $\Omega \times \Omega$, and $k_0(x, y)$ is an arbitrary binary function. Given a set of learning samples $X = \{x_1, \ldots, x_N\} \subseteq \Omega$ on data space $\Omega$, and $Z = \{z_1, \ldots, z_N\} \subseteq X$ is part of or all of the samples in data $X$, the number of samples is $N_z$. Let $M \in \mathbb{R}^{N_x \times N_z}$ be a symmetric positive semi-definite matrix. For any $x \in X$, let $k_Z(x) = \left| k_0(x, z_1) \cdots k_0(x, z_N) \right| \in \mathbb{R}^{N_z}$. Define a new kernel function as follows [39]: For any $x, y \in \Omega$, $k(x, y) = k(x, y) + k_T^z(x)Mk_Z(y)$.

It can be proved that the kernel function $k$ is symmetric positive semi-definite. Existing $\{u_1, \ldots, u_Q\} \subseteq \Omega$ is a set of samples, the number of samples is $Q$, let $K_{UZ} = \begin{bmatrix} k_0(u_1, z_1) & \cdots & k_0(u_1, z_N) \\ \vdots & \ddots & \vdots \\ k_0(u_Q, z_1) & \cdots & k_0(u_Q, z_N) \end{bmatrix}$, then the kernel matrix of sample $X$ with respect to kernel function $k$ is $K = K_{UZ}MK_{UZ}^T$.

 Knowing that $k(x, y)$ is a symmetric positive definite kernel function, then the kernel matrix $K$ is symmetric positive definite, and it is known that $M$ is a symmetric positive semi-definite matrix. So $K_T^z = K_T^z + (K_{UZ}MK_{UZ}^T)^T = K + (K_{UZ}MK_{UZ}^T)^T = K$, that is, $k$ is symmetric; for any $y \in \Omega$, there is $y^TKy = y^TKy + y^TK_{UZ}MK_{UZ}^Ty = y^TKy + (K_{UZ}^T)^TM(K_{UZ}^T)y$.

According to the definite properties of $K$ and the semi-definite properties of $M$, we know that $y^TKy > 0$, $(K_{UZ}^T)y^TM(K_{UZ}^T)y \geq 0$, so $y^T\tilde{k}y > 0$, that is, $\tilde{k}$ is positive definite. In summary, $\tilde{k}$ satisfies Mercer’s condition, so it is a legal kernel function.

2) **RKHS LEARNING BASED ON SAMPLE-DEPENDENT AND LEARNABLE KERNEL FUNCTIONS**

Supposed a source domain data set and a target domain data set in the original data $\Omega$

$$X_s = \{x_s^1, \ldots, x_s^{N_s}\} \subseteq \Omega,$$

$$X_t = \{x_t^1, \ldots, x_t^{N_t}\} \subseteq \Omega, \quad N = N_s + N_t$$

and the source domain data $X_s$ is labeled, and the labels of the unlabeled target domain data $X_t$ need to be identified by $X_s$. However, the distribution of $X_s$ and $X_t$ are different, and identifying directly on $\Omega$ will cause larger error, inevitably. According to section II.B, using the reproducing kernel $k$, the transformation can be defined: $\psi : \Omega \rightarrow H$, for any $x \in \Omega, \psi(x) = k(\bullet, x) \in H$. Therefore, after mapping the source domain data and target domain data to RKHS, the source domain data on RKHS can be expressed as $\psi(X_s) = \{\psi(x_s^1), \ldots, x_s^{N_s}\} \subseteq H$, and the target domain data can be expressed as $\psi(X_t) = \{\psi(x_t^1), \ldots, x_t^{N_t}\} \subseteq H$.

We use the MMD criterion to measure the mean square error of the two data on RKHS, as shown below.

$$\frac{1}{N_s} \sum_{i=1}^{N_s} \psi (x_s^i) - \frac{1}{N_t} \sum_{j=1}^{N_t} \psi (x_t^j)^2$$

$$= \frac{1}{N_s^2} \sum_{i=1}^{N_s} \sum_{j=1}^{N_s} k(x_s^i, x_s^j) + \frac{1}{N_t^2} \sum_{i=1}^{N_t} \sum_{j=1}^{N_t} k(x_t^i, x_t^j)$$

$$\quad - \frac{2}{N_s N_t} \sum_{i=1}^{N_s} \sum_{j=1}^{N_t} k(x_s^i, x_t^j)$$

$$= \Gamma^T_s K_s \Gamma_s + \Gamma^T_t K_t \Gamma_t - 2\Gamma^T K_{st} \Gamma_t = \Gamma^T K \Gamma$$

Where, $K_s = \begin{bmatrix} k(x_s^1, x_s^1) & \cdots & k(x_s^1, x_s^{N_s}) \\ \vdots & \ddots & \vdots \\ k(x_s^{N_s}, x_s^1) & \cdots & k(x_s^{N_s}, x_s^{N_s}) \end{bmatrix} \in \mathbb{R}^{N_s \times N_s}$

$K_t = \begin{bmatrix} k(x_t^1, x_t^1) & \cdots & k(x_t^1, x_t^{N_t}) \\ \vdots & \ddots & \vdots \\ k(x_t^{N_t}, x_t^1) & \cdots & k(x_t^{N_t}, x_t^{N_t}) \end{bmatrix} \in \mathbb{R}^{N_t \times N_t}$

$K_{st} = \begin{bmatrix} k(x_s^1, x_t^1) & \cdots & k(x_s^1, x_s^{N_s}) \\ \vdots & \ddots & \vdots \\ k(x_t^{N_s}, x_t^1) & \cdots & k(x_t^{N_s}, x_t^{N_t}) \end{bmatrix} \in \mathbb{R}^{N_s \times N_t}$
on the SPD manifold is

\[ M \]

the optimal kernel function. On the Riemannian manifold, this paper uses the constraints that must be satisfied, and the data representation can be obtained. The model is as follows:

\[
\text{obj}(M) = \Gamma^T \bar{K} \Gamma \\
= \Gamma^T \left( K + K_{st} M K_{st}^T \right) \Gamma + \mu \text{tr} \left( M^T M \right) \\
\text{s.t. } M^T = M, M \succeq 0
\]

where,

\[
K_{st} = \begin{bmatrix} K_{sc}^T & K_{tc} \\ K_{tc}^T & K_{tc} \end{bmatrix}, \quad K_{sc} = \begin{bmatrix} k_Z^2(x_1) \\ \vdots \\ k_Z^2(x_{N_t}) \end{bmatrix} \\
K_{tc} = \begin{bmatrix} k_Z^2(x_1) \\ \vdots \\ k_Z^2(x_{N_t}) \end{bmatrix}
\]

3) SOLUTION

Find the Euclidean gradient of \( M \) for \( \text{obj}(M) \), there is

\[
\nabla_M \text{obj}(M) = K_{st}^T \Gamma \Gamma^T K_{st} + 2 \beta M
\]

In order to satisfy the constraint that the parameter \( M \) satisfies the symmetric positive definite and facilitate the solution, this paper requires the parameter solution set to satisfy the symmetric positive definite property, and the symmetric positive definite matrix can form a Riemannian manifold. The function on the Riemannian manifold can be optimized by the Riemannian conjugate gradient method. In this paper, the Riemannian conjugate gradient is used to optimize the parameter \( M \). Let \( E \) be a function on the Riemannian manifold, and function \( E \) and the Euclidean gradient of \( x \) in Euclidean space as \( \nabla_E E \). According to the literature [38], the Riemannian gradient of function \( E \) with respect to \( x \) is

\[
\nabla_E E = \frac{1}{2} x \left( \partial_x E + \partial_x^T E \right) x
\]

Then, the Riemannian gradient of \( \text{obj}(M) \) with respect to \( M \) on the SPD manifold is

\[
\nabla_M \text{obj}(M) = \frac{1}{2} M \left( \partial_M \text{obj}(M) + \partial_M^T \text{obj}(M) \right) M
\]

According to the objective function \( \text{obj}(M) \) and its Riemannian gradient on the SPD manifold, this paper uses the conjugate gradient method to iteratively optimize \( M \) to select the optimal kernel function \( \hat{k} \).

