Deep Riemannian Networks with Temporal Approach for Classification in Brain-Computer Interfaces

Alexandre Herrero Matias
School of Electrical and Computer Engineering
University of Campinas (UNICAMP)
Campinas, Brazil
a223665@dac.unicamp.br

João Guilherme Prado Barbon
School of Electrical and Computer Engineering
University of Campinas (UNICAMP)
Campinas, Brazil
j262760@dac.unicamp.br

Lucas Heck dos Santos
Mathematics, Computation and Cognition Center
Federal University of ABC (UFABC)
Santo André, Brazil
lucasheck17@gmail.com

Denis Gustavo Fantinato
School of Electrical and Computer Engineering
University of Campinas (UNICAMP)
Campinas, Brazil
denisf@unicamp.br

***Abstract*—In the context of electroencephalography (EEG) classification for Motor Imagery (MI) tasks, the use of the framework based on Riemannian Geometry (RG) has shown comparable performances to convolutional neural networks. Its application on EEG data usually considers the extraction of a sample covariance matrix, which captures spatial information (between electrodes). However, the temporal information can be used as well, for instance, considering a set of time-delayed covariance matrices. In that sense, in this work, we propose the use of the temporal information, mainly considering Deep Riemannian Networks (DRN). We also present a modified version of SPDNet in order to encompass a set of covariance matrices as input. Our approach enhances SPDNet's ability to learn spatio-temporal patterns, resulting in improved classification performance on EEG data.**

***Index Terms*—Brain-Computer Interface, Electroencephalography, Riemannian Geometry.**

I. Introduction

Brain-Computer Interfaces (BCIs) represent an emerging and fascinating area of research that seeks to establish direct communication between the human brain and computer systems [11], [12]. From medical applications in neuroprosthetics to advancements in cognitive augmentation, BCIs continue to push the boundaries of neuroscience, engineering, and mathematics [9].

One of the most widely used techniques in BCIs is Electroencephalography (EEG), which records electrical activity using non-invasive electrodes positioned at the scalp. EEG provides high temporal resolution, making it very suitable for real-time applications such as motor imagery-based control, neurofeedback, and cognitive state monitoring [8]. Despite its advantages, EEG signals are often noisy, non-stationary, and high-dimensional, posing challenges to their decoding [1], [11].

Traditionally, EEG signal processing has relied on feature engineering techniques such as spectral analysis, spatial filtering, and statistical classification. With advancements in computing, machine learning approaches including deep learning (DL) have improved decoding performance by automatically extracting features and learning complex signal patterns. These have been the state-of-the-art methods in EEG classification [7], [14]–[16].

However, in competitions involving the processing of EEG signals, approaches based on Riemannian Geometry (RG) [13] have stood out in the last decade. RG offers a robust mathematical approach, allowing EEG data to be molded into spaces defined by Symmetric Positive Defined (SPD) matrices, the manifolds, which capture their fundamental geometric properties. This allows operations in a data space that ideally has less variability, thus improving the analysis.

In this context, a combination of RG and DL approaches showed to be very promissing, giving rise to the Deep Riemannian Networks (DRNs) [5], [16], [17], in which we detach the SPDNet [5], EE(G)-SPDNet [16] and TSMNet [6]. These DRNs achieved high accuracies in several EEG datasets and enabled a novel perspective for the development of BCI systems. Although DRNs are able to learn the SPD manifold, the standard use of these matrices essentially gathers spatial information. However, given the high temporal resolution in EEG signals, temporal information might be relevant as

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and grants #2023/00640-1, São Paulo Research Foundation (FAPESP).

well in classification. With that in mind, in this work, we propose to explore time-delayed covariance matrices in order to combine spatial and temporal information for RG, DL and DRN approaches. Additionally, we propose a variation of the SPDNet in order to allow the treatment of a set of time-delayed covariance matrices at once. These methods are tested for the BCI Competition IV Dataset 2A [3] and Cho2017 [4] datasets, showing results very favorable to our proposal.

The rest of this work is organized as follows. In Section II, we provide an overview of RG and its relevance to improving EEG classification. Section III presents the datasets used in this study, detailing their characteristics and preprocessing steps. In Section IV, we describe the classification models employed to evaluate the performance of the proposed approach. Section V discusses the experimental results and their implications. Finally, Section VI concludes and outlines possible directions for future work.

II. Riemannian Geometry

RG is a branch of mathematics that studies differentiable spaces, where concepts such as metrics and curvatures are defined [13]. In the context of BCIs, RG has been exploited to improve the understanding and analysis of neural patterns.

