

# RFNet: Riemannian Fusion Network for EEG-based Brain-Computer Interfaces

Guangyi Zhang and Ali Etemad, *Senior Member, IEEE*

**Abstract**—This paper presents the novel Riemannian Fusion Network (RFNet), a deep neural architecture for learning spatial and temporal information from Electroencephalogram (EEG) for a number of different EEG-based Brain Computer Interface (BCI) tasks and applications. The spatial information relies on Spatial Covariance Matrices (SCM) of multi-channel EEG, whose space form a Riemannian Manifold due to the Symmetric and Positive Definite structure. We exploit a Riemannian approach to map spatial information onto feature vectors in Euclidean space. The temporal information characterized by features based on differential entropy and logarithm power spectrum density is extracted from different windows through time. Our network then learns the temporal information by employing a deep long short-term memory network with a soft attention mechanism. The output of the attention mechanism is used as the temporal feature vector. To effectively fuse spatial and temporal information, we use an effective fusion strategy, which learns attention weights applied to embedding-specific features for decision making. We evaluate our proposed framework on four public datasets from three popular fields of BCI, notably emotion recognition, vigilance estimation, and motor imagery classification, containing various types of tasks such as binary classification, multi-class classification, and regression. RFNet approaches the state-of-the-art on one dataset (SEED) and outperforms other methods on the other three datasets (SEED-VIG, BCI-IV 2A, and BCI-IV 2B), setting new state-of-the-art values and showing the robustness of our framework in EEG representation learning.

## I. INTRODUCTION

Brain Computer Interfaces (BCI) enable communication between users and computers through learning and interpreting brain activity, for example, brain signals and neuroimaging [1]. Non-invasive technologies such as Electroencephalogram (EEG), functional magnetic resonance imaging, functional near-infrared spectroscopy, and magnetoencephalography, have been widely used for BCI. Among the above-mentioned technologies, EEG is one of the most popular for BCI applications due to reasons such as portability and low cost, high temporal resolution, and the ability to provide real-time monitoring.

EEG-based BCI have been widely used in many application areas. For example, BCI can enable users to move virtual/digital objects on screens via imagining specific movements (e.g., left hand and right hand) [2]. BCI can also help identify users' affective states (e.g., happy, sad, or neutral) through learning neural patterns during watching different types of emotionally charged movies [1]. Moreover, BCI can provide early detection of fatigue or other impairments through real-time monitoring of the brain activity of drivers [3].

Various approaches have been proposed and implemented for BCI through learning the most discriminative task-relevant features from EEG signals. For example, spatial filtering has been one of the most commonly used technique in BCI to explore the features containing optimal variances with respect to different tasks [4]. Statistical models have also been used to investigate linear relationships between EEG features and output labels [5]. Numerous machine learning techniques have been implemented to help model the nonlinear relationships encountered in EEG-based classification tasks [6]. More recently, various deep learning techniques have considerably improved the performance of EEG-based BCI systems [7].

Given the complexity and high dimensionality of EEG, most existing solutions are often unable to learn the nonlinearities observed in high-dimensional multi-channel EEG manifolds and extracted representations. As a result, most existing EEG-related works propose pipelines customized for particular classification or regression tasks. Thus, there is a clear lack of a generalized framework for EEG representation learning that performs robustly for different BCI tasks (e.g., Motor Imagery (MI) classification, emotion recognition, and vigilance estimation).

In general, many BCI solutions rely on Spatial Covariance Matrices (SCM) computed from raw multi-channel EEG. Euclidean metrics, however, are not suitable to be directly applied on SCM given that SCM of raw EEG are Symmetric Positive Definite (SPD), where SPD matrices belong to Riemannian manifold rather than the Euclidean space [8], [9]. The swelling effect occurs during averaging SPD matrices since the determinant of the Euclidean mean can be strictly larger than the determinants of the matrices being averaged [8]. For instance, a computed determinant is used to measure dispersion of the multivariate variables such as multiple channels and various frequency sub-bands of EEG. Therefore, applying Euclidean mean directly on SPD matrices will increase the undesired data variation, thus resulting in poor classification performance [9]. To overcome this challenge, methods using Log-Euclidean metric [10] and affine-invariant Riemannian metric [11] have been proposed and successfully implemented in many areas such as image set classification [12] and diffusion tensor magnetic resonance imaging [13] through endowing the SPD matrices with Riemannian metrics. Furthermore, in BCI applications, EEG often suffer from the linear mixing effect due to volume conduction [14]. To tackle this problem, the affine-invariant Riemannian distance is implemented due to its invariance to linear transforms in EEG [11]. Therefore, affine-invariant Riemannian metrics have very recently been applied on SCM of brain signals and been able to outperform other

G. Zhang and A. Etemad are with the Department of Electrical and Computer Engineering & Ingenuity Labs, Queen's University, Kingston, Canada K7L 3N9 e-mail: guangyi.zhang@queensu.ca and ali.etemad@queensu.ca.

metrics without affine-invariance property in BCI applications such as age prediction [15].

In this paper, to provide a generalized solution for effective EEG representation learning, we exploit both spatial correlation information of EEG channels while preserving the properties in the Riemannian manifold [8] and time-dependency relationships through learning entropy and frequency features extracted from different time windows. In this process, we overcame the following challenges:

- Generally data collected from EEG channel in one brain region are attenuated and also mixed with signals from other brain lobes [16]. A popular approach to dealing with such issues has been to explore the most discriminative spatial features using Common Spatial Pattern (CSP) techniques [4]. However, implementation of CSP may result in overfitting when a large number of channels are available, as well as not being robust to signal outliers [17]. To tackle these problems, we use the Riemannian approach with affine-invariant metric followed by a Multi-Layer Perceptron (MLP) with dropout to improve the spatial resolution. In particular, the proposed solution addresses the aforementioned issues since: *i)* SCM are SPD matrices that belong to Riemannian manifold; *ii)* Riemannian metrics such as distance and mean are robust to noise and data outliers [18]; *iii)* Riemannian distance between two SPD matrices are not effected by the aforementioned mixing effect due to the important affine-invariant property; *iv)* The proposed MLP module with dropout is better equipped to deal with overfitting.
- Usually SCM of raw multi-channel EEG are SPD since the signal in any channel cannot be strictly expressed as the linear combination of others (also called full rank) [15]. However, artefact suppression for EEG pre-processing in the form of projecting EEG data to lower rank sub-spaces (e.g., signal space separation methods) [19] may discard some information about multi-channel EEG. Also, insufficient data may cause the poor estimation of SCM [20]. This, in turn, may result in rank deficiency of EEG, leading the SCM to be Symmetric Positive Semi-Definite (SPSD) [15]. In the past, Wasserstein distance has been used on the SPSPD matrices, but it lacks affine-invariant property [15]. To tackle this problem, we exploit Principle Component Analysis (PCA) in order to project the SCM from the SPSPD to SPD using dimensionality reduction, thus enabling the use of affine-invariant distance in Riemannian geometry [15], while capturing most of the variance.
- The fusion of aforementioned spatial and temporal information has the following challenges: *i)* two different representations of information (Riemannian manifold and Euclidean space) have different geometric structures [9]. Specifically, different data distributions and large variations may occur if we fuse EEG representations with different structures [21]; *ii)* Complementary and/or contradictory information may exist between spatial and temporal representations; *iii)* Each representation contributes differently from task to task. To solve these challenges, we use a fusion strategy to learn the discriminative

features effectively through investigating the weights of spatial and temporal dependency information respectively rather than simply adopt naive concatenation of their higher level features.

We build the solutions above in an end-to-end deep architecture which we name Riemannian Fusion Network (RFNet). To show the robustness of our RFNet, we evaluate the proposed architecture on four public datasets based on the following considerations: *i)* covering different application areas of EEG-based BCI such as emotion classification, vigilance estimation, and MI classification; *ii)* including problems with both binary and multi-class classification (2, 3, or 4 classes); *iii)* the tasks containing both continuous label prediction (regression) as well as classification; and *iv)* the tasks consisting of different distributions and number of EEG channels.

Our contributions can be summarized as follows:

- We propose a *novel framework* RFNet for EEG representation learning based on learning spatial information from the Riemannian manifold and temporal information from the Euclidean space with an effective fusion strategy through learning attention weights applied to different embeddings.
- We test the proposed framework on *three different EEG-related problem domains* namely emotion recognition, motor-imagery classification, and vigilance estimation, using *four widely used public datasets*.
- Our method performs excellently in all the experiments, approaching the state-of-the-art in one dataset and considerably outperforming the best results of existing works in the other three datasets, setting *new state-of-the-art*.

The rest of this paper is organized as follows. In Section II, we provide an overview of related work on EEG-based BCI applications in the three different application areas namely emotion recognition, MI classification, and vigilance estimation. Section III gives a systematic description of the proposed architecture including feature extraction and learning for both spatial correlations and temporal dependencies, as well as the fusion strategy used. In Section IV, we give a description of all the datasets, implementation details, and evaluation protocols. We further discuss the results and perform ablation studies. Section V presents the summary and conclusions of this paper.

## II. RELATED WORK

In this section, we summarize the related work on the problem domains studies in this paper. First, we group and study on emotion recognition and vigilance estimation papers together as many common techniques have been used for these two areas. Next, we provide an overview of the related work on motor-imagery classification.

### A. Emotion Recognition and Vigilance Estimation

Recently, numerous EEG-based solutions have been proposed for emotion recognition. Pipelines usually consist of feature extraction followed by a classification or regression network [6]. In these approaches, a critical step is often the selection of powerful features from noisy raw EEG signals due

to the non-linear and non-stationary nature of EEG [22]. As an example, differential Entropy (DE) has been recently reported as an effective and robust feature for emotion classification and vigilance regression models [1], [3]. Successive to feature extraction, various types of algorithms have been successfully exploited for the classification/regression tasks. For instance, Group Sparse Canonical Correlation Analysis (GSCCA) was proposed to model the *linear* relationship between extracted features (including DE) and output labels [5]. To investigate the non-linearities in the aforementioned extracted features, several classical machine learning methods such as k-Nearest Neighbor (kNN) [1], Linear Regression (LR) [1], Graph Regularized Sparse Linear Regression (GRSLR) [23], Support Vector Machine (SVM) [1], and Random Forest (RF) [24] have been used for EEG-based emotion classification. Support Vector Regression (SVR) was employed in [3] to predict continuous values for a regression formulation of the problem. To better learn the extracted features, feed-forward Artificial Neural Network (ANN) such as Graph regularized Extreme Learning Machine (GELM) [25] has been used to improve the performance.