B. DOMAIN ADAPTATION BASED ON SOURCE DICTIONARY

REGULARIZED RKHS SUBSPACE LEARNING

In this section, we will introduce the construction and constraints of the RKHS subspace, and the data representation of the RKHS subspace.

Let \((H, \langle \cdot, \cdot \rangle)\) be the RKHS on the data space \( \Omega \), and use the reproducing kernel \( k \) of \( H \) to define the transformation from the data space \( \Omega \) to RKHS \( H : \Omega \to H \), for any \( x \in \Omega \), \( \phi(x) = k(\bullet, x) \in H \). For any \( x, y \) in \( \Omega \), \( \langle \phi(x), \phi(y) \rangle = k(x, y) \)

Give a dataset in the data space \( \Omega \)

\[
X = \{x_1, \cdots, x_N\} \subseteq \Omega
\]

Use \( \phi \) to transform \( X \) to \( H \):

\[
\phi(X) = \{\phi(x_1), \cdots, \phi(x_N)\} \subseteq H
\]

And we record \( K \) as

\[
K = \begin{bmatrix} \langle \phi(x_1), \phi(x_1) \rangle & \cdots & \langle \phi(x_1), \phi(x_N) \rangle \\
\vdots & \ddots & \vdots \\
\langle \phi(x_N), \phi(x_1) \rangle & \cdots & \langle \phi(x_N), \phi(x_N) \rangle \\
\end{bmatrix} = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_N) \\
\vdots & \ddots & \vdots \\
\end{bmatrix} = [K_{1Col} \cdots K_{NCol}] \in \mathbb{R}^{N \times N}
\]

where \( K_{iCol} \) represents the \( i \)th column of \( K \), \( i = 1, \ldots, N \).

Now, we use \( \phi(X) \) to structure a subspace of \( H \), Let

\[
\theta_i = \sum_{j=1}^{N} w_{ij} \phi(x_j), \quad i = 1, \ldots, d
\]

And we denote \( W \) as

\[
W = \begin{bmatrix} w_{11} & \cdots & w_{1d} \\
\vdots & \ddots & \vdots \\
\vdots & \ddots & \vdots \\
w_{N1} & \cdots & w_{Nd} \end{bmatrix} = [W_{1Col} \cdots W_{dCol}] \in \mathbb{R}^{N \times d}
\]

where \( W_{iCol} \) represents the \( i \)th column of \( W \), \( i = 1, \ldots, N \).

We use \( \Theta = \{\theta_1, \cdots, \theta_d\} \) to span a subspace of \( H \)

\[
\text{span} \Theta = \left\{ \sum_{i=1}^{d} a_i \theta_i \bigg| a_i \in \mathbb{R}, i = 1, \ldots, d \right\}
\]

We hope that \( \Theta \) constitutes the standard orthogonal basis of \( \text{span} \Theta \), then

\[
\begin{bmatrix} \langle \theta_1, \theta_1 \rangle & \cdots & \langle \theta_1, \theta_d \rangle \\
\vdots & \ddots & \vdots \\
\langle \theta_d, \theta_1 \rangle & \cdots & \langle \theta_d, \theta_d \rangle \end{bmatrix} = I_d
\]
For all
\[
\{ \theta_1, \theta_2 \} = \left( \sum_{p=1}^{N} w_p \varphi(x_p), \sum_{q=1}^{N} w_q \varphi(x_q) \right)
\]
\[= \sum_{p=1}^{N} \sum_{q=1}^{N} w_p w_q \varphi(x_p), \varphi(x_q) \]
\[= \sum_{p=1}^{N} \sum_{q=1}^{N} w_p w_q k(x_p, x_q) = W_{dCol} KW_{dCol} \] (23)

we have
\[
\begin{bmatrix}
\langle \theta_1, \theta_1 \rangle & \cdots & \langle \theta_1, \theta_d \rangle \\
\vdots & \ddots & \vdots \\
\langle \theta_d, \theta_1 \rangle & \cdots & \langle \theta_d, \theta_d \rangle
\end{bmatrix}
= \begin{bmatrix}
W_{1Col}^T KW_{1Col} & \cdots & W_{1Col}^T KW_{dCol} \\
\vdots & \ddots & \vdots \\
W_{dCol}^T KW_{1Col} & \cdots & W_{dCol}^T KW_{dCol}
\end{bmatrix}
= W^T KW = I_d \] (24)

Obviously, the subspace \(\text{span} \Theta\) is a \(d\)-dimensional subspace and completely determined by the combinational coefficient \(W\), and \(W\) must satisfy the above constraints.

Based on the RKHS orthonormal basis given above, this section gives a data expression form based on the RKHS subspace.

\[
y = \begin{bmatrix}
\varphi(x), \varphi(x_1) \\
\vdots \\
\varphi(x), \varphi(x_N)
\end{bmatrix}
= \begin{bmatrix}
\sum_{j=1}^{N} w_{j1} \varphi(x_j) \\
\vdots \\
\sum_{j=1}^{N} w_{jd} \varphi(x_j)
\end{bmatrix}
\in \mathbb{R}^d
\]

According to the projected theorem, if \(\{ \theta_1, \ldots, \theta_d \}\) is the standard orthogonal basis of the subspace \(\text{span} \Theta\), the coordinate of \(\varphi(x_i)\) projected on the subspace \(\text{span} \Theta\) is

\[
y_i = \begin{bmatrix}
\sum_{j=1}^{N} w_{j1} \varphi(x_j, x_i) \\
\vdots \\
\sum_{j=1}^{N} w_{jd} \varphi(x_j, x_i)
\end{bmatrix}
= \begin{bmatrix}
W_{1Col}^T k_{iCol} \\
\vdots \\
W_{dCol}^T k_{iCol}
\end{bmatrix}
\] (25)

where \(i = 1, \ldots, d\).

2) DOMAIN ADAPTATION BASED ON RKHS SUBSPACE LEARNING

We aim to study the domain adaptive problem. Supposed a source domain dataset and a target domain dataset in the original data
\[
X_s = \{x_1, \ldots, x_{N_s}\} \subseteq \Omega,
\]
\[
X_t = \{x_1, \ldots, x_{N_t}\} \subseteq \Omega,
\]
\[N = N_s + N_t\]

the source domain data \(X_s\) is labeled. Similarly, the label of the unlabeled target domain data \(X_t\) needs to be identified by \(X_s\). The distribution of \(X_s\) and \(X_t\) are different.

This section focuses on the domain adaptive algorithm based on RKHS subspace learning. First, both domain data are transformed from the original space to the RKHS subspace. And then, the distribution of the source domain and the target domain data will converge in RKHS subspace by the subspace learning.

