Parallel Transport on the Cone Manifold of SPD Matrices for Domain Adaptation

Or Yair, *Student Member, IEEE*, Mirela Ben-Chen, and Ronen Talmon, *Member, IEEE*

***Abstract*—In this paper, we consider the problem of domain adaptation. We propose to view the data through the lens of covariance matrices and present a method for domain adaptation using parallel transport on the cone manifold of symmetric positive-definite matrices. We provide rigorous analysis using Riemannian geometry, illuminating the theoretical guarantees and benefits of the presented method. In addition, we demonstrate these benefits using experimental results on simulations and real-measured data.**

***Index Terms*—Positive definite matrices, domain adaptation, transfer learning, parallel transport.**

I. Introduction

THE increasing technological sophistication of current data acquisition systems gives rise to complex, multimodal datasets in high-dimension. As a result, the acquired data do not live in a Euclidean space, and applying analysis and learning algorithms directly to the data often leads to subpar performance.

To facilitate the analysis and processing of such data, one approach is to observe complex high-dimensional data through the lens of objects with a *known* non-Euclidean geometry. Notable examples of such objects are Symmetric and Positive Definite (SPD) matrices, which live on a cone manifold with a Riemannian metric. One of the most common forms of SPD matrices is a covariance matrix, which captures the linear relations between the different data coordinates. These relations are typically simple to compute, and therefore, recently, have become popular features in many applications in computer vision, medical imaging, and machine learning [1]–[4]. In particular, in [5], [6], the Riemannian geometry of covariance matrices was studied and exploited for medical imaging and physiological signal analysis.

Typically, Riemannian geometry is used to map objects from the non-Euclidean manifold to a linear Euclidean space by projection onto a tangent plane of the manifold. In existing work, the use of Riemannian geometry is usually limited to a single tangent plane. This indicates a hidden assumption that the SPD matrices corresponding to the data are confined to a local region of the manifold. However, the SPD matrices of the data often do not live in a small neighborhood on the manifold, and thus the resulting calculations may be inaccurate.

One such particular scenario, in which SPD matrices span a large portion of the cone manifold, occurs when the data comprise multiple domains corresponding to multiple sessions, subjects, batches, etc. For example, we will show that in a Brain-Computer-Interface (BCI) experiment, the covariance matrices of data acquired from a single subject in a specific session capture well the overall geometric structure of the data. Conversely, when the data consist of measurements from several subjects or several sessions, then the covariance matrices do not live in the same region of the manifold.

Often, multi-domain data pose significant challenges to learning approaches. For example, in the BCI experiment, it is challenging to train a classifier based on data from one subject (session) and apply it to data from another subject (session). This problem is largely referred to as *domain adaptation* or *transfer learning*, and it has attracted a significant research effort in recent years [7], [8].

Broadly, in domain adaptation, the main idea is to adapt a given model that is well performing on a particular domain, to a different yet related domain [7], [8]. Specifically, in the context of the cone manifold of SPD matrices, previous work proposed (geometric) mean subtraction as a simple method for domain adaption of BCI data [6]. Although this approach provided reasonable results for overcoming the differences between multiple sessions of a single subject, we show here that it fails to overcome the differences between multiple subjects. In [3], a Parallel Transport (PT) approach was proposed, which can be applied either directly to the data, or to a generative model of the data to reduce the computational load for large datasets. However, their approach considers a general Riemannian manifold. Since there is no closed-form expression of PT on Riemannian manifolds, besides the sphere manifold and the manifold of all SPD matrices [1], [4], no specific scheme or algorithm was provided. We note that PT can be approximated using Schild's Ladder [9], an approach that has been used extensively on the manifold of imaging data [4], [5], [9], [10].