In order to explore the most discriminative and task-relevant features, deep learning frameworks were applied. For instance, Deep Belief Network (DBN) was employed to extract high-level representations through deep hidden layers [1]. Double-Layered Neural Network with Subnetwork Nodes (DNNSN) was adopted for predicting vigilance labels [26]. Convolutional Neural Network (CNN) along with capsule attention [27] was adopted to learn spatiotemporal EEG information. Spatial-Temporal Recurrent Neural Network (STRNN) [28] and Long Short-Term Memory (LSTM) network [29] have been employed to learn the temporal information embedded in the EEG time-series. Domain Adaptation Network (DAN) [30] was recently exploited to achieve better performance with utilizing prior knowledge of data distribution in the target domain. Bi-hemispheres Domain Adversarial Neural Network (BiDANN) was proposed to minimize the domain shift between training and testing data through a discriminator [31]. Bi-Hemispheric Discrepancy Model (BiHDM) was proposed to improve the performance based on the architecture of an RNN and DAN through learning domain-invariant features from two brain hemispheres [24]. A similar approach, Regional to Global Brain-Spatial-Temporal Neural Network (R2G-STNN), explored spatial-temporal features through Bidirectional LSTM (BiLSTM) and decreased the domain-shift through training a discriminator [32]. Recently, LSTM has also been used to explore Variational Pathway Reasoning (VPR) [33], and has achieved state-of-the-art performance by firstly employing the RNN network to explore the between-electrode dependencies, thus encoding pathways generated from random walk. Then, it chose salient pathways with the most important pair-wise connections via scaling factors as well as pseudo-pathways. Graph Neural Networks (GNN) such as Dynamical Graph Convolutional Neural Networks (DGCNN) [34], and Regularized Graph Neural Networks (RGNN) [35] have recently been utilized to explore the topological structure of EEG electrodes as well as inter-channel relationships by learning graph connections, approaching state-of-the-art results.

## B. EEG Motor Imagery Classification

Many works on EEG-based MI research rely on CSP as the spatial filtering method for classification [4]. Filter Bank Common Spatial Pattern (FBCSP) which decomposes EEG to few sub-frequency bands before the use of CSP has also been frequently used for feature extraction [4]. For example, in binary classification problems (e.g., left hand Vs. right hand), CSP filters maximize the variance of EEG trials from the left-hand class while minimizing the variance of the EEG trial from the right-hand class, through applying simultaneous diagonalization on two covariance matrices from both classes whose eigenvalues are summed to one [4]. CSP filters have also been used in multi-class MI tasks, mainly through one-versus-rest and one-versus-one strategies [36]–[38]. Several classifiers such as SVM, Naïve Bayes (NB), RF, and Linear discriminant analysis (LDA) have been reported with considerable results while using CSP or FBCSP as the feature extraction method [36]–[39]. For multi-class classification, Extended Sequential Adaptive Fuzzy Inference System (ESAFIS) proposed in [40] shows better results on learning CSP features in comparison to other classifiers such as SVM. To better explore the energy features through CSP, deep learning techniques such as CNN with average pooling and the parallel combination of MLP and CNN has been used and achieved better results compared to SVM [41].

Lastly, end-to-end deep learning approaches have recently been adopted in MI classification. CNN-based architectures such as EEGNet have been directly implemented on raw EEG data [42]. In another direct approach, Capsule Networks (CapsNet) have been applied on spectrogram of EEG data, achieving considerable results [43].

## III. PROPOSED ARCHITECTURE

In the following context, the notations used are described as follow: ‘ $a$ ’ represents a scalar, ‘ $\mathbf{a}$ ’ represents a vector, ‘ $\mathbf{A}$ ’ represents a matrix, ‘ $\mathcal{A}$ ’ represents a differentiable manifold.

### A. Solution Overview

We design a novel architecture RFNet for learning spatio-temporal EEG representations. Initially, we apply a filter bank on EEG. In order to learn the spatial information, we first compute SCM of multiple frequency sub-bands. Then, to tackle the possible rank deficiency caused by artefact suppression, we employ PCA to ensure SCM are in the space of SPD, thus enabling the use of affine-invariant Riemannian distance. Next, we apply the Riemannian distance on the SCM and compute the Riemannian mean. Following, we map the spatial information of SCM in Riemannian manifold to the feature vectors in Euclidean space via tangent space learning using the Riemannian mean as the reference. Lastly, we use MLP to embed the spatial information from the feature vectors.

In order to obtain temporal information in Euclidean space, we employ a three-layer LSTM network with attention to learn the temporal dependencies of entropy and frequency features extracted from the same EEG frequency sub-bands as those used in the Riemannian pipeline. Next, we feed forward the

temporal information from the attention mechanism to a Fully Connected (FC) layer to obtain latent representations.

Next, to learn the mutual and selective information embedded in the latent spatial and temporal representations in Euclidean space, we exploit a fusion strategy to obtain a final embedding suitable for various classification or regression tasks.

### B. Data Pre-processing

To keep consistent with the related work using the same datasets, EEG sampling rates were downsampled from  $1000\text{Hz}$  to  $200\text{Hz}$  for emotion and vigilance datasets while being kept unchanged at  $250\text{Hz}$  for both MI datasets. For each of the four datasets, EEG were band-pass filtered between  $0.5\text{--}70\text{Hz}$  to lower artifacts. Then, a notch filter at  $50\text{Hz}$  was applied to reduce power line noise. Signal amplitudes were re-scaling to the range of  $[-1, 1]$  through min-max normalization so that the data discrepancy across different recording sessions was decreased for each subject.

### C. Temporal Feature Processing

1) *Feature Extraction*: Two types of features namely logarithm Power Spectrum Density (PSD) and DE are extracted. PSD is defined in Eq. 1 and DE of EEG time-series  $X$  with a Gaussian distribution is shown in Eq. 2 respectively. To avoid spectral leakage, frequency domain features are extracted through Short-Time Fourier Transform (STFT) from 1-second Hanning windows overlapping by 50%, offering  $L$  windows ( $L = \lfloor 2 \times T - 1 \rfloor$ , where  $T$  is the length of each EEG segment) for feature extraction.

$$S_{xx}(\omega) = \lim_{T \rightarrow \infty} E[|\hat{X}(\omega)|^2]. \quad (1)$$

$$DE = \frac{1}{2} \log(2\pi e \sigma^2), \quad X \sim N(\mu, \sigma^2). \quad (2)$$

2) *LSTM Network with Attention*: LSTM is a type of RNN that enables the learning of both long and short-term dependencies from sequential data (e.g., text, audio, and bio-signals) while addressing gradient exploding and vanishing problems [44]. LSTM networks have been recently successfully implemented on EEG signals in BCI tasks and achieved notable results [22], [29]. Similarly, in our experiments, following feature extraction, concatenated task-relevant EEG features (DE and logarithm PSD) from different frequency sub-bands (total number of  $H$ ) in different windows are fed into  $L$  corresponding LSTM cells (also called time-steps). Then, information in different LSTM cells ( $s_i$ ) are learned through deciding which part to remember or forget through weights updated during network training [22]. As shown in Eq. 3,  $h_i$  is the generated output of the hidden states at each time-step  $i$  which are passed forward to the the LSTM cell of the next LSTM layer for higher level feature representation learning.

$$h_i = \text{LSTM}(s_i), i \in [1, L], \quad (3)$$

To improve the capability of handling temporal information, deep LSTM architectures followed by soft attention or capsule attention mechanisms have been lately implemented showing great performance on different EEG-related classification

or regression tasks [22], [27]. Compared to a conventional LSTM network that only considers the last hidden state  $h_L$  as the network output, a soft attention mechanism evaluates the importance of all output information ( $\{h_i\}_{i=1}^L$ ) from the last LSTM layer by assigning trainable attention weights  $\alpha_i$  applied on each  $h_i$  as shown in Eq. 4 and 5. Thus, more task-relevant information can be obtained by focusing on certain time-steps through optimizing attention weights. The equations are presented as follows:

$$\mathbf{u}_i = \tanh(\mathbf{W}_s \mathbf{h}_i + \mathbf{b}_s), \quad (4)$$

$$\alpha_i = \frac{\exp(\mathbf{u}_i)}{\sum_j \exp(\mathbf{u}_j)}, \quad (5)$$

$$\mathbf{v} = \sum_i \alpha_i \mathbf{h}_i, \quad (6)$$

where vector  $\mathbf{v}$  is the output of the LSTM network with attention, and  $\mathbf{W}_s$  and  $\mathbf{b}_s$  are the trainable parameters. Accordingly, in our architecture, we employ a soft attention mechanism following a three-layer LSTM network to help focus on the most discrepant higher level features in different time-steps.

### D. Spatial Feature Processing

**Background**: As mentioned earlier in the Introduction, SCM of raw EEG are SPD matrices in Riemannian manifold [15]. Riemannian geometry is employed to better learn and manipulate the SPD matrices, in order to capture spatial information. Recent studies show that Riemannian approach achieves better performance than CSP approaches using the same classifier in BCI applications [45]. Other Riemannian approaches using Minimum Distance to Riemannian Mean (MDRM) and Tangent Space LDA (TSLDA) as classifiers consistently outperformed CSP methods in different EEG classification tasks [46]. Local Isometric Embedding (LIE) was proposed based on using tangent space to better learn the features through dimensionality reduction, compared with MDRM classifier and tangent space followed by an SVM classifier [47]. A very recent study claimed that artefact-suppression reduced the robustness of Riemannian-based approaches [15]. Next, we briefly introduce Riemannian geometry.

**Riemannian Geometry**: Let  $\mathcal{M}$  be a differentiable manifold with  $G$  dimensions. As shown in Figure 1,  $\mathbf{T}_C \mathcal{M}$  denotes the tangent space (also called derivative) of  $\mathcal{M}$  at  $\mathbf{C} \in \mathcal{M}$ .