Let
\[
X = X_s \cup X_t = \{x_1, \ldots, x_N\}
= \{x_1^s, \ldots, x_{N_s}^s, x_1^t, \ldots, x_{N_t}^t\} \subseteq \Omega, \quad N = N_s + N_t
\]
Using the RKHS subspace learning framework, have
\[
Y_s = \{y_1^s, \ldots, y_{N_s}^s\} \subseteq \mathbb{R}^d, \quad y_i^s = W_{iCol} k_{iCol},
\]
\[
Y_t = \{y_1^t, \ldots, y_{N_t}^t\} \subseteq \mathbb{R}^d, \quad y_i^t = W_{(N_s+i)Col} k_{(N_s+i)Col},
\]
where the matrix \(W\) is unknown and represents the RKHS subspace. \(Y_s\) and \(Y_t\) are convergent in the distribution of Euclidean space \(\mathbb{R}^d\) by learning \(W\). We measure the convergence of the distribution by Maximum Mean Difference (MMD) criterion:

\[
\left\| \frac{1}{N_s} \sum_{i=1}^{N_s} y_i^s - \frac{1}{N_t} \sum_{i=1}^{N_t} y_i^t \right\|^2
\]
\[
= \left\| W^T \left( \frac{1}{N_s} \sum_{i=1}^{N_s} k_{iCol} - \frac{1}{N_t} \sum_{i=1}^{N_t} k_{(N_s+i)Col} \right) \right\|^2
\]
\[
= \left\| W^T \Psi \right\|^2 = \text{tr} \left( W^T \Psi W \right) \to \min \] (26)

where
\[
\Psi = \frac{1}{N_s} \sum_{i=1}^{N_s} k_{iCol} - \frac{1}{N_t} \sum_{i=1}^{N_t} k_{(N_s+i)Col}
\]

3) DICTIONARY LEARNING IN RKHS SUBSPACES

We hope that the spatial distribution of the same category data in the source and target domain coincides in the subspace. Since the category of the target domain data is unknown, this requirement is difficult to model. We adopt the regular-term of the source domain dictionary and realize that the target domain data is distributed around the source domain data that is almost linearly related to it, indirectly meeting the
requirements. We use the source domain $X_s$ as the dictionary and the target domain as the training sample $X_t$. $X_t$ use the dictionary $X_s$ to linear representation expressed as $\alpha$. In dictionary learning, every data in the target domain can be linearly represented by appropriate original data through learning coding coefficients $\alpha$. And the coding coefficients are sparse, so the L1 regularity is adopted for the coding coefficients. The model of dictionary learning is as follows:

$$DIC^2 (X_s, X_t) = \arg \min_{\alpha} \sum_{j=1}^{N_t} \left| y_j \right|^2 - \sum_{j=1}^{N_t} \alpha_y y_j^2 + \lambda \sum_{j=1}^{N_t} \| \alpha_j \|_1$$

$$= \arg \min_{\alpha} \sum_{j=1}^{N_t} \left| y_j \right|^2 \sum_{j=1}^{N_t} \alpha_y y_j^2 + \lambda \sum_{j=1}^{N_t} \| \alpha_j \|_1$$

$$= arg \min_{\alpha} \sum_{j=1}^{N_t} \sum_{i=1}^{N_t} \left| y_j \right|^2 \sum_{i=1}^{N_t} \alpha_y y_i^2 + \lambda \sum_{j=1}^{N_t} \| \alpha_j \|_1$$

$$= arg \min_{\alpha} \sum_{j=1}^{N_t} \sum_{i=1}^{N_t} \left| y_j \right|^2 \sum_{i=1}^{N_t} \alpha_y y_i^2 + \lambda \sum_{j=1}^{N_t} \| \alpha_j \|_1$$

$$= arg \min_{\alpha} \sum_{j=1}^{N_t} \left| y_j \right|^2 \sum_{i=1}^{N_t} \alpha_y y_i^2 + \lambda \sum_{j=1}^{N_t} \| \alpha_j \|_1$$

where

$$\Phi = \sum_{j} \left( \tilde{K}_{(N_t+j)\text{Col}} - \sum_{i=1}^{N_t} \tilde{K}_{i\text{Col}\alpha_j} \right) \left( \tilde{K}_{(N_t+j)\text{Col}} - \sum_{i=1}^{N_t} \tilde{K}_{i\text{Col}\alpha_j} \right)^T$$

$\Phi$ represents the source domain data, $y^s$ is the target domain, $W$ is the subspace projection matrix. $\alpha$ represents the coding coefficient, the linear representation of the source domain data mentioned above. $\| \alpha \|_1$ controls the sparsity of $\alpha$, and $\| W \|_2$ is used to control the complexity of $W$.

Finally, the objective function of the subspace learning part is as follows:

$$arg \min_{\alpha} \sum_{j=1}^{N_t} \| y_j \|^2 + \lambda \sum_{j=1}^{N_t} \| \alpha_j \|_1$$

$$s.t. W^T KW = I_d$$

$$\Phi = \sum_{j} \left( \tilde{K}_{(N_t+j)\text{Col}} - \sum_{i=1}^{N_t} \tilde{K}_{i\text{Col}\alpha_j} \right) \left( \tilde{K}_{(N_t+j)\text{Col}} - \sum_{i=1}^{N_t} \tilde{K}_{i\text{Col}\alpha_j} \right)^T$$

4) DOMAINT ADAPATION BASED ON SOURCE DICTIONARY REGULARIZED RKHS SUBSPACE LEARNING (SDRRKHS-DA)

Domain adaptive learning based on RKHS subspace learning only overlaps the overall spatial distribution of source domain data and target domain data as much as possible in the subspace. However, we want the spatial distribution in the same category to overlap in the subspace. Comparison to other tasks of domain adaptive learning, mental recognition from EEG seems to be very challenging. Therefore, in this section we will combine the MMD and dictionary learning, and propose a subspace learning algorithm based on the source domain dictionary regularized. The algorithm requires the target domain data to distribute around the source domain with the strongest linear correlation, which indirectly reflects the requirement that the spatial distribution of the source domain and target domain in the same category should be consistent. The data is transformed from the original space to RKHS by the reproducing kernel, and further to a RKHS subspace by the subspace projection. For selecting subspace, MMD minimizes the mean difference to reduce the distribution difference between domains, and source domain dictionary regularization constitutes a new RKHS subspace learning method. The model should learn the relationship between the source domain and the target domain for representing the target domain by the most relevant data in the source domain. Our model is as follows:

$$\arg \min_{\alpha} \sum_{j=1}^{N_t} \| y_j \|^2 + \sum_{j=1}^{N_t} \| \alpha_j \|_1 + \mu \| W \|_2$$

5) SOLUTION TO SDRRKHS-DA

a: SOLUTION TO $W$

To update the subspace projection matrix $W$, we fix the value of the coding coefficient $\alpha$. The problem finally becomes a common problem of solving the subspace projection matrix $W$. The updated objective function of $W$ is

$$\min_{W} W^T \Psi W^T + \mu \| W \|_2$$

$$= \min_{W} W^T \tilde{K} \tilde{K}^T W + \mu \| W \|_2$$

$$= \min_{W} W^T \tilde{K} \tilde{K}^T W + \mu \| W \|_2$$

where, $V = \tilde{K} \tilde{K}^T$. Similarly, $W^T \tilde{K} W = W^T \tilde{K} \tilde{K}^T W = V^T V = I_d$

Then, $\min_{W} W^T \tilde{K} \tilde{K}^T W + \mu \| W \|_2$
Rayleigh entropy can be used to solve the above problem. Take the $d$ eigenvectors corresponding to the first $d$ smallest eigenvalues, which can be used as the subspace projection matrix $W$.