In this paper, we propose a domain adaptation method using the analytic expression of PT on the cone manifold of SPD matrices. We claim that this is a natural and efficient solution for domain adaptation, which enjoys several important benefits. First, the solution is specially designed for SPD

Manuscript received August 2, 2018; revised December 2, 2018 and January 9, 2019; accepted January 15, 2019. Date of publication January 23, 2019; date of current version February 19, 2019. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Wenwu Wang. The work of O. Yair was supported in part by the Adams Fellowship Program of the Israel Academy of Sciences and Humanities and in part by the Israel Science Foundation (Grant 1490/16). The work of M. Ben-Chen was supported in part by the European Research Council (ERC Starting Grant 714776 OPREP) and in part by the Israel Science Foundation (Grant 504/16). The work of R. Talmon was supported by the Israel Science Foundation (Grant 1490/16). *(Corresponding author: Or Yair.)*

O. Yair and R. Talmon are with the Viterbi Faculty of Electrical Engineering, Technion-Israel Institute of Technology, Haifa 32000, Israel (e-mail: oryair@campus.technion.ac.il; ronen@ee.technion.ac.il).

M. Ben-Chen is with the Faculty of Computer Science, Technion-Israel Institute of Technology, Haifa 32000, Israel (e-mail: mirela@cs.technion.ac.il).

Digital Object Identifier 10.1109/TSP.2019.2894801

matrices, which have proven to be good features of data in a gamut of previous work [5], [6], [10]. Second, the analytic form of PT on the cone manifold circumvents approximations. Third, PT can be efficiently implemented, in contrast to the computationally demanding Schild's Ladder approximation. We establish the mathematical foundation of the proposed domain adaptation method. To this end, we provide new results in the geometry of SPD matrices. In addition, we show applications to both simulation and real recorded data, obtaining improved performance compared to the competing methods.

We note that an important requirement of the presented domain adaptation is that the covariance matrices are strictly positive. Indeed, positive semi-definite matrices lie at the boundary of the cone manifold of SPD matrices, and applying PT is not possible; technically, PT involves the inverse of SPD matrices, which does not exist if the matrices are only positive semi-definite. This requirement entails that a sufficient number of samples needs to be available, so that the covariance matrices are positive and can be accurately estimated. Specifically, if the number of samples over time is smaller than the dimension of the observations, the associated covariance matrices are low rank and only positive semi-definite. In order to relax this requirement and to promote strictly positive covariance matrices, some regularization could be considered, or alternatively, dimensionality reduction methods could be applied to preprocess the data.

Broadly, the use of positive semi-definite matrices may arise in many other situations as well. One prominent case is when positive semi-definite kernel functions are used as features instead of the covariance matrices. Using kernel functions could be highly beneficial since kernels can naturally accommodate nonlinear relationship between the data samples. In addition, in contrast to the covariance, kernel functions have the ability to capture the dynamics of time series.

In [11], an extension of the Riemannian framework to positive semi-definite matrices with a fixed rank was presented, which includes the corresponding Riemannian metric and the geodesic path. While this work allows the extension of some of the concepts presented in this paper, the main ingredient, PT, is still lacking and needs to be developed. This extends the scope of this paper and will be addressed in future work.

In parallel to our study, recent work [12] has proposed a scheme for transfer learning using the Riemannian geometry of SPD matrices, with a tight connection to the present work. We will show that the affine transformation proposed in [12] can be recast as PT. In this paper we provide the mathematical foundation to analyze this transport, we discuss the advantage of our solution compared to [12], and we point out the special case in which the two methods coincide.

In this paper we present an unsupervised approach for domain adaptation. Due to the lack of labeled data and of prior statistical models, the criterion of successful adaptation is not explicitly defined. As a result, the relatedness of two domains [13], that is the definition of a measure of how much two domains are related is challenging and is not provided in the present work. This is an important component in domain adaptation, since the adaptation of unrelated domains could lead to poor, or even 'negative', results. This problem will be the subject of future work, focusing on the statistical moments (or the principal directions) of the obtained joint representation. The alignment between the moments of the representation of two domains will facilitate the definition of relatedness and adaptation refinement.