If EEG signals can be mapped onto a set of SPD matrices, RG can be applied. In this case, the statistical properties of the signals will be completely described, as long as it is assumed that the EEG signal is stationary and its statistics follow the Gaussianity assumption [10]. Mathematically, a matrix $\mathbf{P} \in \mathbb{R}^{N\times N}$ is considered SPD if it belongs to the SPD matrix space, formally defined as:

$$\mathrm{SPD}(N) = \mathrm{S}(N) \cap \mathrm{P}(N) \tag{1}$$

where $\mathrm{S}(N) = \left\{\mathbf{P} \in \mathbb{R}^{N\times N} \mid \mathbf{P} = \mathbf{P}^{\intercal}\right\}$ denotes the set of symmetric matrices, and $\mathrm{P}(N) = \left\{\mathbf{P} \in \mathbb{R}^{N\times N} \mid \mathbf{u}^{\intercal}\mathbf{P}\mathbf{u} > 0, \forall \mathbf{u} \in \mathbb{R}^{N}\right\}$ denotes the set of positive definite matrices, where $(\cdot)^{\intercal}$ denotes transposition.

When equipped with a Riemannian metric, the space of SPD matrices becomes a Riemannian differentiable manifold $\mathcal{M}$. Within this geometric framework, the concepts of Riemannian distance and tangent space are essential. Given two SPD matrices $\mathbf{A}, \mathbf{B} \in \mathrm{SPD}(N)$, their Riemannian distance is defined by:

$$\delta_R(\mathbf{A}, \mathbf{B}) = \left\| \log\left(\mathbf{A}^{-\frac{1}{2}}\mathbf{B}\mathbf{A}^{-\frac{1}{2}}\right) \right\|_F \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm:

$$\|\mathbf{M}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |m_{ij}|^2} \tag{3}$$

This Riemannian distance represents the geodesic length—i.e., the shortest path—between two points on the manifold and satisfies the axioms of a metric space: non-negativity, symmetry, and the triangle inequality.

Another key structure in Riemannian geometry is the tangent space, which is a linear approximation of the manifold at a specific point. It facilitates computations in an otherwise non-linear space. The tangent space $\mathcal{T}(N)$ at a point $\mathbf{P} \in \mathrm{SPD}(N)$ is defined as [17]:

$$\mathcal{T}(N) = \left\{ \mathbf{s}_i = \mathrm{upper}(\mathbf{P}^{-\frac{1}{2}}\mathrm{Log}_{\mathbf{P}}(\mathbf{P}_i)\mathbf{P}^{-\frac{1}{2}}) \in \mathbb{R}^{N(N+1)/2} \right\} \tag{4}$$

Here, $\mathbf{P}$ is the tangent point, i is the index of the point $\mathbf{P}_i \in \mathcal{M}$, upper$(\cdot)$ extracts and vectorizes the upper triangular part of a matrix, and $\mathrm{Log}_{\mathbf{P}}(\mathbf{P}_i) = \mathbf{P}^{\frac{1}{2}} \log\left(\mathbf{P}^{-\frac{1}{2}}\mathbf{P}_i\mathbf{P}^{-\frac{1}{2}}\right)\mathbf{P}^{\frac{1}{2}}$ denotes the Riemannian logarithmic map.

By projecting SPD matrices into the tangent space, we can approximate the Riemannian distance with a simpler Euclidean distance, as illustrated in Fig. 1.

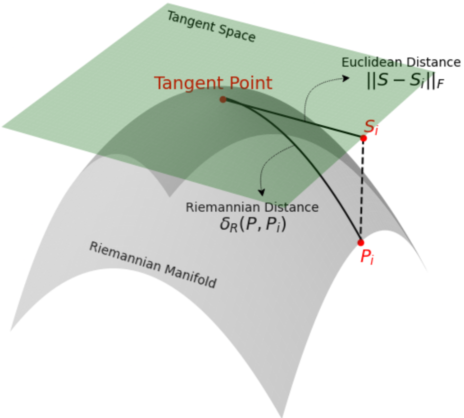

Fig. 1. Visualization of the Riemannian manifold.

A. Classical RG Approach for EEG Signals

To work with RG our data needs to be contained in the space of SPD matrices, but since EEG data are composed of a set of time series, requires a transformation/mapping. The simplest way is to estimate the auto-covariance matrix of the input EEG signal.

For a random vector $\hat{\mathbf{X}} \in \mathbb{R}^n$, the true covariance matrix is:

$$\mathbf{\Sigma} = \mathbb{E}[(\hat{\mathbf{X}} - \mu)(\hat{\mathbf{X}} - \mu)^{\intercal}] \tag{5}$$

where $\mathbb{E}[\cdot]$ is the expectation operator and $\mu = \mathbb{E}[\hat{\mathbf{X}}]$ is the expected value (mean).