\[ b: \text{SOLUTION TO } \alpha \]

In the process of updating the coding coefficient $\alpha$, we first fix the value of the projection matrix $W$, and the problem becomes the most primitive sparse coding problem. The objective function for updating the coding result is:

\[
\min_{\alpha} \left\| W^T \tilde{K}_{jCol} - \sum_{i=1}^{N_t} W^T \tilde{K}_{iCol} \alpha_{ij} \right\|^2_2 + \lambda \left\| \alpha_j \right\|_1
\]

\[
= \min_{\alpha} \tilde{K}_{jCol}^T W W^T \tilde{K}_{jCol} + \sum_{i=1}^{N_t} \tilde{K}_{jCol}^T W W^T \tilde{K}_{iCol} \alpha_{ij}
\]

\[
+ \sum_{i=1}^{N_t} \alpha_{ij} \tilde{K}_{jCol}^T W W^T \tilde{K}_{iCol} \alpha_{ij}
\]

\[
+ \sum_{i=1}^{N_t} \alpha_{ij} \tilde{K}_{iCol}^T W W^T \tilde{K}_{jCol} \alpha_{ij} + \lambda \left\| \alpha_j \right\|_1
\]  \hspace{1cm} (33)

The above problem is a typical Lasso optimization problem, and the toolbox of SPAMS [40] and CVX [41] can quickly solve this problem.

Algorithm 1 EEG-KLSD Algorithm

\textbf{Input:} source domain data $X_s$, and target domain data $X_t$, kernel $k$ and binary function $k_0$, parameter $\mu$, $\lambda$, $\eta$, $\beta$

\textbf{Output:} the projection matrix $W$

1. According to $K_{stc}^T \Gamma \Gamma^T K_{stc} + 2\beta M$, use conjugate gradient method to optimize $M$;
2. According to $M$ and $k(x, y) = k(x, y) + k_2(x)MK_2(y)$, find the kernel function $\tilde{k}$;
3. Calculation matrix $\tilde{K}$ and $L$, randomly initialize coding coefficients $\alpha$;
4. Calculate matrix $\Phi$ by coding coefficient;
5. Perform eigenvalue decomposition on $\Theta$, and take the $d$ eigenvectors corresponding to the first $d$ smallest eigenvalues to form $W$;
6. Solve coding coefficients $\alpha$ by mexlasso;
7. Iteratively solve steps 5 and 6, and stop iterating until the loss value is less than the set threshold;
8. Get the projection matrix $W$.

\section{Complexity Analysis}

We use $O_1$ and $O_2$ to represent the time complexity and space complexity, where $N_t$ represents the number of samples in the source domain, $N_i$ represents the target domain, $N = N_s + N_t$. In RKHS learning, we first need to construct the kernel matrix, so the time complexity adopted is $O_1(N^2 + N_tN + N_sN)$, the space complexity is $O_2(N^2 + N_tN + N_sN)$, the time complexity of optimizing the kernel function using Riemannian conjugate gradient is $O_1(\frac{3}{2}N^3)$, and the space complexity is $O_2(2N^2 + N_1N)$. If the number of the optimization algorithm iterations is $l_1$, then the total time complexity is $O_1(l_1N^3)$, and the space complexity is $O_2(2l_1N^2 + l_1N)$.

In RKHS subspace learning, for the RKHS subspace learning algorithm based on the source domain dictionary, we need to update $W$ through SVD decomposition. For each SVD, the time complexity is $O_1(l_2N^3)$, and the space complexity is $O_2(N^2)$. Then, the sparse matrix $\alpha$ is updated by MexLasso, and the time complexity is $O_1(l_4(N^4 + N^3 + N_1N))$, and the space complexity is $O_2(l_4(N^2 + N_1N))$.

According to the time complexity and space complexity of RKHS learning and RKHS subspace learning, and the overall time complexity is $O_1(l_2(N^2 + N_tN + N_sN + \frac{4l_1+3l_2}{3}N^3 + l_2(N^4 + N_sN))$, and the space complexity is $O_2((2l_1 + 2l_2 + 1)N_2^2 + (l_2 + 1)N_1N + (l_1 + l_2)N_1N + N_sN)$.

\section{Comparison with Other Related State-of-the-Art Algorithms}

In this chapter, we will introduce five state-of-the-art algorithms. The five algorithms all adopt the subspace learning method to solve the domain adaption. TCA and SSTCA are some of the earliest subspace learning algorithms and achieve good results in domain adaption. IGLDA, TIT, and GSL proposed in recent years, and they all have their advantages in domain adaptation. This chapter mainly describes the theoretical comparison between our algorithm and the comparison algorithms, and the experimental comparison will be elaborated in the next chapter.

\subsection{Comparison to TCA}

The model of TCA [32] is

\[
\min_{W} \left\| \frac{1}{N_t} \sum_{i=1}^{N_t} W^T K_{iCol} - \frac{1}{N_t} \sum_{j=1}^{N_t} W^T K_{iCol} \right\|^2_2 + \mu \left\| W \right\|_2
\]

\[ s.t. W^TkHKW = I_d \]  \hspace{1cm} (34)

TCA adopted MMD as the domain adaptive criterion, map the data by a kernel function to RKHS, and then through the subspace projection matrix $W$ mapped to RKHS subspace. The data in the subspace minimizes the distance of the source domain and target domain data. TCA adopts $W$ for controlling the complexity of $W$. $W^TkHKW = I_d$ maximizes the variance of data after mapping for preserving attributes that are useful for categorizing tasks. The difference between EEG-KLSD and TCA is that EEG-KLSD adopts the two learning framework and the source domain dictionary regularization so that the spatial distribution of the source domain and target domain data in the same category in the subspace is as close as possible. In addition, the TCA subspace constraints are only derived from mathematical formulas,
while our subspace constraints are derived from the RKHS subspace orthonormal basis.

B. COMPARISON TO SSTCA
Based on TCA, SSTCA [32] adds a manifold-regular optimization term, and its model is

\[
\begin{align*}
\min_W & \left\| \frac{1}{N_s} \sum_{i=1}^{N_s} W^T K_{iCol} - \frac{1}{N_t} \sum_{j=1}^{N_t} W^T K_{(N_t+j)Col} \right\|^2 \\
& + \frac{\lambda}{N^2} \sum_{j,l=1}^{N_t} \left\| W^T K_{jCol} - W^T K_{lCol} \right\|^2_P + \mu \| W \|_2^2 \\
\text{s.t.} & \quad W^T KHK_{s}HKW = I
\end{align*}
\]

(35)

\( W^T KHK_{s}HKW = I \) uses HSIC to enhance the correlation between labels and data; The second term in formula (32) is a manifold-regular term that preserves the data-geometric structure. The difference between EEG-KLSD and SSTCA is that EEG-KLSD adopts a two learning framework. In subspace learning, SSTCA uses manifold regularization, and it wants to map all data into the subspace as close as possible. The source domain dictionary regularization combines the linear discrimination and hopes that the spatial distribution of the source domain and the target domain data in the same category in the subspace coincidences as much as possible.