The inner product of two tangent vectors ( $\mathbf{T}_1, \mathbf{T}_2 \in \mathbf{T}_C \mathcal{M}$ ) is defined as [48]:

$$\langle \mathbf{T}_1, \mathbf{T}_2 \rangle_C = \text{Tr}(\mathbf{T}_1 \mathbf{C}^{-1} \mathbf{T}_2 \mathbf{C}^{-1}), \quad (7)$$

where  $\text{Tr}(\cdot)$  is a trace operator. Also, the inner product introduces the norm of a tangent vector  $\mathbf{T}$  as [15]:

$$\|\mathbf{T}\|_C = [\langle \mathbf{T}, \mathbf{T} \rangle_C]^{1/2} = [\text{Tr}(\mathbf{T} \mathbf{C}^{-1} \mathbf{T} \mathbf{C}^{-1})]^{1/2}. \quad (8)$$

Logarithm mapping (Log) in Eq. 9 helps project  $\mathbf{C}'$  from  $\mathcal{M}$  to  $\mathbf{T}'$  in  $\mathbf{T}_C \mathcal{M}$ . Meanwhile, Exponential mapping (Exp) in Eq. 10 is introduced to project  $\mathbf{T}'$  back to  $\mathbf{C}'$  as shown in following [45]:

$$\mathbf{T}' = \text{Log}_C(\mathbf{C}') = \mathbf{C}^{1/2} \log(\mathbf{C}^{-1/2} \mathbf{C}' \mathbf{C}^{-1/2}) \mathbf{C}^{1/2}, \quad (9)$$

$$\mathbf{C}' = \text{Exp}_{\mathcal{C}}(\mathbf{T}') = \mathbf{C}^{1/2} \exp(\mathbf{C}^{-1/2} \mathbf{T}' \mathbf{C}^{-1/2}) \mathbf{C}^{1/2}, \quad (10)$$

where  $\mathbf{C}, \mathbf{C}' \in \mathcal{M}$ ,  $\mathbf{T}' \in \mathbf{T}_{\mathcal{C}}\mathcal{M}$ ,  $\log(\cdot)$ ,  $\exp(\cdot)$  are logarithm and exponential operations applied on a matrix.

Riemannian distance (also called geodesic distance) is a very important metric representing the distance of the shortest path between  $\mathbf{C}$  and  $\mathbf{C}'$  (shown as the curve in Figure 1) on manifold  $\mathcal{M}$ . The geodesic distance ( $\delta_R$ ) is equivalent to the length of its tangent vector [48], [49], expressed as follows:

$$\delta_R(\mathbf{C}, \mathbf{C}') = \|\text{Log}_{\mathcal{C}}(\mathbf{C}')\|_{\mathcal{C}} = \|\mathbf{T}'\|_{\mathcal{C}}. \quad (11)$$

In the context of this work, we denote  $S_N = \{\mathbf{M} \in \mathbb{R}^{N \times N}, \mathbf{M}^T = \mathbf{M}, \mathbf{x}^T \mathbf{M} \mathbf{x} \geq 0, \forall \mathbf{x} \in \mathbb{R}^N \setminus \mathbf{0}\}$  as the space of SPSPD matrices. Similarly  $S_N^+ = \{\mathbf{M} \in \mathbb{R}^{N \times N}, \mathbf{M}^T = \mathbf{M}, \mathbf{x}^T \mathbf{M} \mathbf{x} > 0, \forall \mathbf{x} \in \mathbb{R}^N \setminus \mathbf{0}\}$  is defined as the space of SPD matrices,  $S_R = \{\mathbf{M} \in S_N, \text{rank}(\mathbf{M}) = R, R < N\}$  is the space of SPSPD matrices, where  $\text{rank}(\mathbf{M})$  is the rank of a matrix, and  $S_R^+ = \{\mathbf{M} \in \mathbb{R}^{R \times R}, \mathbf{M}^T = \mathbf{M}, \mathbf{x}^T \mathbf{M} \mathbf{x} > 0, \forall \mathbf{x} \in \mathbb{R}^R \setminus \mathbf{0}\}$  is the subspace of SPD matrices with full rank  $R$ .

**Our Method:** To learn spatial information embedded in multi-channel EEG, we compute SCM on filtered signals in each frequency sub-band. Suppose  $\{\mathbf{X}_i\}_{i=1}^P \in \mathbb{R}^{N \times T}$ , where  $i$  denotes  $i^{\text{th}}$  segment,  $P$  is the EEG segment number,  $N$  is the EEG channel number, and  $T$  represents the number of data samples per segment. The SCM can be estimated as  $\mathbf{C}_i = \frac{1}{(T-1)} \mathbf{X}_i \mathbf{X}_i^T$ , which may be in  $S_R$  after artefact suppression. PCA is then employed to project SCM from  $S_R$  to  $S_R^+$  in order to enable the affine-invariant property during geodesic distance calculation. At last, we obtain the spatial information (geodesic distance) of SCM in the Riemannian manifold as feature vectors in Euclidean space via tangent space learning based on the chosen reference matrix as described in the following text.

1) *Dimensionality Reduction on  $\{\mathbf{C}_i\}_{i=1}^P$  in  $S_R$ :* Although  $\mathbf{C}_i$  of raw EEG  $\mathbf{X}_i$  are in  $S_N^+$ , artifact suppression may destroy some important information [19] that could result in the rank deficiency of EEG data [15], leading for  $\mathbf{C}_i$  to belong to  $S_R$ . Wasserstein distance has been applied on  $\mathbf{C}_i$  in  $S_R$  [15]. However, it lacks affine-invariance in dealing with the linear mixing effect in multi-channel EEG recordings [15]. Therefore, we perform dimensionality reduction to project  $\mathbf{C}_i$  to  $S_R^+$ , enabling the use of affine-invariant Riemannian distance. First, we use PCA to estimate spatial filter  $\mathbf{W}$  [15]. Specifically, we sort the eigenvector matrix  $\mathbf{V}$  based on descending eigenvalues, and choose  $\mathbf{V}$  containing only top  $R$  eigenvalues of the averaged  $\{\mathbf{C}_i\}_{i=1}^P$  (Eq. 14) as the spatial filter  $\mathbf{W}$ , thus maximizing the variance [15]. Then, we apply the spatial filter  $\mathbf{W} \in \mathbb{R}^{N \times R}$  on signal  $\mathbf{X}_i \in \mathbb{R}^{N \times T}$ , leading to  $\mathbf{X}'_i = \mathbf{W}^T \mathbf{X}_i \in \mathbb{R}^{R \times T}$ . Accordingly, the  $\mathbf{C}_i$  of  $\mathbf{X}'_i$  is expressed as  $\mathbf{W}^T \mathbf{C}_i \mathbf{W} \in \mathbb{R}^{R \times R}$  in  $S_R^+$ .

2) *Affine-invariant Distance of  $\{\mathbf{C}_i\}_{i=1}^P$  in  $S_R^+$ :* Since  $\mathbf{C}_i$  is in  $S_R^+$  after dimensionality reduction, we apply Riemannian distance on  $\mathbf{C}_i$  where the affine-invariant property of distance is illustrated as following:

$$\delta_R(\mathbf{C}, \mathbf{C}_i) = \delta_R(\mathbf{W}^T \mathbf{C} \mathbf{W}, \mathbf{W}^T \mathbf{C}_i \mathbf{W}), \quad (12)$$

where  $\mathbf{C}, \mathbf{C}_i, \mathbf{W} \in \mathbb{R}^{R \times R}$ ,  $\mathbf{W} \mathbf{W}^{-1} = \mathbf{I}$ , and  $\mathbf{W}^T \mathbf{X}'_i$  is the linear transform of the EEG [15], which was proven in [49].

---

**Algorithm 1** Riemannian Mean Algorithm
 

---

```

1: procedure ESTIMATION( $\mathbf{C}_{\text{ref}}$ )
2:    $\mathbf{C}_{\text{ref}}$  Initialization: Eq. 14
3:   repeat
4:      $\mathbf{J} = \frac{1}{P} \sum_{i=1}^P \text{Log}_{\mathbf{C}_{\text{ref}}}(\mathbf{C}_i)$ 
5:      $\mathbf{C}_{\text{ref}} = \text{Exp}_{\mathbf{C}_{\text{ref}}}(\mathbf{J})$ 
6:   until  $\|\mathbf{J}\|_F < \epsilon$ 
7:   return  $\mathbf{C}_{\text{ref}}$ 
8: end procedure
```

---

The affine-invariance property (Eq. 12) enables the geodesic distance between  $\mathbf{C}$  and  $\mathbf{C}_i$  in  $\mathcal{M}$  to be invariant when linear transforms are applied to EEG (e.g., linear mixing effect). Unlike many approaches, we do not apply Log-Euclidean distance on  $\mathbf{C}_i$  in  $S_R^+$  due to the lack of the affine-invariance property [9], [18]. Next, instead of conventional Riemannian approach that apply geodesic distances directly for classification (e.g., MDRM) [50], we preserve the geodesic information ( $\|\mathbf{T}_i\|_{\mathcal{C}}$ ) of  $\{\mathbf{C}_i\}_{i=1}^P$  as feature vectors in Euclidean space (Eq. 11) for further higher level feature learning. To achieve this, we first carefully select the reference matrix so that the spatial information in Riemannian manifold represented by geodesic distance between  $\{\mathbf{C}_i\}_{i=1}^P$  and the reference matrix in  $S_R^+$  can be projected onto the same tangent space, in order to best capture geodesic information of  $\{\mathbf{C}_i\}_{i=1}^P$  in  $\mathcal{M}$ .

3) *Choice of Reference for  $\{\mathbf{C}_i\}_{i=1}^P$  in  $S_R^+$ :* We denote  $\mathbf{C}_{\text{ref}} \in \mathcal{M}$  as the reference matrix for  $\{\mathbf{C}_i\}_{i=1}^P$  during tangent space ( $\mathbf{T}_{\mathbf{C}_{\text{ref}}}\mathcal{M}$ ) learning. In the recent studies, the approaches using Riemannian mean ( $\bar{\mathbf{C}}_R$ ) as the reference ( $\mathbf{C}_{\text{ref}}$ ) during the tangent space learning have outperformed approaches that used other references such as Identity matrix ( $\mathbf{I}$ ) and Euclidean mean ( $\bar{\mathbf{C}}_E$ ) [45], [50]. The Euclidean distance, Euclidean mean, and Riemannian mean equations are presented as following:

$$\delta_E(\mathbf{C}, \mathbf{C}_i) = \|\mathbf{C} - \mathbf{C}_i\|_F, \quad (13)$$

where  $\|\cdot\|_F$  is the Frobenius norm of a matrix.

$$\bar{\mathbf{C}}_E = \arg \min_{\mathbf{C}} \left( \sum_{i=1}^P \delta_E^2(\mathbf{C}, \mathbf{C}_i) \right) = \frac{1}{P} \sum_{i=1}^P \mathbf{C}_i, \quad (14)$$

$$\bar{\mathbf{C}}_R = \arg \min_{\mathbf{C}} \left( \sum_{i=1}^P \delta_R^2(\mathbf{C}, \mathbf{C}_i) \right). \quad (15)$$

Accordingly, we employ the Riemannian mean as  $\mathbf{C}_{\text{ref}}$ . Since there are no closed-form solutions for computing Riemannian mean, we implement the gradient descent algorithm (Algorithm 1) presented in [51] for its efficient computation [50]. The algorithm implements an iterative procedure to approximate Riemannian mean through minimizing the arithmetic mean of the tangent vectors  $\mathbf{J}$ . In the first step, we initialize  $\mathbf{C}_{\text{ref}}$  using arithmetic mean. Then we use Logarithm mapping to project  $\mathbf{C}_i$  to the tangent space  $\mathbf{T}_{\mathbf{C}_{\text{ref}}}\mathcal{M}$  and compute  $\mathbf{J}$ . Next, we project  $\mathbf{J}$  back to  $\mathcal{M}$  to update  $\mathbf{C}_{\text{ref}}$ . The algorithm terminates if either Frobenius norm of  $\mathbf{J}$  is less than the tolerance value ( $\epsilon = 10^{-9}$ ) or the algorithm reaches maximum iteration of 50 times. Lastly, we use Riemannian mean as  $\mathbf{C}_{\text{ref}}$  to project geodesic information of  $\{\mathbf{C}_i\}_{i=1}^P$  onto the same tangent space [48].