Since we do not know the true distribution, we estimate $\hat{\mathbf{\Sigma}}$ using a finite set of samples. Assuming that $\mathbf{X}_i \in R^{N\times L}$ denotes the i-th windowed EEG signal with N electrodes and L time samples within the window, its auto-covariance matrix can be estimated as follows:

$$\hat{\mathbf{\Sigma}}_i = \left(\frac{1}{L-1}\right)\mathbf{X}_i\mathbf{X}_i^{\intercal} \tag{6}$$

hence, each EEG window can be mapped to an estimated auto-covariance matrix.

However before applying RG, we need to make sure that this covariance matrix is SPD. This holds if and only if $\mathbf{X}$ has a non-degenerate distribution (all the random variables in $\mathbf{X}$

need to be linearly independent), which we can assume since our input data are neural signals.

In the context of EEG analysis, the covariance matrix is estimated with respect to the electrode signals. By leveraging the geometric structure of the SPD manifold, more meaningful patterns and relationships in the EEG data can be captured, which may improve the performance of classification and feature extraction tasks.

B. The Use of Time-Delayed Covariance Matrices

The RG based approaches show great potential and its application on the research of EEG signals is in a constant stage of development, which implies the need for more in-depth analysis to evaluate its performance in different scenarios and with more complex data. Currently, RG-based approaches focus predominantly on identifying spatial patterns present in the data, which is a fundamental feature of their operation.

This approach presents a severe limitation when exploiting the temporal information that can be crucial for a more complete understanding of the signals. In tasks such as EEG analysis, temporal variations over time can contain essential information about neural dynamics, especially in movement classification tasks or cognitive processes.

In the context of RG, one possibility to encompass temporal information is to map the EEG signal to a set of SPD matrices. Basically, by considering a set of time steps, a set of "Time-Delayed Covariance Matrices" (TDCMs) can be calculated. Let $\mathbf{X} \in \mathbb{R}^{N \times L}$ be:

$$\mathbf{X}_d = \begin{bmatrix} x_{e_1}(t-d) & x_{e_1}(t-d+1) & \cdots & x_{e_1}(t-d+L) \\ x_{e_2}(t-d) & x_{e_2}(t-d+1) & \cdots & x_{e_2}(t-d+L) \\ \vdots & \vdots & \ddots & \vdots \\ x_{e_N}(t-d) & x_{e_N}(t-d+1) & \cdots & x_{e_N}(t-d+L) \end{bmatrix}_{N \times L} \tag{7}$$

where e_n represents the n-th channel, L the number of samples per channel, t the time of the sample and $x_{e_n}(t)$ the sample of the n-th channel at time t. We'll then add a d amount of delay to this input data.

Now, a TDCM (for delay d) can be estimated as:

$$\hat{\mathbf{M}}_d = \left(\frac{1}{L-1}\right) \mathbf{X}_0 {\mathbf{X}_d}^{\mathsf{T}} \tag{8}$$

being $\hat{\mathbf{M}}_d \in \mathbb{R}^{N \times N}$. Considering that no data outside the EEG window can be used, only d samples are not used for estimation of $\hat{\mathbf{M}}_d$ – and $\mathbf{X}_d$ is a $N \times L$ matrix.

Also, since $\mathbf{X}_0$ and $\mathbf{X}_d$ are not identical, to make sure $\hat{\mathbf{M}}_d$ is still an SPD matrix we use a relation that ensures symmetry with respect to the main diagonal:

$$\mathbf{M}_d = \frac{\hat{\mathbf{M}}_d + \hat{\mathbf{M}}_d^{\mathsf{T}}}{2} \tag{9}$$

Following, in order to show that $\mathbf{M}_d$ is positive definite, approximating $\mathbf{M}_d \approx \hat{\mathbf{\Sigma}}$ (equation 6), is possible to show that for any non-zero vector $\mathbf{z} \in \mathbb{R}^n$, $\mathbf{z}^{\mathsf{T}}\mathbf{M}_d\mathbf{z} > 0$:

$$\mathbf{z}^{\mathsf{T}}\mathbf{M}_d\mathbf{z} = \mathbf{z}^{\mathsf{T}}(\mathbb{E}[(\mathbf{X} - \mu)(\mathbf{X} - \mu)^{\mathsf{T}}])\mathbf{z} \tag{10}$$

Using the linearity of expectation:

$$\mathbf{z}^{\mathsf{T}}\mathbf{M}_d\mathbf{z} = \mathbb{E}[\mathbf{z}^{\mathsf{T}}(\mathbf{X} - \mu)(\mathbf{X} - \mu)^{\mathsf{T}}\mathbf{z}] \tag{11}$$