C. COMPARISON TO IGLDA
The IGLDA [33] model is as follows:

\[
\begin{align*}
\min_W & \left\| \frac{1}{N_s} \sum_{i=1}^{N_s} W^T K_{iCol} - \frac{1}{N_t} \sum_{j=1}^{N_t} W^T K_{(N_t+j)Col} \right\|^2 \\
& + \frac{c}{N_t(N_t-1)} \sum_{i=1}^{N_t} \sum_{j=i+1}^{N_t} \left\| W^T K_{iCol} - W^T K_{jCol} \right\|^2 \\
& + \mu \| W \|_2^2 \\
\text{s.t.} & \quad W^T KHKW = I_d
\end{align*}
\]

(36)

where, \( c \) is the number of categories, \( N_t \) is the number of data in each category. From the above model, IGLDA is similar to TCA. They both adopt MMD as the criteria for domain adaptation and \( \mu \| W \|_2^2 \) for controlling the complexity of \( W \). \( W^T KHKW = I_d \) also maximizes the variance of mapped data. The difference between EEG-KLSD and IGLDA is not only in the two learning framework but also in the regular terms. The regular term adopted by IGLDA is similar to the inter-class divergence of source domain data, but it does not require the distance of source domain data of different categories to be as far as possible, which will lead to misjudgment of target domain data labels. The source domain dictionary regularization combines the linear discrimination and hopes that the target domain data can be distributed around the source domain data that is linearly related to it. In general, the data with the strongest linear correlation are the same class, so in theory, source domain dictionary regularization can improve the classification accuracy very well.

D. COMPARISON TO TIT
TIT [34] obtains false labels through multiple experiments to improve the final accuracy. Its model is as follows:

\[
\begin{align*}
\min_W & \left\| \frac{1}{N_t} \sum_{i=1}^{N_t} W^T K_{iCol} - \frac{1}{N_t} \sum_{j=1}^{N_t} W^T K_{(N_t+j)Col} \right\|^2 \\
& + \frac{\lambda}{2} \sum_{j,l=1}^{N_t} \left\| W^T K_{jCol} - W^T K_{lCol} \right\|^2_P + \mu \| W \|_2^2 \\
& + \beta \left\| K_t - W_t W_t^T K_t \right\|^2_F + \mu \| W \|_2^2 \\
\text{s.t.} & \quad W^T KHKW = I_d
\end{align*}
\]

(37)

where \( W = \{W_i, W_t\} \), \( K_t \) represents the target domain data on RKHS, and \( W_t \) represents the dimension-reduction matrix of \( K_t \). As in the above models, TIT also adopts MMD, but this model adopts \( \| W \|_2^2 \) in the construction subspace, which means that \( W \) are as sparse as possible. Based on SSTCA, TIT adds a regularization \( \| K_t - W_t W_t^T K_t \|_F \) that is similar to PCA and makes the projected target domain data as close as possible. If KNN is used as a classifier, this regular term does not play a role. Compared with TIT, EEG-KLSD adopts a two learning framework and maps both source domain and target domain data to the same subspace, so that the target domain data is distributed around the linearly related source domain data, which is better than TIT in theory.

E. COMPARISON TO GSL
The model of GSL [35] is as follows:

\[
\begin{align*}
\min_W & \left\| W_t X - \hat{Y} \circ M \right\|^2_F \\
& + \alpha \| Z \|_s \\
\text{s.t.} & \quad M \geq 0
\end{align*}
\]

(38)

GSL algorithm maps source domain and target domain data to two different subspaces by two different projection matrices and solves the problem that the number of source domain data and target domain data is not equal by \( Z \) so that the projected source domain and target domain data are closer to each other. GSL expects the learning of the source domain subspace \( W_t \) to guide the learning of the target domain subspace \( W_t \), thus minimizing the Bregmans divergence between \( W_t \) and \( W_t \). \( W_t X - \hat{Y} \circ M \) trains a classifier. Unlike EEG-KLSD, GSL is not trained on the RKHS. The reason for the comparison with GSL is that \( W_t X - W_t X \) adopted in GSL is somewhat similar to our source domain dictionary regularizer and also has achieved great results in domain adaptation.

V. EXPERIMENTS
This section performs classification tasks on the BCI Competition IV 2a dataset and compares our algorithm with the TCA, SSTCA, IGLDA, TIT, and GSL algorithms mentioned in the article, and verifies the effectiveness of our algorithm.
A. THE INTRODUCTION OF BCI IV 2a DATASET

The BCI Competition IV 2a dataset is obtained by collecting the EEG signals of 9 healthy subjects. They are performing four different motion-imaging tasks, which are imagining the movements of the left hand, right hand, feet, and tongue. For each subject, two sessions were recorded on two different days, each session contained 288 EEG signals, and 72 trials were performed for each category. In each experiment, the researcher will give a reminder in the form of an arrow pointing to the left, right, down, or up corresponding to one of the four categories to remind the subject to perform the corresponding motion-imaging task. EEG signals are recorded in the sensory-motor area of the subject through 22 electrodes on the device at a sampling rate of 250 Hz. From the appearance of the prompt to the end of the motion image task, the motion imagery lasted for 4 seconds. Starting from the suggestion that the user is prompted to perform the mental task, the time interval of the processed data is limited to between 0.5 seconds and 2.5 seconds. The EEG signal of each experiment is band-pass filtered by a fifth-order Butterworth filter in the 10-30Hz frequency band. The subjects were identified as A01-A09, the first experiment day was E, and the second one was T, e.g. the data of Subject 1 on the first experiment day was identified as A01E.

B. VALIDITY EXPERIMENT

In this section, we will prove the validity of the data and the validity of the two innovations in our algorithm. We will first verify whether the data meets the basic conditions of domain adaptation, and then verify whether the probability distribution of the source domain data and the target domain data after processing by our algorithm is consistent. In terms of algorithm effectiveness, we design the proposed algorithm into three experiments, namely RKHS learning + MMD, MMD + regular term, and RKHS learning + regular term, which are compared with MMD algorithm and baseline respectively. Baseline is an algorithm that uses pca for dimensionality reduction in the original space, and then uses SVM for classification. The purpose of pca dimensionality reduction in baseline is to ensure the consistency of the experiment, because the algorithm and MMD proposed in this paper both use subspace dimensionality reduction.

In terms of data, we divided each subject’s two sessions recorded on two different days into a source domain and a target domain, and used TSNE [42] to visualize the data. We found that the data in the source domain and the target domain meet the basic requirements of domain adaptation. The source domain and the target domain are geometrically separated, which proves that the probability distributions of the source domain and the target domain are not the same but related. The distribution diagram of source domain data and target domain data is shown in Figure 2.

We used A01T to A09T as the source domain, and A01E to A09E as the target domain. A total of 9 experiments were designed. For each experiment, all the source domain data is used as the training set, and the target domain data is used as the test set. We will use RKHS learning+MMD (RL+MMD), MMD+source domain dictionary regularization(MMD+SDR), EEG-KLSD (the proposed method), MMD and baseline for experiments. This experiment uses SVM as the classifier, trains the SVM model according to the low-dimensional representation of the training set data and its labels, and then predicts the low-dimensional representation of the test set. The same experiment is repeated 10 times. Our algorithm parameter settings are: EEG-KLSD: $\mu = 1$, $\lambda = 0.1$, $\eta = 1$, $\beta = 1$. The kernel function uniformly uses the rbf kernel, and the dimension of all data is reduced to 50.