4) *Tangent Space Learning for  $\{\mathbf{C}_i\}_{i=1}^P$  in  $S_R^+$* : As mentioned in the previous section, we obtain the geodesic information ( $\|\mathbf{T}_i\|_{\mathbf{C}_{\text{ref}}}$ ) between  $\mathbf{C}_i$  and  $\mathbf{C}_{\text{ref}}$  in  $S_R^+$  based on Logarithm mapping (Eq. 9) using the estimated Riemannian mean (Algorithm 1) as  $\mathbf{C}_{\text{ref}}$ . Then, in order to obtain the feature vectors containing geodesic information in  $\mathcal{M}$  for classification or regression purposes, we require a mapping  $\phi_{\mathbf{C}_{\text{ref}}}: \mathbf{T}_{\mathbf{C}_{\text{ref}}}\mathcal{M} \rightarrow \mathbb{R}^{R \times (R+1)/2}$ , such that  $\forall \mathbf{T}_i \in \mathbf{T}_{\mathbf{C}_{\text{ref}}}\mathcal{M}$ ,  $\|\mathbf{T}_i\|_{\mathbf{C}_{\text{ref}}} = \|\phi_{\mathbf{C}_{\text{ref}}}(\mathbf{T}_i)\|_2$  [15]. From Eq. 8 and 9, we have:

$$\begin{aligned} \|\mathbf{T}_i\|_{\mathbf{C}_{\text{ref}}} &= \left[ \text{Tr} [\text{Log}_{\mathbf{C}_{\text{ref}}}(\mathbf{C}_i) \mathbf{C}_{\text{ref}}^{-1} \text{Log}_{\mathbf{C}_{\text{ref}}}(\mathbf{C}_i) \mathbf{C}_{\text{ref}}^{-1}] \right]^{1/2} \\ &= \left[ \text{Tr} [\log(\mathbf{C}_{\text{ref}}^{-1/2} \mathbf{C}_i \mathbf{C}_{\text{ref}}^{-1/2}) \log(\mathbf{C}_{\text{ref}}^{-1/2} \mathbf{C}_i \mathbf{C}_{\text{ref}}^{-1/2})] \right]^{1/2} \quad (16) \\ &= \|\mathbf{S}_i\|_F = \|\text{Vect}(\mathbf{S}_i)\|_2, \end{aligned}$$

where  $\text{Vect}(\cdot)$  is the vectorization operator and  $\mathbf{S}_i = \log(\mathbf{C}_{\text{ref}}^{-1/2} \mathbf{C}_i \mathbf{C}_{\text{ref}}^{-1/2})$ . Therefore, from Eq. 16, we have  $\phi_{\mathbf{C}_{\text{ref}}}(\mathbf{T}_i) = \text{Vect}(\mathbf{S}_i)$ .

5) *Vectorization*: To further present the geodesic information of  $\mathbf{C}_i$  in  $\mathcal{M}$  as spatial feature vectors, we denote the spatial information mapping as  $\Phi_{\mathbf{C}_{\text{ref}}}: \mathcal{M} \rightarrow \mathbb{R}^{R \times (R+1)/2}$ , such that  $\forall \mathbf{C}_i \in \mathcal{M}$ ,  $\Phi_{\mathbf{C}_{\text{ref}}}(\mathbf{C}_i) = \phi_{\mathbf{C}_{\text{ref}}}(\text{Log}_{\mathbf{C}_{\text{ref}}}(\mathbf{C}_i))$ . From Eq. 11 and 16, we obtain  $\delta_R(\mathbf{C}_{\text{ref}}, \mathbf{C}_i) = \|\Phi_{\mathbf{C}_{\text{ref}}}(\mathbf{C}_i)\|_2 = \|\text{Vect}(\mathbf{S}_i)\|_2$ . Furthermore, if  $\{\mathbf{C}_i\}_{i=1}^P$  are in a small region on  $\mathcal{M}$  as stated in [15], [50], we have:

$$\delta_R(\mathbf{C}_i, \mathbf{C}_j) \approx \|\Phi_{\mathbf{C}_{\text{ref}}}(\mathbf{C}_i) - \Phi_{\mathbf{C}_{\text{ref}}}(\mathbf{C}_j)\|_2, i \neq j, \forall i, j \in [1, P], \quad (17)$$

where  $\mathbf{C}_{\text{ref}}$  is the Riemannian mean of  $\{\mathbf{C}_i\}_{i=1}^P$ . Eq. 17 demonstrates that the geodesic distance between  $\mathbf{C}_i$  and  $\mathbf{C}_j$  can be approximated by the geodesic distance between  $\{\mathbf{C}_i\}_{i=1}^P$  and Riemannian mean. Therefore, the geodesic information obtained through the tangent space learning using Riemannian mean as reference are able to represent geodesic information for  $\{\mathbf{C}_i\}_{i=1}^P$ . Important information of  $\Phi_{\mathbf{C}_{\text{ref}}}(\mathbf{C}_i)$  are completely determined by upper triangular components of  $\mathbf{S}_i$  since it is symmetric. Thus, we use the half-vectorization  $\text{Upper}(\mathbf{S}_i)$  instead of full-vectorization  $\text{Vect}(\mathbf{S}_i)$  to represent  $\Phi_{\mathbf{C}_{\text{ref}}}(\mathbf{C}_i)$ . As suggested in [45], we apply coefficient of  $\sqrt{2}$  on off-diagonal elements, in order to maintain equality brought by norms  $\|\mathbf{S}_i\|_F = \|\text{Upper}(\mathbf{S}_i)\|_2$  (also as in Eq. 16), where  $\text{Upper}(\mathbf{S}_i) = [\mathbf{S}_{i,1,1}, \dots, \sqrt{2}\mathbf{S}_{i,1,R}; \mathbf{S}_{i,2,2}, \dots, \sqrt{2}\mathbf{S}_{i,2,R}; \dots; \mathbf{S}_{i,R,R}] \in \mathbb{R}^{R(R+1)/2}$ . We implement an MLP block to learn the features vectors  $\text{Upper}(\mathbf{S}_i)$  concatenated from different frequency sub-bands, as shown in Figure 2. To this end, we obtain the spatial information from  $\mathbf{C}_i$  in Riemannian manifold as feature vectors  $\Phi_{\mathbf{C}_{\text{ref}}}(\mathbf{C}_i)$  in Euclidean space through establishing an information vectorization mapping to preserve local structures such as geodesic information of  $\{\mathbf{C}_i\}_{i=1}^P$  in the Riemannian manifold.

#### E. Fusion Strategy

The strategy for the fusion of spatial and temporal information plays an essential role in dealing with multimodal or multi-learning approaches of one modality in order to perform classification/regression. Attention mechanisms have been successfully implemented for refining fusion weights

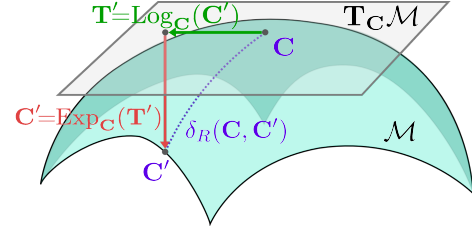


Fig. 1. The Concept of Riemannian Manifold.

applied to different modalities. For instance, for EEG and Electrooculogram (EOG) representation learning, a CapsNet was used in [27] as an attention mechanism for fusion.

In the context of this problem, various tasks rely differently on spatial and temporal information. Therefore, a fusion strategy presented as Figure 4 is adopted. This strategy is inspired by [52] where a hybrid attention-based multimodal architecture was proposed to learn acoustic and textual features and achieved the state-of-the-art performance on several spoken language classification tasks [52]. In our architecture, we first use encoders to learn embedding-specific features. Then we employ soft attention to learn the weight ( $\alpha$ ) applied on each embedding-specific feature. Next, we compute the new weighted embedding by multiplying the weight score with the original individual learning embedding. The weighted score of  $(1 + \alpha)$  is adopted to apply the learned weight in order to maintain the original characteristic [52]. Finally, we perform decision-level fusion on the concatenation of the two new embeddings using an FC layer equipped with different activation functions with respect to discrepant tasks, as illustrated in Figures 2 and 4.

## IV. EXPERIMENTS

### A. Datasets

In the following sections we describe the four datasets used in this study. The EEG data in these datasets have all been recorded with the international 10 – 20 system.

1) *SEED*: The SEED dataset was collected as discussed in [1] to perform three emotion classification tasks (positive, neutral, and negative). 15 film clips were chosen as stimuli in the experiments. 15 subjects (8 females and 7 males, with an average age of  $23.3 \pm 2.4$ ) participated in the experiments. Each subject performed experiments in two runs of experiment with 15 sessions in each run, yielding a total of 30 sessions. Each session includes four stages: 5 seconds notice before the movie starts, around 4 minutes of movie watching, 45 seconds of self-assessment, and 15 seconds of rest. 62 EEG channels were recorded at a sampling frequency of 1000Hz. EEG signals are split into EEG segments of  $T = 8$  seconds with no overlap, as presented in Table I.