Since $\mathbf{z}^{\mathsf{T}}(\mathbf{X} - \mu)$ is a scalar, we denote it as $\mathbf{Y} = \mathbf{z}^{\mathsf{T}}(\mathbf{X} - \mu)$, so we get:

$$\mathbf{z}^{\mathsf{T}}\mathbf{M}_d\mathbf{z} = \mathbb{E}[\mathbf{Y}^2] \tag{12}$$

Since the expectation of a squared quantity is always non-negative, we conclude that $\mathbf{z}^{\mathsf{T}}\mathbf{M}_d\mathbf{z} \geq 0$. Lastly, in order to have $\mathbf{z}^{\mathsf{T}}\mathbf{M}_d\mathbf{z} > 0$, $\mathbf{M}_d$ needs to be full rank which, again, we can assume since our input data are neural signals.

1) Expanded Time-Delayed Covariance Matrices: After generating the TDCMs, we now move on to the RG training phase. However, some methods, such as SPDNet, receives as input a single covariance matrix, being necessary to perform a combination of the temporal information. In order to achieve this, we propose de use of a time-delayed expanded covariance matrix (TDECM).

TDECMs are block matrices composed of p TDCMs in a configuration analogous to a covariance matrix:

$$\mathbf{A}_p = \begin{bmatrix} \mathbf{M}_0 & \mathbf{M}_1 & \cdots & \mathbf{M}_p \\ {\mathbf{M}_1}^{\mathsf{T}} & \mathbf{M}_0 & \cdots & \mathbf{M}_{p-1} \\ \vdots & \vdots & \ddots & \vdots \\ {\mathbf{M}_p}^{\mathsf{T}} & {\mathbf{M}_{p-1}}^{\mathsf{T}} & \cdots & \mathbf{M}_0 \end{bmatrix}_{(p+1)N \times (p+1)N} \tag{13}$$

By using TDECMs we want to look for possible correlations between the different amounts of delay.

2) Delay Tensors: A more intuitive way of grouping multiple TDCMs into a single input would be to organize them into a single tensor:

$$\mathbf{T}_p = (\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_p) \tag{14}$$

where $\mathbf{T}_p \in \mathbb{R}^{N \times N \times p}$. In this case, however, the model should be able to process this data structure. In order to do so, we propose a modification on the SPDNet to process a tensor on its input. The description of the modified SPDNet is presented on Section IV-C.

III. Datasets Description

To evaluate the performance of our solutions, we employed two widely used and publicly available datasets: BCI Competition IV Dataset 2A and Cho2017. These datasets offer rich EEG recordings and well-defined experimental protocols, making them suitable benchmarks for research in signal processing and machine learning within the BCI domain.

A. BCI Competition IV Dataset 2A

The BCI Competition IV Dataset 2A is a benchmark dataset specifically designed for evaluating motor imagery classification techniques. It contains EEG recordings from 9 subjects, each performing four different MI tasks: left hand, right hand, both feet, and tongue. The data was collected using 22 EEG channels and 3 EOG channels, sampled at 250 Hz with a bandpass filter from 0.5 to 100 Hz, and a notch filter at 50 Hz to eliminate power line noise. Each subject participated in two sessions recorded on different days, with each session containing 288 trials (72 per class).

B. *Cho2017 Dataset*

The Cho2017 dataset, introduced by Cho et al. (2017), is another motor imagery dataset aimed at facilitating the development of EEG decoding methods. It includes EEG data from 54 subjects, making it significantly larger and more diverse in terms of inter-subject variability compared to most BCI datasets. Subjects were instructed to perform two MI tasks: imagining left-hand and right-hand movements. EEG signals were recorded using 64 channels at a sampling rate of 512 Hz, later preprocessed for analysis.

IV. Classification Models

In order to test the use of the temporal information, we evaluated the resulting performance through five classification models widely used in EEG-based BCI research. These models represent different paradigms, from classical Riemannian geometry-based classifiers to deep learning architectures that leverage spatial and temporal EEG features.