The bolded data in Table 2 is the highest classification accuracy in each experiment. From the table, we can know that RL+MMD and SDR+MMD have higher classification accuracy than baseline and MMD algorithms. The average classification accuracy of RL+MMD algorithm is 11.63% and 3.02% higher than baseline and MMD algorithms, respectively. The SDR+MMD algorithm has improved the classification accuracy more significantly. The SDR+MMD algorithm is 35.53% and 26.91% higher than the baseline and MMD algorithms in average classification accuracy, respectively, which proves that the two innovations we proposed are Effective. In addition, the EEG-KLSD algorithm obtained by combining RL and SDR has a higher classification accuracy than RL+MMD and MMD+SDR, which use one of the innovative points alone. The EEG-KLSD algorithm is better than RL+MMD and MMD+ respectively. The SDR algorithm is 24.9% and 1.02% higher, which also proves that the EEG-KLSD algorithm is effective.

In addition to the classification accuracy, it can also be seen from the data distribution that our algorithm has played a very good role in the consistency of the probability distribution of the source domain data and the target domain data.
The point in Figure 3 is the data saved after processing by our algorithm, and then visualized using the tSNE algorithm. The red dots represent the source domain data, and the cyan dots represent the target domain data. From Figure 2 we can see that the distribution of source domain data and target domain data have overlapped, which is completely different from the situation in Figure 2 where the source domain data and target domain data can be completely separated and have boundaries. This situation shows that the probability distribution of the source domain and the probability distribution of the target domain are almost the same. In this case, using the source domain data to identify the target domain data can get a higher accuracy rate. Therefore, Figure 3 shows that our algorithm is very effective.

**TABLE 2. Classification accuracy rate of validity experiment (%).**

| number | Baseline | MMD | RL+MMD | SDR+MMD | EEG-KLSD |
|--------|----------|-----|--------|---------|----------|
| A01    | 34.72    | 52.95 | 59.03  | 79.17   | 80.56    |
| A02    | 26.04    | 25.69 | 25.69  | 58.33   | 59.38    |
| A03    | 25.35    | 44.82 | 47.69  | 78.47   | 79.17    |
| A04    | 24.65    | 28.81 | 27.53  | 58.68   | 59.03    |
| A05    | 22.92    | 25.66 | 29.51  | 43.40   | 44.10    |
| A06    | 34.03    | 34.37 | 35.07  | 45.49   | 46.88    |
| A07    | 29.51    | 46.98 | 52.08  | 74.65   | 75.21    |
| A08    | 27.08    | 41.32 | 43.40  | 78.23   | 79.51    |
| A09    | 50.35    | 55.56 | 59.38  | 78.13   | 79.86    |
| average| 30.52    | 39.13 | 42.15  | 66.06   | 67.08    |
| variance| 63.90   | 127.73 | 155.38 | 193.81  | 196.90  |

**FIGURE 3.** Distribution of source domain data (A01T to A09T) and target domain data (A01E to A09E) after our algorithm processing.

The C. EEG CLASSIFICATION

We compare our algorithm with five state-of-the-art algorithms to prove that our algorithm. We used A01T to A09T as the source domain, and A01E to A09E as the target domain. There are 9 experiments in total. For each experiment, all the source domain is used as the training set, and the target domain is used as the test set. The EEG-KLSD and the comparison algorithm are used to train on the training set to obtain the low-dimensional representation in the subspace of the training set. The training set and the test set are respectively multiplied by the low-dimensional representation to get the reduced dimensional training and test data. The classifier is SVM in this experiment and its model trains on the low-dimensional representation of the training set data and labels. SVM predicts the low-dimensional representation of the test set. Our algorithm and comparison algorithm parameter settings are as follows: EEG-KLSD: $\mu = 1, \lambda = 0.1, \eta = 1, \beta = 1$; TCA: $\mu = 1$; SSTCA: $\mu = 1, \lambda = 10^{-3}$; IGLDA: $\mu = 1, \lambda = 10^{-7}$; TIT: $\mu = 10^{-5}, \lambda = 10^{-7}, \beta = 1$; GSL: $\lambda = 0.1, \beta = 4 \times 10^9$. The kernel function and binary function of the EEG-KLSD algorithm use the rbf kernel uniformly, and the kernel function in the comparison algorithm also uses the rbf kernel. The dimension of all data is reduced to 50. The same experiment was repeated 10 times.

Number represents different subjects. The bolded data in Table 3 is the highest classification accuracy in each experiment. In this experiment, in addition to the experiments of the algorithm in this article and the five state-of-the-art algorithms, experiments on the baseline and MMD algorithm are also added. The baseline algorithm in this section is the same as the baseline algorithm mentioned in the previous section. From the table 3, we can see that the five state-of-the-art algorithms are significantly improved compared with the baseline algorithm and the MMD algorithm. Compared with the baseline algorithm, the five state-of-the-art algorithms have increased by 22.48%, 22.88%, 23.08%, 28.67%, and 29.26% respectively. Compared with the MMD algorithm, the five state-of-the-art algorithms have increased by 13.87%, 14.27%, 14.47%, 20.06%, and 20.65% respectively. It can be seen that the five state-of-the-art algorithms have a large improvement compared with the ordinary baseline algorithm and the MMD algorithm, but our algorithm also has a good improvement compared with the five state-of-the-art algorithms. This shows the superiority of our algorithm. From the table, we can see that compared with the other five the state-of-the-art algorithms, our algorithm has improved their average accuracy by 14.08%, 13.69%, 13.48%, 7.89%, 7.3%, respectively. The improvement rate reached 12.21% 26.57%, which shows that the improvement effect of classification accuracy is very significant.
TABLE 4. Classification accuracy of different dimensions in A01 (%).

| number | TCA  | SSTCA | IGLDA | TIT  | GSL  | EEG-KLSD |
|--------|------|-------|-------|------|------|----------|
| 10     | 70.56| 73.02 | 69.79 | 74.58| 72.12| 79.17    |
| 20     | 73.47| 75.14 | 73.65 | 76.22| 75.35| 79.51    |
| 30     | 74.69| 78.86 | 75.49 | 77.22| 74.31| 79.86    |
| 50     | 75.31| 75.07 | 76.14 | 76.75| 76.39| 80.21    |
| 70     | 76.63| 74.65 | 75.14 | 78.98| 76.15| 79.51    |
| 90     | 75.00| 75.35 | 76.18 | 76.39| 77.71| 80.56    |
| 120    | 75.56| 75.28 | 76.46 | 77.78| 77.43| 79.66    |
| 150    | 75.66| 75.73 | 75.10 | 76.74| 76.39| 79.93    |

average 74.61 74.89 74.74 76.58 75.73 79.80
variance 3.05 0.59 4.19 0.78 2.88 0.17

TABLE 5. Classification accuracy of different dimensions in A02 (%).

| number | TCA  | SSTCA | IGLDA | TIT  | GSL  | EEG-KLSD |
|--------|------|-------|-------|------|------|----------|
| 10     | 31.11| 30.94 | 30.45 | 48.09| 56.11| 58.33    |
| 20     | 29.13| 31.98 | 28.13 | 48.92| 53.75| 58.68    |
| 30     | 29.48| 30.94 | 29.79 | 49.58| 49.72| 58.33    |
| 50     | 29.31| 30.42 | 28.82 | 51.63| 50.34| 59.38    |
| 70     | 29.31| 28.47 | 29.44 | 50.28| 49.20| 59.03    |
| 90     | 28.78| 28.23 | 28.75 | 51.42| 48.75| 57.99    |
| 120    | 28.16| 28.75 | 28.40 | 50.14| 47.50| 58.68    |
| 150    | 28.19| 27.71 | 27.50 | 52.57| 46.15| 58.33    |

average 29.18 29.68 28.91 50.33 50.19 58.59
variance 0.75 2.17 0.79 1.94 9.32 0.17

in ten dimensions of 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100, for a total of 20 experiments. In the above 36 experiments, we tested and recorded the classification accuracy of 5 comparison algorithms and our algorithm. SVM is the classifier, trains its model according to the low-dimensional representation of the training set data and labels, and then predicts the low-dimensional representation of the test set. Our algorithm and comparison algorithm parameter settings are as follows: EEG-KLSD: \( \mu = 1, \lambda = 0.1, \eta = 1, \beta = 1; \) TCA: \( \mu = 1; \) SSTCA: \( \mu = 1, \lambda = 10^{-5}; \) IGLDA: \( \mu = 1, \lambda = 10^{-7}; \) TIT: \( \mu = 10^{-5}, \lambda = 10^{-7}, \beta = 1; \) GSL: \( \lambda = 0.1, \beta = 4 \times 10^{7}. \) The kernel function and binary function of the EEG-KLSD algorithm use the rbf kernel uniformly, and the kernel function in the comparison algorithm also uses the rbf kernel. The same experiment was repeated 10 times.