This paper is organized as follows. In Section II, we present preliminaries on the Riemannian geometry of SPD matrices. In Section III, we formulate the problem, present the proposed domain adaptation method, and provide mathematical analysis and justification. Section IV shows experimental results on both simulation and real data. Finally, we conclude the paper in Section V.

II. Preliminaries on Riemannian Geometry of SPD Matrices

In this section we provide the preliminaries regarding SPD matrices, and we refer the reader to the book [14] for a detailed exposition of this topic. We note that in this paper we focus on covariance matrices, however the statements also hold for general SPD matrices. By definition, an SPD matrix $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ has only strictly positive eigenvalues. An alternative definition is that for any vector $\boldsymbol{v} \neq \boldsymbol{0}$ the quadratic form is strictly positive, i.e., $\boldsymbol{v}^T \boldsymbol{P}\boldsymbol{v} > 0$.

A. Metric and Distance

The definition of an SPD matrix entails that the collection of all SPD matrices constitutes a convex half-cone in the vector space of real $n \times n$ symmetric matrices. This cone forms a differentiable Riemannian manifold $\mathcal{M}$ equipped with the following inner product

$$\langle \boldsymbol{S}_1, \boldsymbol{S}_2 \rangle_{\mathcal{T}_{\boldsymbol{P}}\mathcal{M}} = \left\langle \boldsymbol{P}^{-\frac{1}{2}} \boldsymbol{S}_1 \boldsymbol{P}^{-\frac{1}{2}}, \boldsymbol{P}^{-\frac{1}{2}} \boldsymbol{S}_2 \boldsymbol{P}^{-\frac{1}{2}} \right\rangle \qquad (1)$$

where $\mathcal{T}_{\boldsymbol{P}}\mathcal{M}$ is the tangent space at the point $\boldsymbol{P} \in \mathcal{M}$, $\boldsymbol{S}_1, \boldsymbol{S}_2 \in \mathcal{T}_{\boldsymbol{P}}\mathcal{M}$, and $\langle \cdot, \cdot \rangle$ is the standard Euclidean inner product operation. The symmetric matrices $\boldsymbol{S} \in \mathcal{T}_{\boldsymbol{P}}\mathcal{M}$ in the tangent plane live in a linear space, and therefore, we can view them as vectors (with a proper representation). Throughout this paper, we interchangeably use the terms vectors and symmetric matrices when referring to $\boldsymbol{S} \in \mathcal{T}_{\boldsymbol{P}}\mathcal{M}$.

This Riemannian manifold is a Hadamard manifold, namely, it is simply connected and it is a complete Riemannian manifold of non-positive sectional curvature. Manifolds with non-positive curvature have a unique geodesic curve between any two points, a property that will later be exploited. Specifically, the unique geodesic curve between any two SPD matrices $\boldsymbol{P}_1, \boldsymbol{P}_2 \in \mathcal{M}$ is given by [14, Thm 6.1.6]

$$\varphi(t) = \boldsymbol{P}_1^{\frac{1}{2}} \left(\boldsymbol{P}_1^{-\frac{1}{2}} \boldsymbol{P}_2 \boldsymbol{P}_1^{-\frac{1}{2}} \right)^t \boldsymbol{P}_1^{\frac{1}{2}}, \quad 0 \leq t \leq 1 \qquad (2)$$

The arc-length of the geodesic curve defines the following Riemannian distance on the manifold [14]:

$$\begin{aligned} d_R^2(\boldsymbol{P}_1, \boldsymbol{P}_2) &= \left\| \log\left(\boldsymbol{P}_2^{-\frac{1}{2}} \boldsymbol{P}_1 \boldsymbol{P}_2^{-\frac{1}{2}} \right) \right\|_F^2 \\ &= \sum_{i=1}^{n} \log^2\left(\lambda_i \left(\boldsymbol{P}_2^{-\frac{1}{2}} \boldsymbol{P}_1 \boldsymbol{P}_2^{-\frac{1}{2}} \right) \right) \end{aligned}$$