A. *Minimum Distance to Mean (MDM)*

A common method for classifying SPD matrices is known as Minimum Distance to Mean (MDM) [2]. Given an unlabeled SPD matrix, its label will be the same as that of the already labeled matrices whose centroid has the smallest Riemannian distance, as illustrated in Figure 2:

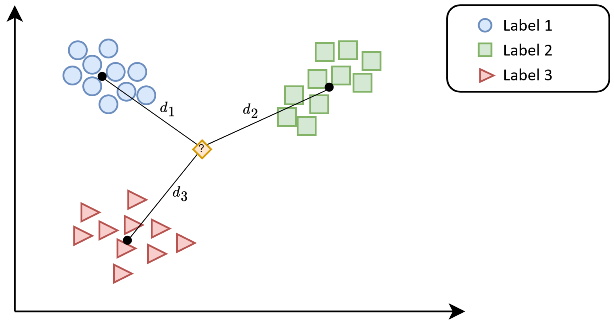

Fig. 2. MDM Algorithm.

The centroid is calculated using the Riemannian geometric mean of the positions of the matrices of the same label:

$$\mathfrak{G}_r(\mathbf{P}_{r1}, \dots, \mathbf{P}_{rn}) = \underset{\mathbf{M}\in\mathcal{M}}{\arg\min} \sum_{k=1}^{n} \delta_R^2(\mathbf{M}, \mathbf{P}_{rk}) \tag{15}$$

where r is the label of matrices we want to calculate the centroid of. With this, it is possible to describe the classification of an unlabeled SPD matrix $\mathbf{P} \in \mathcal{M}$ by the MDM algorithm as follows:

$$c = \underset{l}{\arg\min}\ \delta_R(\mathbf{P}, \mathfrak{G}_l) \tag{16}$$

where c is the label with which the matrix $\mathbf{P}$ will be labeled.

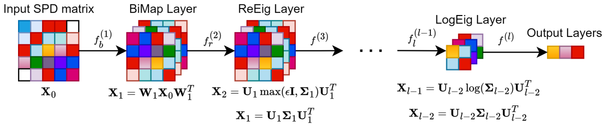
Input SPD matrix
BiMap Layer
ReEig Layer
LogEig Layer
Output Layers
$f_b^{(1)}$
$f_r^{(2)}$
$f^{(3)}$
$f_l^{(l-1)}$
$f^{(l)}$
$\mathbf{X}_0$
$\mathbf{X}_1 = \mathbf{W}_1\mathbf{X}_0\mathbf{W}_1^T$
$\mathbf{X}_2 = \mathbf{U}_1 \max(\epsilon\mathbf{I}, \mathbf{\Sigma}_1)\mathbf{U}_1^T$
$\mathbf{X}_1 = \mathbf{U}_1\mathbf{\Sigma}_1\mathbf{U}_1^T$
$\mathbf{X}_{l-1} = \mathbf{U}_{l-2} \log(\mathbf{\Sigma}_{l-2})\mathbf{U}_{l-2}^T$
$\mathbf{X}_{l-2} = \mathbf{U}_{l-2}\mathbf{\Sigma}_{l-2}\mathbf{U}_{l-2}^T$

Fig. 3. SPDNet structure. Adapted from [5]

B. *SPDNet*

SPDNet has been specifically designed to process data represented by SPD matrices, which, as described earlier, is the primary data structure used in RG. Its components are described follows:

1) **BiMap layers** (represented in the second block of Figure 3): They convert SPD matrices into new matrices, which ideally contain more information. Mathematically, it is defined as follows:

$$\mathbf{X}_k = \mathbf{W}_k\mathbf{X}_{k-1}\mathbf{W}_k^T \tag{17}$$

where $\mathbf{X}_{k-1}$ is the input SPD matrix in the k-th layer, $\mathbf{W}_k$ is the transformation matrix and $\mathbf{X}_k$ is the resulting matrix.

2) **ReEig Layers** (represented in the third block of Figure 3): They introduce non-linearity (similar to ReLU) by modifying the eigenvalues of the SPD matrix:

$$\mathbf{X}_k = \mathbf{U}_{k-1} \max(\epsilon\mathbf{I}, \mathbf{\Sigma}_{k-1})\mathbf{U}_{k-1}^T \tag{18}$$

where $\mathbf{U}_{k-1}$ and $\mathbf{\Sigma}_{k-1}$ are obtained by eigenvalue decomposition ($\mathbf{X}_{k-1} = \mathbf{U}_{k-1}\mathbf{\Sigma}_{k-1}\mathbf{U}_{k-1}^T$) and ϵ is a chosen threshold. The use of the ReEig layer prevents the matrices from becoming non-positive.

3) **LogEig Layers** (represented in the penultimate block of Figure 3): They transform the input matrix to a space where linear operations can be applied more effectively, while still respecting the intrinsic geometric structure of the SPD space:

$$\mathbf{X}_k = \mathbf{U}_{k-1} \log(\mathbf{\Sigma}_{k-1})\mathbf{U}_{k-1}^T \tag{19}$$

where $\mathbf{U}_{k-1}$ and $\mathbf{\Sigma}_{k-1}$ are obtained in the same way by eigenvalue decomposition, and the base of the logarithm is e. The mapping to tangent space via logarithm, performed by Eq. 19 allows the resulting SPD matrix to be manipulated in a space where linear operations are valid and easier to perform. These properties are important for classification in the final layers of the network.