The bolded data in Table 4 and Table 5 are the highest classification accuracy in each experiment. It can be seen from Table 4 and Table 5 that our algorithm has higher classification accuracy in different dimensions than TCA, SSTCA, IGLDA, TIT, and GSL algorithms. In the A01 data set, the average classification accuracy of the EEG-KLSD algorithm is 5.19%, 4.91%, 5.06%, 4.22%, 4.07% higher than the five comparison algorithms, respectively, and the increase rates are 6.96%, 6.56%, 6.77%, 4.21%, 5.38%. In the A02 data set, the average classification accuracy of the EEG-KLSD algorithm is 29.41%, 28.92%, 29.68%, 8.26%, and 8.4% higher than the five comparison algorithms, respectively, and the increase rate is 100.77%, 97.43%, 102.67%, 16.42%, 16.74%. It can be seen that the EEG-KLSD algorithm has a very large improvement on some data sets, and the classification accuracy of the comparison algorithm is twice as high.

It can be seen from Figure 4 and Figure 5 that the classification accuracy of the five comparison algorithms when the dimensions are 10 and 20 is lower than the classification accuracy when the dimension is 30 or more. It may be that the dimensionality reduction is too low, which reduces the effect of the regular term of the comparison algorithm. For the TCA algorithm, it requires the maximization of the variance in the process of subspace dimensionality reduction so that the data can be separated, but when the dimensionality is too low, the data loss is too much, which may cause the data to be separated and affect the classification effect. The source domain dictionary regularization used by our algorithm can make good use of the information of the source domain data. Even if the dimensionality reduction dimension is small, the classification accuracy of the algorithm will not be seriously affected. From Figure 4 and Figure 5, we can also see that the classification accuracy of the EEG-KLSD algorithm in each dimension is very stable and does not fluctuate greatly, while the classification accuracy of the comparison algorithm fluctuates relatively large. For some researchers who need to reduce the dimensionality below 30 dimensions, our algorithm is a better choice.

Next, we will continue to explore the influence of the parameter \( \mu \) and \( \lambda \) on the effect of the EEG-KLSD algorithm. This experiment will use A02 as the experimental data, and use its source domain data and target domain data to analyze the influence of the parameters, and the subspace dimension
is set to 50 dimensions. In this experiment, first fix $\eta = 1$, and set $\mu$ and $\lambda$ to five values of $10$, $1$, $0.1$, $0.01$, and $0.001$ respectively, to test the classification accuracy of the EEG-KLSD algorithm under these different $\mu$ and $\lambda$. The parameter $\mu$ controls the complexity of the subspace projection matrix $W$, and the parameter $\lambda$ controls the sparsity of the coding coefficient $\alpha$. Generally speaking, the larger the value of the parameter $\mu$, the lower the complexity of $W$; and the larger the value of the parameter $\lambda$, the sparser the coding coefficient $\alpha$.

The classification effect of the parameters $\mu$ and $\lambda$ on the EEG-KLSD algorithm is shown in Table 6. It can be seen from the table that under different values of $\mu$, the classification effect of EEG-KLSD at $\lambda = 0.1$ is the best. When the value of $\mu$ is fixed, as the value of $\lambda$ decreases from $10$ to $0.001$, the classification accuracy of the EEG-KLSD algorithm first increases and then decreases. Therefore, it can be considered that the coding coefficient $\alpha$ is sparser, the accuracy of algorithm classification may not be higher. The reason for this result is that when the value of $\lambda$ is large, the coding coefficients are too sparse, causing a large amount of effective data to be lost when the source domain dictionary linearly represents the target domain data, and the target domain data cannot be truly represented by the source domain dictionary. When the value of $\lambda$ is small, the coding coefficients are too redundant, and some source domain dictionaries that have nothing to do with the target domain data are also used to represent the target domain data, causing the model to learn wrong information and reducing the classification accuracy. When the value of $\lambda$ is fixed, the parameter $\mu$ decreases from $10$ to $0.001$. From the table, we find that the regularity of the change in classification accuracy caused by the change of $\mu$ value is not particularly obvious. But in general, when the $\mu$ value is between $0.01$ and $1$, the classification accuracy of the EEG-KLSD algorithm is higher, and this range is a good choice for selecting $\mu$ value.

### VI. CONCLUSION

In order to solve the problem of the mismatch between the marginal probability distribution of the source domain and the target domain in EEG mental recognition, we propose an EEG mental recognition algorithm based on RKHS learning and source domain dictionary regularized subspace learning. In this article, our main contributions are as follows: (1) At present, most of the domain adaptation learning only learns the linear subspace of RKHS, and RKHS itself is not learned. In view of the complexity and nonlinearity of EEG mental recognition, this paper proposes an EEG mental recognition algorithm based on two learning. The two learning is RKHS learning and RKHS subspace learning. Compared with single learning, two learning makes the marginal distribution of source domain data and target domain data closer. (2) According to the Moore-Aronszajn theorem, RKHS can be uniquely generated by the kernel function. The existing RKHS is rarely learnable. The reason is that it is difficult to find a kernel function that is learnable and easy to optimize. In RKHS learning, this paper uses a learnable kernel function that we have published, and the kernel function is easy to optimize. The kernel function can be used to generate a learnable RKHS. (3) In RKHS subspace learning, most of the existing methods use the MMD, but it cannot make the spatial distribution of the same category of source and target domain data overlap as much as possible. And the label of the target domain data is unknown. So we propose an RKHS subspace learning algorithm based on source domain dictionary regularization. The algorithm requires that the target domain data distributes around the source domain with the strongest linear correlation based on MMD. It indirectly reflects the requirement that the spatial distribution of the source domain data and target domain data of the same category is as consistent as possible. Sufficient experiments show that the source domain data and target domain data processed by our algorithm overlap geometrically. Moreover, our algorithm far exceeds the classification effect of several state-of-the-art algorithms in terms of classification accuracy, which fully proves the effectiveness of our algorithm.

MMD is a commonly-used domain adaptive criterion. But two zero mean random variables, although their distributions are different, their MMD is zero. Therefore, finding better criteria and the addition of various rules are important research directions of domain adaptation. In our proposed algorithm, a regular term of dictionary learning is exploited for BCI application. Other regularization methods can be tried in theory and practice in the future.

### REFERENCES

[1] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[2] M. Jiang, Z. Huang, L. Qiu, W. Huang, and G. G. Yen, “Transfer learning-based dynamic multiobjective optimization algorithms,” *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 501–514, Aug. 2018.