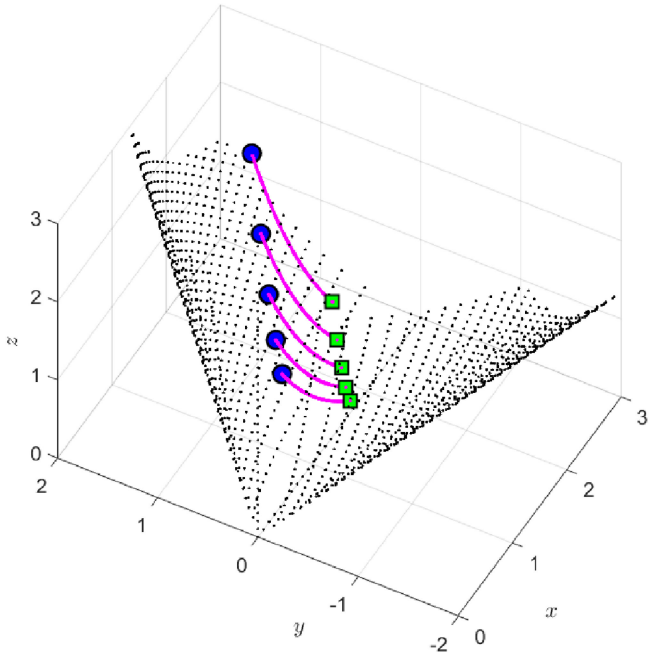

Fig. 1. The cone manifold of 2×2 SPD matrices. The black dots mark the boundary of the cone (i.e., matrices with eigenvalue zero). Each magenta curve is the geodesic between pairs of matrices (blue circles and green squares). All the geodesic curves are of the same length (i.e., the Riemannian distance between all the pairs is equal).

where $\boldsymbol{P}_1, \boldsymbol{P}_2 \in \mathcal{M}$, $\|\cdot\|_F$ is the Frobenius norm, $\log(\boldsymbol{P})$ is the matrix logarithm, and $\lambda_i(\boldsymbol{P})$ is the i-th eigenvalue of $\boldsymbol{P}$. We additionally denote by $\frac{d}{dt}\varphi(t) = \varphi'(t) \in \mathcal{T}_{\varphi(t)}\mathcal{M}$ the velocity vector of the geodesic at $t \in [0,1]$. Figure 1 presents an illustration of the geodesic curve and the Riemannian distance. The cone manifold of 2×2 SPD matrices can be displayed in $\mathbb{R}^3$, since any symmetric matrix $\boldsymbol{P} = \left(\begin{smallmatrix} x & y \\ y & z \end{smallmatrix}\right)$ is positive if and only if $x > 0, z > 0$ and $y^2 < xz$.

B. Exponential and Logarithm Maps

The Logarithm map, which projects an SPD matrix $\boldsymbol{P}_i \in \mathcal{M}$ to the tangent plane $\mathcal{T}_{\boldsymbol{P}}\mathcal{M}$ at $\boldsymbol{P} \in \mathcal{M}$, is given by

$$\boldsymbol{S}_i = \mathrm{Log}_{\boldsymbol{P}}(\boldsymbol{P}_i) = \boldsymbol{P}^{\frac{1}{2}} \log\left(\boldsymbol{P}^{-\frac{1}{2}} \boldsymbol{P}_i \boldsymbol{P}^{-\frac{1}{2}}\right) \boldsymbol{P}^{\frac{1}{2}} \in \mathcal{T}_{\boldsymbol{P}}\mathcal{M}$$