2) *SEED-VIG*: The SEED-VIG dataset was collected by [3] to estimate driver vigilance. A total of 23 subjects (12 female and 11 male, with an average age of  $23.3 \pm 1.4$ ) participated in the experiment. 17 channels EEG were collected at sampling frequency of 1000Hz. The duration of each experiment was around 2 hours, yielding 885 EEG trials in total. Participants

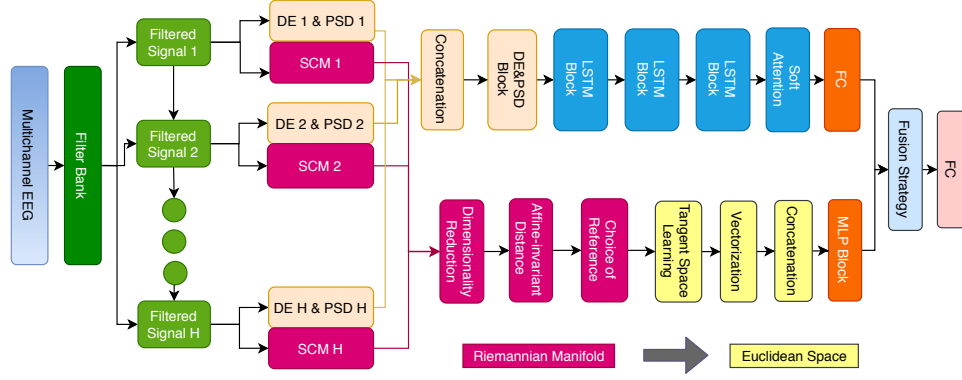


Fig. 2. The overview of the experiment work-flow is presented.

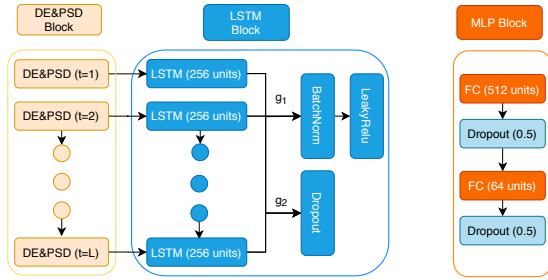


Fig. 3. The details of the blocks used in the main framework.

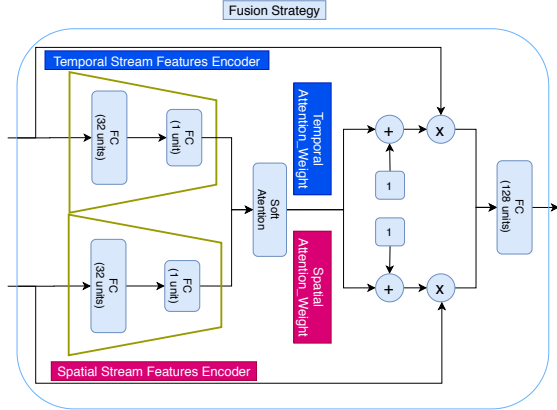


Fig. 4. The fusion strategy used in our network.

were asked to drive the simulated car in a virtual environment. Most experiments were performed after lunch so that fatigue during simulated driving could be easily induced [3]. The vigilance estimation annotation used a metric called PERCLOS [3], which was measured using eye-tracking glasses. Similar to the SEED dataset, EEG signals are split into EEG segments of 8 seconds, with no overlap, as shown in Table I.

3) *BCI-IV 2a*: The BCI-IV 2a dataset was collected by [53] to classify four MI tasks (left hand, right hand, tongue and both feet). Each of 9 subjects (4 female and 5 male, with an average age of  $23.1 \pm 2.6$ ) participated experiments in two session on two different days. Each session contain 72 trials

for each of the four classes, yielding 288 trials in total. All sessions contain data without feedback. In these sessions, each trial length is 7.5 seconds including fixation, visual cue and MI period, and rest. 22 EEG channels were recorded at the sampling frequency of  $250\text{Hz}$  during the experiment. We use the interval of  $[2.0 - 6.0\text{s}]$  in each trial ( $T = 4\text{s}$ ), as presented in Table I.

4) *BCI-IV 2b*: The BCI-IV 2b dataset was collected by [2] to perform binary MI classification (left hand versus right hand). Each of the 9 subjects (4 female and 5 male, with an average age of  $24.2 \pm 3.7$ ) participated in five sessions of the experiment. The first two sessions were conducted without feedback while rest three sessions were conducted with feedback. Each of the first two sessions contain 120 trials and each of last three sessions contain 160 trials. In the sessions containing data without feedback, each trial length is 8 seconds including fixation, visual cue, MI period, and rest. In the sessions containing data with feedback, each trial length is 8 seconds including visual cue, feedback period, and rest. 3 EEG channels were recorded at a sampling frequency of  $250\text{Hz}$ . We use the interval of  $[3.4 - 7.4\text{s}]$  in each trial ( $T = 4\text{s}$ ), as presented in Table I.

## B. Implementation details

1) *Filter Bank*: Prior to feature extraction, two types of filter banks were adopted. For the SEED dataset, DE and logarithm PSD features were calculated on the STFT outputs from five important EEG rhythms, notably delta ( $1 - 3\text{Hz}$ ), theta ( $4 - 7\text{Hz}$ ), alpha ( $8 - 13\text{Hz}$ ), beta ( $14 - 30\text{Hz}$ ), and gamma ( $31 - 50\text{Hz}$ ) bands [1]. For SEED-VIG and both MI datasets, DE and logarithm PSD features were determined on the STFT outputs in the range of  $(0.5 - 50.5\text{Hz})$  using a  $2\text{Hz}$  resolution [3], yielding a total of 50 frequency sub-bands, as shown in Table I.

2) *Temporal Information Stream*: The total number of DE and logarithm PSD features extracted from each of the Hanning windows is  $2 \times H \times N$ , as presented in Table I. These extracted features are then fed to our attention based LSTM network. Dropout rates of 0.2, 0, 1, 0.1 are applied after each LSTM layer with 256 hidden units respectively to reduce overfitting in BCI-IV 2a dataset, and  $g1 = 0, g2 = 1$ ,



TABLE I  
IMPLEMENTATION DETAILS FOR ALL FOUR DATASETS.

| Dataset   | Filter Bank |                | EEG Trial | Temporal Information Stream |                | Spatial Information Stream |           |                                  |
|-----------|-------------|----------------|-----------|-----------------------------|----------------|----------------------------|-----------|----------------------------------|
| Dataset   | $H$         | Range          | $T$       | $L$                         | Features No.   | $N$                        | Best Rank | Features No.                     |
| SEED      | 5           | $1.0 - 50.0Hz$ | $8s$      | 15                          | $10 \times 62$ | 62                         | 48        | $5 \times 48 \times (48 + 1)/2$  |
| SEED-VIG  | 25          | $0.5 - 50.5Hz$ | $8s$      | 15                          | $50 \times 17$ | 17                         | 11        | $25 \times 11 \times (11 + 1)/2$ |
| BCI-IV 2a | 25          | $0.5 - 50.5Hz$ | $4s$      | 7                           | $50 \times 22$ | 22                         | 18        | $25 \times 18 \times (18 + 1)/2$ |
| BCI-IV 2b | 25          | $0.5 - 50.5Hz$ | $4s$      | 7                           | $50 \times 3$  | 3                          | 3         | $25 \times 3 \times (3 + 1)/2$   |

as shown in Figure 3. For the rest of datasets, Batch Normalization (BatchNorm) is applied after each LSTM layer to accelerate the training phase. BatchNorm layers reduce the covariance shift of LSTM output values in each batch, thus increasing model stability [54]. Then, LeakyReLU (slope of 0.3) is adopted to enable the activation of hidden neurons for the BatchNorm layer's negative output values [55], and  $g_1 = 1, g_2 = 0$ , as shown in Figure 3. An FC layer of 64 units is used for temporal information embedding before fusion, as shown in Figure 2.

3) *Spatial Information Stream*: Total number of spatial information features in concatenated feature vectors  $\Phi_{C_{ref}}(C_i)$  from all the frequency sub-bands is  $H \times R \times (R + 1)/2$ , as presented in Table I. As presented in Figure 3, the MLP block consists of two FC layers with 512 and 64 units where each FC layer is followed by a dropout layer (drop rate of 0.5) to prevent overfitting.

4) *Fusion Strategy*: As shown in Figure 4, each of the two encoders used in the fusion block contains an FC layer of 32 units followed by a single-unit FC layer. The two encoders learn the temporal- and spatial-specific features respectively, as mentioned earlier in Section III.E. Lastly, successive to the employed soft-attention mechanism, an FC layer with 128 units is used to learn the fused and weighted embeddings.

5) *Loss Function*: The loss function of the model and the activation function of the output layer (the final FC in the model as shown in Figure 4) have been chosen with respect to different task. Since the different datasets involve different classification or regression tasks, different activation function were selected accordingly. Particularly, softmax, sigmoid, softmax and sigmoid were used for SEED, SEED-VIG, BCI-IV 2a, and BCI-IV 2b dataset respectively. Moreover, loss functions were selected with consideration of the different tasks and activation functions. Specifically, categorical cross-entropy, mean squared error, categorical cross-entropy and binary cross-entropy were used for the 4 datasets respectively. Adam optimizer [56] with default learning rate is used to help minimize the loss. We use 200 epochs and batch size of 32 to efficiently train our network. The pipeline is implemented using TensorFlow on a pair of NVIDIA RTX 2080Ti GPUs and all the hyper-parameters were systematically tuned for best performance.

### C. Evaluation Protocol

To evaluate our architecture, we adopt the same subject-dependent protocols that have been used in the original papers accompanying the datasets. In the following sections we describe the evaluation protocol details and metrics in detail for each dataset.

TABLE II  
COMPARISON OF DIFFERENT SOLUTIONS AND RESULTS FOR THE SEED DATASET.

| Paper       | Year | Input        | Method   | Acc. $\pm$ SD       |
|-------------|------|--------------|----------|---------------------|
| [1]         | 2015 | DE           | SVM      | $0.8399 \pm 0.0972$ |
| [1]         | 2015 | DE           | DBN      | $0.8608 \pm 0.0834$ |
| [5]         | 2017 | DE           | GSCCA    | $0.8296 \pm 0.0995$ |
| [25]        | 2017 | DE           | GELM     | $0.9107 \pm 0.0754$ |
| [28]        | 2018 | DE           | STRNN    | $0.8950 \pm 0.0763$ |
| [34]        | 2018 | DE           | DGCNN    | $0.9040 \pm 0.0849$ |
| [31]        | 2018 | DE           | BiDANN   | $0.9238 \pm 0.0704$ |
| [24]        | 2019 | DE           | BiHDM    | $0.9312 \pm 0.0606$ |
| [23]        | 2019 | DE           | GRSLR    | $0.8841 \pm 0.0821$ |
| [32]        | 2019 | DE           | R2G-STNN | $0.9338 \pm 0.0596$ |
| [35]        | 2020 | DE           | RGNN     | $0.9424 \pm 0.0595$ |
| [33]        | 2020 | DE           | VPR      | $0.9430 \pm 0.0650$ |
| <b>Ours</b> | 2020 | SCM, DE, PSD | RFNet    | $0.9372 \pm 0.0571$ |

1) *SEED*: As in [1], we use the pre-defined 9 sessions as training data and the remaining 6 sessions as testing data in each experiment run, yielding 248 and 170 EEG trials for training and testing, respectively.