[3] Y. Xu, X. Fang, J. Wu, X. Li, and D. Zhang, “Discriminative transfer subspace learning via low-rank and sparse representation,” *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 850–863, Feb. 2016.

[4] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, “Adapting visual category-models to new domains,” in *Proc. CVPR*, Sep. 2010, pp. 213–220.

[5] S. Wang, L. Zhang, W. Zuo, and B. Zhang, “Class-specific reconstruction transfer learning for visual recognition across domains,” *IEEE Trans. Image Process.*, vol. 29, pp. 2424–2438, 2020.

[6] L. Duan, I. W. Tsang, D. Xu, and S. J. Maybank, “Domain transfer SVM for video concept detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1375–1381.

[7] M. Harandi, C. Sanderson, C. Shen, and B. Lovell, “Dictionary learning and sparse coding on Grassmann manifolds: An extrinsic solution,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3120–3127.

[8] Y. Li, J. Liu, H. Lu, and S. Ma, “Learning robust face representation with classwise block-diagonal structure,” *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 12, pp. 2051–2062, Dec. 2014.

---

**TABLE 6. Classification accuracy of different dimensions in A02 (%).**

| $\mu$     | $\lambda = 10$ | $\lambda = 1$ | $\lambda = 0.1$ | $\lambda = 0.01$ | $\lambda = 0.001$ |
|-----------|----------------|---------------|-----------------|-----------------|-----------------|
| $\mu = 10$ | 33.19          | 55.63         | 57.25           | 49.76           | 36.67           |
| $\mu = 1$  | 36.70          | 55.38         | 59.38           | 48.30           | 39.58           |
| $\mu = 0.1$| 34.24          | 55.17         | 58.51           | 51.84           | 35.35           |
| $\mu = 0.01$| 33.33         | 55.73         | 57.17           | 51.67           | 37.50           |
| $\mu = 0.001$| 31.28        | 55.10         | 56.81           | 54.10           | 39.03           |
Y. Wang, S. Gao, and X. Gao, “Common spatial pattern method for channel
L. F. Nicolas-Alonso and J. Gomez-Gil, “Brain computer interfaces, a
G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, “Robust recovery
S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, “Domain adaptation via
S. Said, L. Bombrun, Y. Berthoumieu, and J. Manton, “Riemannian Gaus-
P. Zanini, M. Congedo, C. Jutten, S. Said, and Y. Berthoumieu, “Trans-
S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa, “Generalized
S. Chen, L. Han, X. Liu, Z. He, and X. Yang, “Subspace distribution
H. Li, S. J. Pan, S. Wang, and A. C. Kot, “Heterogeneous domain adapta-
s. G. Liu, Z. Lin, and Y. Yu, “Robust subspace segmentation by low-rank
G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, “Robust recovery of
J. H. Huo, D. Liu, D. T. Lee, and S.-F. Chang, “Robust visual domain adaptation with low-rank reconstruction,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., June 2012, pp. 2168–2175.
D. Valeriani, R. Poli, and C. Cinel, “Enhancement of group perception via a collaborative brain–computer interface,” IEEE Trans. Biomed. Eng., vol. 64, no. 6, pp. 1238–1248, Jun. 2017.
Y. Zhou, S. He, Q. Huang, and Y. Li, “A hybrid asynchronous brain-computer interface combining SSVEP and EEG signals,” IEEE Trans. Biomed. Eng., vol. 67, no. 10, pp. 2881–2892, Oct. 2020.
Y. Jiao, T. Zhou, L. Yao, G. Zhou, X. Wang, and Y. Zhang, “Multi-view multi-scale optimization of feature representation for EEG classification improvement,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 28, no. 12, pp. 2589–2597, Dec. 2020, doi: 10.1109/TNSRE.2020.3049084.
Y. Zhang, T. Zhou, W. Wu, H. Xie, H. Zhu, G. Zhou, and A. Cichocki, “Improving EEG decoding via clustering-based multitask feature learning,” IEEE Trans. Neural Netw. Learn. Syst., early access, Feb. 8, 2021, doi: 10.1109/TNNLS.2021.3053576.
S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa, “Generalized domain-adaptive dictionaries,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2013, pp. 361–368.
S. Shekhar, V. M. Patel, H. Van Nguyen, and R. Chellappa, “Coupled projections for adaptation of dictionaries,” IEEE Trans. Image Process., vol. 24, no. 10, pp. 2941–2954, Oct. 2015.
F. Zhu and L. Shao, “Weakly-supervised cross-domain dictionary learning for visual recognition,” Int. J. Comput. Vis., vol. 109, nos. 1–2, pp. 42–59, Aug. 2014.
S. Li, M. Shao, and Y. Fu, “Cross-view projective dictionary learning for person re-identification,” in Proc. IJCAI, Jun. 2015, pp. 2155–2161.
L. F. Nicolas-Alonso and J. Gomez-Gil, “Brain computer interfaces, a review,” Sensors, vol. 12, no. 2, pp. 1211–1279, 2012.
Y. Wang, S. Gao, and X. Gao, “Common spatial pattern method for channel selection in motor imagery based brain-computer interface,” in Proc. IEEE Eng. Med. Biol. 27th Annu. Conf., Jan. 2005, pp. 5392–5395.
T.-E. Kam and S.-W. Lee, “Time-dependent common spatial patterns optimization for EEG signal classification,” in Proc. 1st Asian Conf. Pattern Recognit., Nov. 2011, pp. 643–646.
P. Zanini, M. Congedo, C. Jutten, S. Said, and Y. Berthoumieu, “Transfer learning: A Riemannian geometry framework with applications to brain–computer interfaces,” IEEE Trans. Biomed. Eng., vol. 65, no. 5, pp. 1107–1116, May 2018.
S. Said, L. Bombrun, Y. Berthoumieu, and J. Manton, “Riemannian Gaussian distributions on the space of symmetric positive definite matrices,” 2015, arXiv:1507.01766.
S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, “Domain adaptation via transfer component analysis,” IEEE Trans. Neural Netw., vol. 22, no. 2, pp. 199–210, Feb. 2011.
M. Jiang, W. Huang, Z. Huang, and G. G. Yen, “Integration of global and local metrics for domain adaptation learning via dimensionality reduction,” IEEE Trans. Cybern., vol. 47, no. 1, pp. 38–51, Dec. 2017.

WENJIE LEI received the B.S. degree in electronic information engineering from Nanchang University, Nanchang, China, in 2019. She is currently pursuing the master’s degree with the School of Electronics and Information Technology, Sun Yat-Sen University, Guangzhou, China. Her current research interests include machine learning and domain adaptation.

ZHENMING MA received the B.Sc. degree in radio technology and the M.Sc. degree in electronic and communication system from the South China University of Technology, Guangzhou, China, in 1982 and 1985, respectively, and the Ph.D. degree in pattern recognition and intelligent control from Tsinghua University, Beijing, China, in 1989. He is currently a Professor with the School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China. His current research interest includes machine learning.

SHUYU LIU received the Ph.D. degree from Sun Yat-sen University, China, in 2010. He is currently the Director of the Experimental Teaching Center, School of Pharmacy, Sun Yat-sen University. His research interests include biological image processing, biomedical engineering, and intelligent medicine.

YUANPING LIN received the B.S. degree in electronic information engineering from South China Agricultural University, Guangzhou, China, in 2017. He is currently pursuing the master’s degree with the School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou. His current research interests include machine learning and domain adaptation.

***