C. *SPDNet-Par*

In order to process the TDCMs in the form of a tensor, we propose a novel DRN named SPDNet-Par, which is a modified version of the original SPDNet architecture, as shown in Fig. 4.

The standard SPDNet is not designed to process tensorial input structures, as it operates on single SPD matrices. To

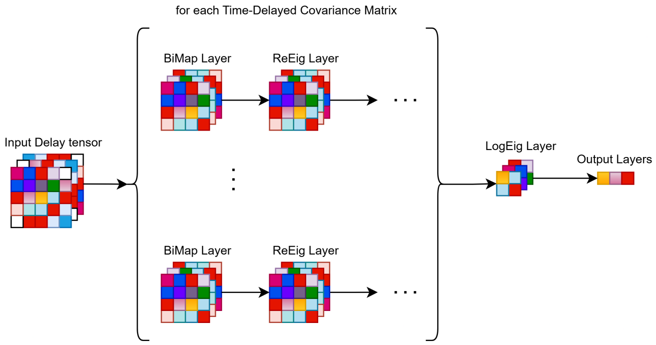

Fig. 4. SPDNet-Par structure. Adapted from [5]

address this limitation, we restructured the network to independently handle each TDCM comprising the input tensor. These matrices are processed in parallel with its intermediate outputs being subsequently aggregated and passed through the remaining layers of the network for unified processing.

D. EEGNet

EEGNet is a compact convolutional neural network (CNN) designed for efficient learning on raw EEG time-series data. As ilustrated in Fig. 5, the architecture employs:

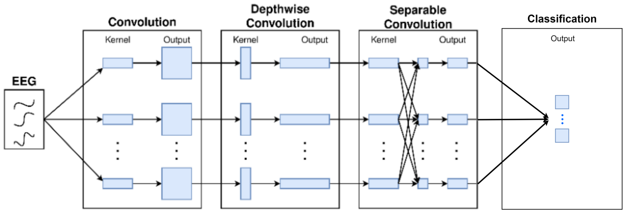

Fig. 5. EEGNet structure. Adapted from [7]

1) **Temporal Convolution**: Extracts frequency-specific patterns from raw EEG signals.
2) **Depthwise Convolution**: Captures spatial dependencies across channels, simulating bandpass filtering and spatial filtering.
3) **Separable Convolution**: Reduces parameter count while enhancing feature extraction by applying a pointwise convolution after depthwise convolution.

While EEGNet is not inherently designed to handle SPD matrices or exploit Riemannian structure, it performs well on BCI tasks, especially with limited data and low-latency constraints. However, it treats the input space as Euclidean, which may limit its effectiveness in capturing the non-linear, manifold-based structure of EEG covariance data.

To incorporate time-delayed information into the input data for the EEGNet architecture, we adopted a strategy of concatenating the temporal representations along the channels dimension. In this approach, for each predefined delay, a copy of the original EEG signal, shifted in time, is generated, and its corresponding features are stacked with the original channels. As a result, the network receives an expanded input where each original channel is augmented with its time-delayed versions.

E. EE(G)-SPDNet

EE(G)-SPDNet (End-to-End EEG SPDNet) is a DRNN capable of being trained end-to-end, equipped with a filter bank that is adapted to the datasets, therefore offering greater potential for application across different paradigms.

The solution found for incorporating the filter bank was the insertion of convolutional layers at the input of an SPDNet. This way, the one-dimensional convolution operation (along the temporal axis) acts directly on the EEG signals, functioning as a band-pass filter:

1) **Convolutional Layer**: The initial convolutional layer follows a structure similar to the EEGNet architecture, acting as temporal filters. However, there are two possible approaches:
   - **Conventional filtering (channel-independent filtering)**: a single filter (kernel) is applied to all channels — each filter produces an output of the same length as the input EEG signal (same-type convolution)
   - **Channel-specific filtering**: one or more filters per channel — this approach has more trainable parameters but offers greater flexibility in selecting individual frequencies for each channel. In this case, the outputs of each convolution are stacked, as shown in the adjacent figure.
2) **SCM Pooling Layer**: Subsequently, the SCM (Sample Covariance Matrix) pooling layer extracts the covariance matrix estimates.
   - **Channel-specific filtering**: if this option was used, the estimation of the covariance matrix is straightforward.
   - **Conventional filtering (channel-independent filtering)**: in this option, each kernel, after convolution, generates a signal with the same dimension as the input, being possible to extract a set of estimated covariance matrices for each filter:
     For instance, two SPD matrices $\mathbf{S}_n$ and $\mathbf{S}_m$ of dimensions n and m, respectively, can be combined as follows:
     $$\mathrm{Conc}(\mathbf{S}_n, \mathbf{S}_m) = \begin{bmatrix} \mathbf{S}_n & \mathbf{0}_{n\times m} \\ \mathbf{0}_{m\times n} & \mathbf{S}_m \end{bmatrix} \qquad (20)$$
     This procedure allows SPD matrices of different dimensions to be combined, generating another matrix $\mathrm{Conc}(\mathbf{S}_n, \mathbf{S}_m)$ — a block diagonal matrix — of higher dimension that is also SPD.

In this work, the filtering strategy adopted was the channel-specific filtering.

F. TSMNet

TSMNet [6] is an architecture designed to address challenges of unsupervised domain adaptation (UDA) in EEG signals, especially in inter-session and inter-subject transfer scenarios. It uses domain adaptation techniques and batch momentum normalization.

TABLE I
RESULTS

Dataset	MDM	SPDNet	SPDNet-Par	EEGNet	EEG-SPDNet	TSMNet
BNCI2014001 $p = 0$	70.54 ± 16.18	71.94 ± 15.88	68.50 ± 16.20	$\mathbf{68.98 \pm 13.79}$	71.47 ± 16.97	72.37 ± 16.73
BNCI2014001 $p = 1$	$\mathbf{72.90 \pm 14.55}$	$\mathbf{73.98 \pm 13.95}$	$\mathbf{70.40 \pm 15.50}$	64.43 ± 05.47	$\mathbf{72.58 \pm 16.71}$	70.96 ± 14.24
BNCI2014001 $p = 10$	71.12 ± 15.90	73.30 ± 14.45	64.52 ± 13.33	66.04 ± 11.36	72.14 ± 16.82	$\mathbf{74.20 \pm 15.83}$
Cho2017 $p = 0$	53.38 ± 12.74	61.01 ± 12.26	58.78 ± 11.69	$\mathbf{52.48 \pm 07.04}$	60.94 ± 11.86	62.21 ± 12.38
Cho2017 $p = 1$	53.38 ± 12.86	61.62 ± 12.58	$\mathbf{59.09 \pm 12.13}$	51.28 ± 05.82	61.29 ± 11.85	62.15 ± 12.50
Cho2017 $p = 10$	$\mathbf{54.42 \pm 13.05}$	$\mathbf{62.28 \pm 12.78}$	56.52 ± 08.15	51.57 ± 05.47	$\mathbf{61.59 \pm 12.46}$	$\mathbf{63.79 \pm 12.75}$

Its main innovation is the introduction of the SPD Domain-Specific Momentum Batch Normalization (SPDDSMBN) layer, which transforms domain-specific inputs in the SPD space into domain-invariant outputs.

V. RESULTS

To evaluate the effectiveness of our proposed methods, we compared the classification accuracy of the five classification models. For the sake of reproducibility, the hyperparameter configurations for each model are provided in Table II.

TABLE II
HYPERPARAMETERS

Hyperparameters	SPDNet	SPDNet-Par	EEGNet	EE(G)-SPDNet	TSMNet
Learning Rate	0.01	0.001	0.01	0.007	0.009
Batch Size	64	32	32	32	16
Epochs	500	500	75	100	1000

We used the no delay ($p = 0$) case as a baseline to our comparison. Across the two datasets, it is possible to see that the addition of temporal information consistently outperformed the baselines, as shown on Table I.

The results show that the best delay varies depending on the dataset characteristics. For the BCI Competition IV Dataset 2a, the best performance was achieved with a delay of 1 ($p = 1$), indicating that this dataset benefits primarily from capturing short-term temporal dependencies. In contrast, for the Cho2017 dataset, the configuration with a delay of 10 ($p = 10$) produced superior results, suggesting that this dataset contains more relevant long-term temporal dynamics.

A noteworthy point to highlight is the performance of our proposed solution, the SPDNet-Par, which was designed to incorporate the temporal information by means of parallel processing of TDCMs. Although SPDNet-Par demonstrated consistent and competitive results across both datasets, it was not able to outperform the original SPDNet. This suggests that, despite the potential of parallel temporal modeling, our strategy may not fully capture the discriminative features as effectively as the hierarchical transformations applied in the standard SPDNet pipeline.