2) *SEED-VIG*: For this dataset, we use 5-fold cross-validation to split the data into training and testing sets, as in [3]. Two frequently used evaluation metrics for regression, notably Root Mean Squared Error (RMSE) and Pearson Correlation Coefficient (PCC) have been used [3].

3) *BCI-IV 2a*: As in [53], we use the pre-defined training and testing data to evaluate our model, where each contains 288 EEG trials. Both Accuracy (Acc.) and Kappa values ( $K = \frac{P_0 - P_e}{1 - P_e}$ ) are used where  $P_0$  is the observed agreement ratio (identical to accuracy), and  $P_e$  is the expected agreement ratio while labels are assigned randomly. This metric aims to evaluate the agreement between two label vectors.

4) *BCI-IV 2b*: As per [2], we use the pre-defined training data (first three sessions) with a total of 400 trials and testing data (last two sessions) with a total of 320 trials to evaluate our model. We also use the same evaluation metrics as in the BCI-IV 2a dataset.

### D. Results and Comparison

1) *SEED*: Table II shows the performance comparison between our model and other related works on the SEED dataset. We compare our results to the existing methods that include statistical models, machine learning method, and deep learning algorithms. Generally, deep learning techniques outperforms machine learning methods (e.g., SVM, KNN, LR [1]). In [34], [35], DGCNN and RGNN learned the spatial information through discovering the topological structure of EEG channels using graphs, achieving accuracies of 90.40%



TABLE III  
COMPARISON OF DIFFERENT SOLUTIONS AND RESULTS FOR THE SEED-VIG DATASET.

| Paper       | Year | Input        | Method | RMSE $\pm$ SD                         | PCC $\pm$ SD                          |
|-------------|------|--------------|--------|---------------------------------------|---------------------------------------|
| [57]        | 2016 | DE           | GELM   | 0.1037 $\pm$ 0.0309                   | 0.7013 $\pm$ 0.1045                   |
| [29]        | 2016 | DE           | LSTM   | 0.0927 $\pm$ 0.0259                   | 0.8237 $\pm$ 0.0831                   |
| [3]         | 2017 | DE           | SVR    | 0.1327 $\pm$ 0.0303                   | 0.7001 $\pm$ 0.2250                   |
| [26]        | 2018 | DE           | DNNSN  | 0.1175 $\pm$ 0.0420                   | 0.7201 $\pm$ 0.1706                   |
| <b>Ours</b> | 2020 | SCM, DE, PSD | RFNet  | <b>0.0348 <math>\pm</math> 0.0265</b> | <b>0.9890 <math>\pm</math> 0.0081</b> |

and 94.24%, respectively. In [28], [32], STRNN and R2G-STNN utilized both spatial and temporal information with RNN or LSTM to provide performances of 89.50% and 93.38%, respectively. In [24], [31], BiDANN and BiHDM employed DAN to utilize the prior distribution information of the target domain, achieving very high performances of 92.38% and 93.12% respectively. Our model fully explores the spatial and temporal information, approaching the state-of-the-art result without prior information of the target domain.

2) *SEED-VIG*: The comparison of our model and other existing work on the SEED-VIG dataset is shown in Table III. In [3] a baseline SVR model obtained an RMSE of 0.1327 and a PCC of 0.7001. In [26], DNNSN used subnetwork nodes to process the DE features, achieving an RMSE of 0.1175 and a PCC of 0.7201. In [57], GELM outperformed SVR with an RMSE of 0.1037 and a PCC of 0.7013. In [29], temporal dependency information learned by LSTM provided a considerable results with an RMSE of 0.0927 and a PCC of 0.8237. Our model achieves considerably superior results with an RMSE of 0.0348 and a PCC of 0.9890, *setting a new state-of-the-art* for this dataset.

3) *BCI-IV 2a*: We compare the performance of our architecture on this dataset to other methods as presented in Table IV. In [4], [36], [37], pipelines consisting of CSP or FBCSP as feature extractors, followed by machine learning technique (e.g., NB, LDA) have been implemented. In [58], a filter method based on Multivariate Empirical Mode Decomposition (MEMD) was employed. In [37], CSP followed by LDA achieves a best kappa value of 0.6156. In [59], the use of a CNN applied to SCM outperformed the aforementioned pipelines with a kappa of 0.6594 and accuracy of 0.7446. In [47], pipelines used the LIE approach to extract spatial features, followed by an SVM classifier. Pipelines using CSP as feature extractor and CNN and MLP as classifier achieved an accuracy of 70.60%. Our model *achieves state-of-the-art* results with a kappa of 0.6734 and an accuracy of 75.51%, among the related works. For fair comparison, we do not consider the references that have employed different evaluation protocols on this dataset (e.g., [42], [46]). Moreover, we do not compare our results to references that have performed binary classification (e.g. [45]).

4) *BCI-IV 2b*: Table V presents the results of our method and related works using this dataset. In [36], FBCSP followed by NB as the classifier shows very good performance with a kappa of 0.6000, obtaining the first rank in the BCI competition. A very similar method, using an RF instead of the NB achieves very similar results in [39]. In [43], deep learning techniques such as CNN and CapsNet have been employed to learn the discriminative information from

TABLE IV  
COMPARISON OF DIFFERENT SOLUTIONS AND RESULTS FOR THE BCI-IV 2A DATASET.

| Paper       | Year | Input        | Method         | K/Acc. $\pm$ SD   |
|-------------|------|--------------|----------------|---|
| [36]        | 2012 | SCM          | CSP + NB       | K: 0.5700 $\pm$ 0.1830  |
| [4]         | 2012 | SCM          | FBCSP + NB     | K: 0.5720 $\pm$ 0.2123  |
| [37]        | 2013 | SCM          | CSP + LDA      | K: 0.6156 $\pm$ 0.1961  |
| [58]        | 2018 | SCM          | MEMD           | K: 0.6011 $\pm$ 0.2273  |
| [59]        | 2018 | SCM, Raw EEG | CNN            | K: 0.6594 $\pm$ 0.2044  |
| [47]        | 2019 | SCM          | LIE + SVM      | K: 0.5633 $\pm$ 0.2128  |
| [41]        | 2015 | SCM          | CSP + CNN, MLP | Acc.:0.7060 $\pm$ 0.1560  |
| [59]        | 2018 | SCM, Raw EEG | CNN            | Acc.:0.7446 $\pm$ 0.1533  |
| <b>Ours</b> | 2020 | SCM, DE, PSD | RFNet          | K: <b>0.6734 <math>\pm</math> 0.1381</b><br>Acc.: <b>0.7551 <math>\pm</math> 0.1058</b> |

TABLE V  
COMPARISON OF DIFFERENT SOLUTIONS AND RESULTS FOR THE BCI-IV 2B DATASET.

| Paper       | Year | Input        | Method       | K/Acc. $\pm$ SD   |
|-------------|------|--------------|--------------|---|
| [36]        | 2012 | SCM          | FBCSP + NB   | K: 0.6000 $\pm$ 0.2762  |
| [39]        | 2014 | SCM          | FBCSP + RF   | K: 0.5988 $\pm$ 0.2611  |
| [40]        | 2018 | SCM          | CSP + ESAFIS | K: 0.6174 $\pm$ 0.1822<br>Acc.:0.8090 $\pm$ 0.0907                                      |
| [43]        | 2019 | Spectrogram  | CNN          | Acc.:0.7499 $\pm$ 0.1452  |
| [43]        | 2019 | Spectrogram  | CapsNet      | Acc.:0.7700 $\pm$ 0.1472  |
| <b>Ours</b> | 2020 | SCM, DE, PSD | RFNet        | K: <b>0.6720 <math>\pm</math> 0.2800</b><br>Acc.: <b>0.8360 <math>\pm</math> 0.1390</b> |

spectrograms instead of SCM, achieving accuracies of 74.99% and 77.00%, respectively. In [40], a method using CSP for feature extraction and ESAFIS for classification obtained the best result with a kappa of 0.6174 and an accuracy of 80.90%. Our framework achieved considerably better results with a kappa of 0.6720 and an accuracy of 83.60%, *setting a new state-of-the-art*. Similar to other BCI-IV 2a, references that use different evaluation protocols (e.g., [60]–[62]) are not listed in this table.

## E. Discussion

Table VI presents the summary of the performance of our RFNet model compared to the state-of-the-art in the four datasets. We also show the performance of our individual learning streams with the same parameter settings as used in RFNet. We observe that the spatial information stream performs better in both MI datasets while the temporal information stream performs superior for emotion recognition and vigilance estimation datasets. This demonstrates the necessity to exploit both spatial and temporal information from EEG, in order to develop a generalized model suitable for different BCI applications (e.g., emotion recognition, vigilance estimation, and MI classification). Moreover, we observe that our model

achieves much better results than both individual learning streams even when the difference among the performance of the two streams is very small as with the BCI-IV 2b dataset. Interestingly, the performance of our model is only slightly better than each individual stream when the difference between them is large, as seen with the SEED-VIG and BCI-IV 2a datasets. This indicates that the two streams are likely to contain more contradictory information, resulting in difficulty for the model to learn a strong relationship between learned representations and outputs.

### F. Ablation Experiments

We conduct numerous ablation studies to evaluate the impact of different components of our framework on the performance.

1) *Impact of LSTM layers on Temporal Information Learning*: We evaluate the depth of the LSTM network and the performance of the LSTM compared with BiLSTM on learning temporal information. As shown in Figure 5, LSTM with three layers consistently has the best performance among LSTMs with different numbers of layers. Also, the LSTM performs better than Bi-LSTM with the same number of layers for most datasets.

2) *Importance of Riemannian Approach on Spatial Information Learning*: To show the importance of the Riemannian approach on spatial information learning, we compare our solution with a Euclidean approach that directly employs vectorization followed by MLP on spatial covariance matrices ( $\{C_i\}_{i=1}^P$ ). [45]. We also implement other deep learning techniques such as CNN and CapsNet [63] directly on  $\{C_i\}_{i=1}^P$  without vectorization for comparison. Table VII shows the comparison of these different approaches applied on SCM for spatial information learning. Our Riemannian approach consistently outperforms other approaches for all 4 datasets.

Next, we explore the learned representation space using Uniform Manifold Approximation and Projection (UMAP) [64] to better understand the impact of our Riemannian approach. Figure 7 shows the comparison between the feature spaces using our Riemannian approach  $\Phi_{C_{ref}}(\{C_i\}_{i=1}^P)$  versus a direct vectorization of spatial covariance matrices without Riemannian  $\text{Vect}(\{C_i\}_{i=1}^P)$  for a sample subject. In SEED, SEED-VIG, and BCI-IV 2a datasets, the information in  $\Phi_{C_{ref}}$  with the Riemannian approach are clearly more separable than the information in  $\text{Vect}(\{C_i\}_{i=1}^P)$  without Riemannian. In BCI-IV 2b, the difference in separability is very small. This is likely due to the limited number of channels ( $N = 3$ ). Our observations are consistent with the comparison results shown in Table VII. Overall, RFNet results in superior separability in the feature space.

3) *Impact of Dimensionality Reduction on Spatial Information Learning*: We evaluate the effect of dimensionality reduction by observing the performance of spatial information learning with different  $R$  values representing the full rank of the covariance matrix. To this end, we perform a grid search on  $R$  in the range of  $[1, N - 1]$ . Figure 6 shows the effect of dimensionality reduction with different  $R$  values on spatial information learning, based on different evaluation metrics

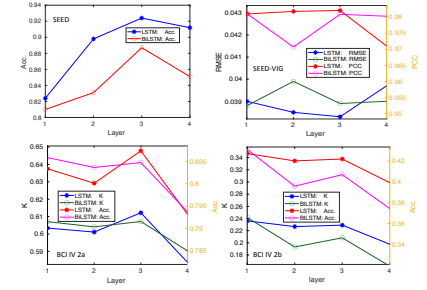


Fig. 5. Impact of LSTM layers on the temporal stream.

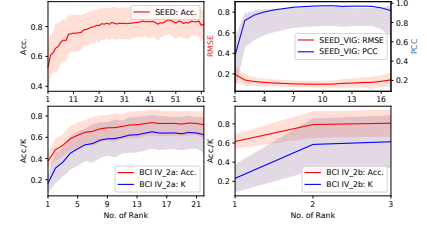


Fig. 6. Effect of dimensionality reduction on the spatial stream.

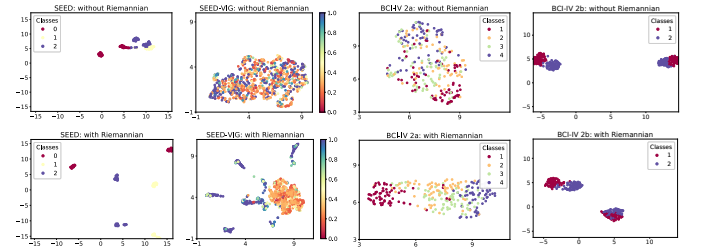


Fig. 7. Comparison between spatial information vectors without Riemannian (1<sup>st</sup> row) and with Riemannian (2<sup>nd</sup> row) using UMAP.

for the 4 datasets. We observe that the best performances are achieved at the rank  $R$  of 48, 11, 18 for SEED, SEED-VIG, and BCI-IV 2a datasets, respectively. For the BCI-IV 2b dataset, only 3 EEG channels are available, therefore dimensionality reduction is not necessary, hence, the best performance has been expectedly achieved at  $N = 3$ .

4) *Impact of Fusion strategy on Both Learning Embeddings*: We employ different feature fusion techniques such as naive concatenation, soft attention, and our fusion strategy. As shown in Table VIII, compared with naive concatenation and soft attention mechanisms, our fusion strategy provides better performance for all the 4 datasets, showing the robustness of our RFNet architecture.

## V. CONCLUSIONS

In this paper, we propose a novel deep EEG representation fusion architecture to learn the most discriminative and complementary spatial and temporal information. Spatial information are efficiently learned from covariance matrices through our Riemannian approach. Temporal information are obtained from features extracted from different time-steps in EEG sequences through our deep LSTM network with attention.

TABLE VI  
RESULT SUMMARY FOR ALL FOUR DATASETS AS WELL AS THE DIFFERENT STREAMS WITHIN OUR NETWORK.

| Dataset          | SEED         | SEED-VIG     |              | BCI-IV 2a    |              | BCI-IV 2b    |              |
|------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Metric           | Acc.         | RMSE         | PCC          | K            | Acc.         | K            | Acc.         |
| State-of-the-art | [33]: 0.9430 | [29]: 0.0927 | [29]: 0.8237 | [59]: 0.6594 | [59]: 0.7446 | [40]: 0.6174 | [40]: 0.8090 |
| Temporal         | 0.9240       | 0.0383       | 0.9821       | 0.2291       | 0.4219       | 0.6144       | 0.8073       |
| Spatial          | 0.8570       | 0.0918       | 0.8830       | 0.6636       | 0.7477       | 0.6217       | 0.8111       |
| RFNet            | 0.9372       | 0.0348       | 0.9890       | 0.6734       | 0.7551       | 0.6720       | 0.8360       |

TABLE VII  
IMPACT OF RIEMANNIAN APPROACH ON SPATIAL INFORMATION LEARNING.

| Dataset       | SEED            | SEED-VIG        |                 | BCI-IV 2a       |                 | BCI-IV 2b       |                 |
|---------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Metric        | Acc.±SD         | RMSE±SD         | PCC±SD          | K±SD            | Acc.±SD         | K±SD            | Acc.±SD         |
| SCM+Vect+MLP  | 0.7560 ± 0.1150 | 0.1593 ± 0.0678 | 0.6421 ± 0.2193 | 0.2043 ± 0.1187 | 0.4031 ± 0.0890 | 0.6071 ± 0.2767 | 0.8037 ± 0.1373 |
| SCM+CNN       | 0.7771 ± 0.1231 | 0.1751 ± 0.0537 | 0.5423 ± 0.2161 | 0.3820 ± 0.1896 | 0.5365 ± 0.1430 | 0.3737 ± 0.1989 | 0.6869 ± 0.0991 |
| SCM+CapsNet   | 0.6853 ± 0.1407 | 0.1903 ± 0.0658 | 0.4763 ± 0.1710 | 0.2684 ± 0.1558 | 0.4513 ± 0.1153 | 0.3412 ± 0.2047 | 0.6703 ± 0.1021 |
| SCM+Riem.+MLP | 0.8570 ± 0.9322 | 0.0918 ± 0.0276 | 0.8830 ± 0.0849 | 0.6636 ± 0.1437 | 0.7477 ± 0.1172 | 0.6217 ± 0.3001 | 0.8111 ± 0.1380 |

TABLE VIII  
IMPACT OF DIFFERENT FUSION METHODS ON LEARNED EMBEDDINGS.

| Dataset        | SEED            | SEED-VIG        |                 | BCI-IV 2a       |                 | BCI-IV 2b       |                 |
|----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Metric         | Acc.±SD         | RMSE±SD         | PCC±SD          | K±SD            | Acc.±SD         | K±SD            | Acc.±SD         |
| Concatenation  | 0.9100 ± 0.0825 | 0.0355 ± 0.0261 | 0.9857 ± 0.0091 | 0.6405 ± 0.1586 | 0.7304 ± 0.1190 | 0.6623 ± 0.2520 | 0.8312 ± 0.1264 |
| Soft attention | 0.9250 ± 0.0711 | 0.0350 ± 0.0227 | 0.9887 ± 0.0089 | 0.6619 ± 0.1422 | 0.7464 ± 0.1066 | 0.6397 ± 0.2600 | 0.8203 ± 0.1295 |
| RFNet          | 0.9372 ± 0.0571 | 0.0348 ± 0.0265 | 0.9890 ± 0.0081 | 0.6734 ± 0.1381 | 0.7551 ± 0.1058 | 0.6720 ± 0.2800 | 0.8360 ± 0.1390 |

Our fusion strategy exploits the complementary information from both information streams. We tested our framework with four public datasets with various types of tasks in the three popular EEG fields of emotion recognition, vigilance estimation, and MI classification. Our results demonstrate the robustness of our model in both fields on binary classification, multi-class classification, and even regression. We set new state-of-the-art results for MI classification on BCI-IV 2a and BCI-IV 2b datasets, as well as vigilance estimation on SEED-VIG, while approaching the existing state-of-the-art for emotion recognition on the SEED dataset.

## REFERENCES

- [1] W.-L. Zheng and B.-L. Lu, "Investigating critical frequency bands and channels for eeg-based emotion recognition with deep neural networks," *IEEE Transactions on Autonomous Mental Development*, vol. 7, no. 3, pp. 162–175, 2015.
- [2] R. Leeb, C. Brunner, G. Müller-Putz, A. Schlögl, and G. Pfurtscheller, "Bci competition 2008–graz data set b," *Graz University of Technology, Austria*, pp. 1–6, 2008.
- [3] W.-L. Zheng and B.-L. Lu, "A multimodal approach to estimating vigilance using eeg and forehead eeg," *Journal of neural engineering*, vol. 14, no. 2, p. 026017, 2017.
- [4] K. K. Ang, Z. Y. Chin, C. Wang, C. Guan, and H. Zhang, "Filter bank common spatial pattern algorithm on bci competition iv datasets 2a and 2b," *Frontiers in Neuroscience*, vol. 6, p. 39, 2012.
- [5] W. Zheng, "Multichannel eeg-based emotion recognition via group sparse canonical correlation analysis," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 9, no. 3, pp. 281–290, 2016.
- [6] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, "A review of classification algorithms for eeg-based brain–computer interfaces," *Journal of Neural Engineering*, vol. 4, no. 2, p. R1, 2007.
- [7] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, and F. Yger, "A review of classification algorithms for eeg-based brain–computer interfaces: a 10 year update," *Journal of Neural Engineering*, vol. 15, no. 3, p. 031005, 2018.
- [8] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Geometric means in a novel vector space structure on symmetric positive-definite matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 29, no. 1, pp. 328–347, 2007.
- [9] E. K. Kalunga, S. Chevallier, Q. Barthélemy, K. Djouani, Y. Hamam, and E. Monacelli, "From euclidean to riemannian means: Information geometry for ssvep classification," in *International Conference on Geometric Science of Information*. Springer, 2015, pp. 595–604.
- [10] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Log-euclidean metrics for fast and simple calculus on diffusion tensors," *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, vol. 56, no. 2, pp. 411–421, 2006.
- [11] X. Pennec, P. Fillard, and N. Ayache, "A riemannian framework for tensor computing," *International Journal of Computer Vision*, vol. 66, no. 1, pp. 41–66, 2006.
- [12] Z. Huang, R. Wang, S. Shan, X. Li, and X. Chen, "Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification," in *International Conference on Machine Learning*, 2015, pp. 720–729.
- [13] P. T. Fletcher and S. Joshi, "Riemannian geometry for the statistical analysis of diffusion tensor data," *Signal Processing*, vol. 87, no. 2, pp. 250–262, 2007.
- [14] S. P. van den Broek, F. Reinders, M. Donderwinkel, and M. Peters, "Volume conduction effects in eeg and meg," *Electroencephalography and Clinical Neurophysiology*, vol. 106, no. 6, pp. 522–534, 1998.
- [15] D. Sabbagh, P. Ablin, G. Varoquaux, A. Gramfort, and D. A. Engemann, "Manifold-regression to predict from meg/eeg brain signals without source modeling," in *Advances in Neural Information Processing Systems*, 2019, pp. 7321–7332.
- [16] W. Wu, Z. Chen, X. Gao, Y. Li, E. N. Brown, and S. Gao, "Probabilistic common spatial patterns for multichannel eeg analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 639–653, 2014.
- [17] V. Mishuhina and X. Jiang, "Feature weighting and regularization of common spatial patterns in eeg-based motor imagery bci," *IEEE Signal Processing Letters*, vol. 25, no. 6, pp. 783–787, 2018.
- [18] M. Congedo, A. Barachant, and R. Bhatia, "Riemannian geometry for eeg-based brain–computer interfaces; a primer and a review," *Brain-Computer Interfaces*, vol. 4, no. 3, pp. 155–174, 2017.
- [19] M. A. Uusitalo and R. J. Ilmoniemi, "Signal-space projection method for separating meg or eeg into components," *Medical and Biological Engineering and Computing*, vol. 35, no. 2, pp. 135–140, 1997.