Additionally, it is important to note that the only model for which the best result was obtained with a delay of 0 ($p = 0$) was the EEGNet. This outcome is expected, since EEGNet is the only model in our experiments that is not RG-based. Unlike the RG-based models, which benefit from covariance-based representations enriched with temporal information, EEGNet directly learns spatial-temporal patterns through convolutional operations.

To sum up, we can observe an average accuracy improvement of 2% across all subjects on Riemannian-based solutions. This demonstrates that capturing temporal dependencies through the TDCMs significantly improves the model's ability to learn discriminative features, confirming the importance of spatio-temporal modeling in EEG classification tasks.

VI. CONCLUSION

In this study, we explored the integration of temporal information into Riemannian-based deep learning models for EEG signal classification. By leveraging the structure of SPDNet and augmenting its input space with TDCMs, we aimed to overcome the limitations of purely spatial feature extraction. Our theoretical analysis confirmed that TDCMs, derived from time-lagged covariance estimations, maintain the mathematical properties required for processing in the Riemannian manifold. This extension enriches SPDNet's representational capacity, enabling it to better capture the complex spatio-temporal patterns of neural activity. These findings lay the groundwork for future empirical evaluation and optimization, with potential applications in improving BCI performance across a variety of tasks.

REFERENCES

[1] Reza Abiri, Soheil Borhani, Eric W Sellers, Yang Jiang, and Xiaopeng Zhao. A comprehensive review of EEG-based brain-computer interface paradigms. *Journal of neural engineering*, 16(1):011001, 2019.

[2] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. Multiclass Brain–Computer Interface Classification by Riemannian Geometry. *IEEE Transactions on Biomedical Engineering*, 59(4):920–928, 2012.

[3] Clemens Brunner, Robert Leeb, and Gernot Müller-Putz. BCI Competition 2008–Graz data set A, 2024.

[4] H. Cho, M. Ahn, S. Ahn, M. Kwon, and S. C. Jun. EEG datasets for motor imagery brain-computer interface, 2017.

[5] Zhiwu Huang and Luc Van Gool. A Riemannian Network for SPD Matrix Learning, 2016.

[6] Reinmar J. Kobler, Jun-ichiro Hirayama, Qibin Zhao, and Motoaki Kawanabe. SPD domain-specific batch normalization to crack interpretable unsupervised domain adaptation in EEG. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc.

[7] Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces. *Journal of Neural Engineering*, 15(5):056013, 2018.

[8] Sarah Negreiros de Carvalho Leite. *Contribuições ao Desenvolvimento de Interfaces Cérebro-Computador Baseadas em Potenciais Evocados Visualmente em Regime Estacionário*. Phd thesis, University of Campinas, 2016.

[9] Ravikiran Mane, Tushar Chouhan, and Cuntai Guan. BCI for stroke rehabilitation: motor and beyond. *Journal of neural engineering*, 17(4):041001, 2020.

[10] Maher Moakher. A Differential Geometric Approach to the Geometric Mean of Symmetric Positive-Definite Matrices. *SIAM J. Matrix Analysis Applications*, 26:735–747, 01 2005.

[11] C.S. Nam, A. Nijholt, and F. Lotte. *Brain–Computer Interfaces Handbook: Technological and Theoretical Advances*. CRC Press, 2018.

[12] L. F. Nicolas-Alonso and J. Gomez-Gil. Brain computer interfaces, a review. Sensors. *Sensors*, 12(2):1211–1279, 2012.

[13] Pedro Luiz Coelho Rodrigues, Christian Jutten, and Marco Congedo. Riemannian Procrustes Analysis: Transfer Learning for Brain–Computer Interfaces. *IEEE Transactions on Biomedical Engineering*, 66(8):2390–2401, 2019.

[14] Eduardo Santamaría-Vázquez, Víctor Martínez-Cagigal, Fernando Vaquerizo-Villar, and Roberto Hornero. EEG-Inception: A Novel Deep Convolutional Neural Network for Assistive ERP-Based Brain-Computer Interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 28(12):2773–2782, 2020.

[15] Robin Tibor Schirrmeister, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and Tonio Ball. Deep learning with convolutional neural networks for EEG decoding and visualization. *Human Brain Mapping*, 38(11):5391–5420, 2017.

[16] Daniel Wilson, Robin Tibor Schirrmeister, Lukas Alexander Wilhelm Gemein, and Tonio Ball. Deep Riemannian Networks for EEG Decoding, 2023.

[17] Xiaofeng Xie, Zou Xiaokun, Rongnian Tang, Yao Hou, and Feifei Qi. Multiple graph fusion based on Riemannian geometry for motor imagery classification. *Applied Intelligence*, 52, 2022.