- [20] D. A. Engemann and A. Gramfort, "Automated model selection in covariance estimation and spatial whitening of meg and eeg signals," *NeuroImage*, vol. 108, pp. 328–342, 2015.
- [21] Z. Huang, R. Wang, S. Shan, and X. Chen, "Hybrid euclidean-and-riemannian metric learning for image set classification," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 562–577.
- [22] G. Zhang, V. Davoodnia, A. Sepas-Moghaddam, Y. Zhang, and A. Etemad, "Classification of hand movements from eeg using a deep attention-based lstm network," *IEEE Sensors Journal*, vol. 20, no. 6, pp. 3113–3122, 2019.
- [23] Y. Li, W. Zheng, Z. Cui, Y. Zong, and S. Ge, "Eeg emotion recognition based on graph regularized sparse linear regression," *Neural Processing Letters*, vol. 49, no. 2, pp. 555–571, 2019.
- [24] Y. Li, W. Zheng, L. Wang, Y. Zong, L. Qi, Z. Cui, T. Zhang, and T. Song, "A novel bi-hemispheric discrepancy model for eeg emotion recognition," *arXiv preprint arXiv:1906.01704*, 2019.
- [25] W.-L. Zheng, J.-Y. Zhu, and B.-L. Lu, "Identifying stable patterns over time for emotion recognition from eeg," *IEEE Transactions on Affective Computing*, 2017.
- [26] W. Wu, Q. J. Wu, W. Sun, Y. Yang, X. Yuan, W.-L. Zheng, and B.-L. Lu, "A regression method with subnetwork neurons for vigilance estimation using eeg and eeg," *IEEE Transactions on Cognitive and Developmental Systems*, 2018.
- [27] G. Zhang and A. Etemad, "Capsule attention for multimodal eeg and eeg spatiotemporal representation learning with application to driver vigilance estimation," *arXiv preprint arXiv:1912.07812*, 2019.
- [28] T. Zhang, W. Zheng, Z. Cui, Y. Zong, and Y. Li, "Spatial-temporal recurrent neural network for emotion recognition," *IEEE Transactions on Cybernetics*, no. 99, pp. 1–9, 2018.
- [29] N. Zhang, W.-L. Zheng, W. Liu, and B.-L. Lu, "Continuous vigilance estimation using lstm neural networks," in *International Conference on Neural Information Processing*. Springer, 2016, pp. 530–537.
- [30] H. Li, Y.-M. Jin, W.-L. Zheng, and B.-L. Lu, "Cross-subject emotion recognition using deep adaptation networks," in *International Conference on Neural Information Processing*. Springer, 2018, pp. 403–413.
- [31] Y. Li, W. Zheng, Z. Cui, T. Zhang, and Y. Zong, "A novel neural network model based on cerebral hemispheric asymmetry for eeg emotion recognition," in *IJCAI*, 2018, pp. 1561–1567.
- [32] Y. Li, W. Zheng, L. Wang, Y. Zong, and Z. Cui, "From regional to global brain: A novel hierarchical spatial-temporal neural network model for eeg emotion recognition," *IEEE Transactions on Affective Computing*, 2019.
- [33] T. Zhang, Z. Cui, C. Xu, W. Zheng, and J. Yang, "Variational pathway reasoning for eeg emotion recognition," in *AAAI*, 2020, pp. 2709–2716.
- [34] T. Song, W. Zheng, P. Song, and Z. Cui, "Eeg emotion recognition using dynamical graph convolutional neural networks," *IEEE Transactions on Affective Computing*, 2018.
- [35] P. Zhong, D. Wang, and C. Miao, "Eeg-based emotion recognition using regularized graph neural networks," *IEEE Transactions on Affective Computing*, 2020.
- [36] M. Tangermann, K.-R. Müller, A. Aertsen, N. Birbaumer, C. Braun, C. Brunner, R. Leeb, C. Mehring, K. J. Müller, G. Müller-Putz *et al.*, "Review of the bci competition iv," *Frontiers in Neuroscience*, vol. 6, p. 55, 2012.
- [37] H. Ghaheri and A. Ahmadyfard, "Extracting common spatial patterns from eeg time segments for classifying motor imagery classes in a brain computer interface (bci)," *scientiainiranica*, vol. 20, no. 6, pp. 2061–2072, 2013.
- [38] M. Hersche, T. Rellstab, P. D. Schiavone, L. Cavigelli, L. Benini, and A. Rahimi, "Fast and accurate multiclass inference for mi-bcis using large multiscale temporal and spectral features," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 1690–1694.
- [39] M. Bentleimsan, E.-T. Zemouri, D. Bouchaffra, B. Yahya-Zoubir, and K. Ferroudji, "Random forest and filter bank common spatial patterns for eeg-based motor imagery classification," in *2014 5th International Conference on Intelligent Systems, Modelling and Simulation*. IEEE, 2014, pp. 235–238.
- [40] H.-J. Rong, C. Li, R.-J. Bao, and B. Chen, "Incremental adaptive eeg classification of motor imagery-based bci," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–7.
- [41] S. Sakhavi, C. Guan, and S. Yan, "Parallel convolutional-linear neural network for motor imagery classification," in *2015 23rd European Signal Processing Conference (EUSIPCO)*. IEEE, 2015, pp. 2736–2740.
- [42] M. Hersche, P. Rupp, L. Benini, and A. Rahimi, "Compressing subject-specific brain-computer interface models into one model by superposition in hyperdimensional space," in *Design, Automation and Test in Europe (DATE 2020), Grenoble, France, March 09-13, 2020*. IEEE, 2020.
- [43] K.-W. Ha and J.-W. Jeong, "Decoding two-class motor imagery eeg with capsule networks," in *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE, 2019, pp. 1–4.
- [44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [45] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Classification of covariance matrices using a riemannian-based kernel for bci applications," *Neurocomputing*, vol. 112, pp. 172–178, 2013.
- [46] X. Xie, Z. L. Yu, H. Lu, Z. Gu, and Y. Li, "Motor imagery classification based on bilinear sub-manifold learning of symmetric positive-definite matrices," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 6, pp. 504–516, 2016.
- [47] S. Li, X. Xie, Z. Gu, Z. L. Yu, and Y. Li, "Motor imagery classification based on local isometric embedding of riemannian manifold," in *2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, 2019, pp. 2368–2372.
- [48] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on riemannian manifolds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1713–1727, 2008.
- [49] W. Förstner and B. Moonen, "A metric for covariance matrices," in *Geodesy-the Challenge of the 3rd Millennium*. Springer, 2003, pp. 299–309.
- [50] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Multiclass brain-computer interface classification by riemannian geometry," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 4, pp. 920–928, 2011.
- [51] P. T. Fletcher and S. Joshi, "Principal geodesic analysis on symmetric spaces: Statistics of diffusion tensors," in *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis*. Springer, 2004, pp. 87–98.
- [52] Y. Gu, K. Yang, S. Fu, S. Chen, X. Li, and I. Marsic, "Hybrid attention based multimodal network for spoken language classification," in *Proceedings of the conference. Association for Computational Linguistics. Meeting*, vol. 2018. NIH Public Access, 2018, pp. 2379–2390.
- [53] C. Brunner, R. Leeb, G. Müller-Putz, A. Schlögl, and G. Pfurtscheller, "Bci competition 2008–graz data set a," *Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology*, vol. 16, 2008.
- [54] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.
- [55] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1, 2013, p. 3.
- [56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [57] X.-Q. Huo, W.-L. Zheng, and B.-L. Lu, "Driving fatigue detection with fusion of eeg and forehead eeg," in *IEEE International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 897–904.
- [58] P. Gaur, R. B. Pachori, H. Wang, and G. Prasad, "A multi-class eeg-based bci classification using multivariate empirical mode decomposition based filtering and riemannian geometry," *Expert Systems with Applications*, vol. 95, pp. 201–211, 2018.
- [59] S. Sakhavi, C. Guan, and S. Yan, "Learning temporal information for brain-computer interface using convolutional neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5619–5629, 2018.
- [60] J. Luo, Z. Feng, J. Zhang, and N. Lu, "Dynamic frequency feature selection based approach for classification of motor imageries," *Computers in Biology and Medicine*, vol. 75, pp. 45–53, 2016.
- [61] L. Sun, Z. Feng, B. Chen, and N. Lu, "A contralateral channel guided model for eeg based motor imagery classification," *Biomedical Signal Processing and Control*, vol. 41, pp. 1–9, 2018.
- [62] D. Li, J. Wang, J. Xu, and X. Fang, "Densely feature fusion based on convolutional neural networks for motor imagery eeg classification," *IEEE Access*, vol. 7, pp. 132 720–132 730, 2019.
- [63] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing Systems*, 2017, pp. 3856–3866.
- [64] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.