The Exponential map, which projects a vector $\boldsymbol{S}_i \in \mathcal{T}_{\boldsymbol{P}}\mathcal{M}$ back to the manifold $\mathcal{M}$ is given by

$$\boldsymbol{P}_i = \mathrm{Exp}_{\boldsymbol{P}}(\boldsymbol{S}_i) = \boldsymbol{P}^{\frac{1}{2}} \exp\left(\boldsymbol{P}^{-\frac{1}{2}} \boldsymbol{S}_i \boldsymbol{P}^{-\frac{1}{2}}\right) \boldsymbol{P}^{\frac{1}{2}} \in \mathcal{M} \quad (3)$$

An important property relates the Logarithm and Exponential maps to the geodesic curve. Formally, let $\boldsymbol{P}_1, \boldsymbol{P}_2 \in \mathcal{M}$, and consider the (unique) geodesic $\varphi(t)$ from $\boldsymbol{P}_1$ to $\boldsymbol{P}_2$. The initial velocity $\varphi'(0) \in \mathcal{T}_{\boldsymbol{P}_1}\mathcal{M}$ is given by the Logarithm map $\varphi'(0) = \mathrm{Log}_{\boldsymbol{P}_1}(\boldsymbol{P}_2)$. Similarly, the Exponential map projects the initial velocity vector $\varphi'(0)$ back to $\boldsymbol{P}_2$, namely, $\boldsymbol{P}_2 = \mathrm{Exp}_{\boldsymbol{P}_1}(\varphi'(0))$.

C. Riemannian Mean

The Riemannian mean $\overline{\boldsymbol{P}}$ of a set $\{\boldsymbol{P}_i | \boldsymbol{P}_i \in \mathcal{M}\}$ is defined using the Fréchet mean:

$$\overline{\boldsymbol{P}} \triangleq \arg\min_{\boldsymbol{P} \in \mathcal{M}} \sum_i d_R^2(\boldsymbol{P}, \boldsymbol{P}_i) \quad (4)$$

A special case is the Riemannian mean $\overline{\boldsymbol{P}}$ of two SPD matrices $\boldsymbol{P}_1, \boldsymbol{P}_2 \in \mathcal{M}$, which has a closed-form expression, and is located at the midpoint of the geodesic curve:

$$\overline{\boldsymbol{P}} = \varphi(\tfrac{1}{2}) = \boldsymbol{P}_1^{\frac{1}{2}} \left(\boldsymbol{P}_1^{-\frac{1}{2}} \boldsymbol{P}_2 \boldsymbol{P}_1^{-\frac{1}{2}}\right)^{\frac{1}{2}} \boldsymbol{P}_1^{\frac{1}{2}}$$

Generally, for more than two matrices, the solution of the optimization problem (4) can be obtained by an iterative procedure. Barachant *et al.* [6] presented an algorithm based on [15] for estimating the Riemannian mean. For completeness, we include their algorithm in Appendix F.

Given a set $\{\boldsymbol{P}_i | \boldsymbol{P}_i \in \mathcal{M}\}$ and its Riemannian mean $\overline{\boldsymbol{P}}$, there is a commonly used approximation of the Riemannian distances on $\mathcal{M}$ in the neighborhood of $\overline{\boldsymbol{P}}$. Specifically, the approximation of the Riemannian distance d_R^2 is given by:

$$d_R^2(\boldsymbol{P}_i, \boldsymbol{P}_j) \approx \left\|\tilde{\boldsymbol{S}}_i - \tilde{\boldsymbol{S}}_j\right\|_F^2 \quad (5)$$

where $\tilde{\boldsymbol{S}}_i = \overline{\boldsymbol{P}}^{-\frac{1}{2}} \mathrm{Log}_{\overline{\boldsymbol{P}}}(\boldsymbol{P}_i) \overline{\boldsymbol{P}}^{-\frac{1}{2}}$. For more details on the accuracy of this approximation, see [2].

III. Domain Adaptation With Parallel Transport

A. Overview

Let $\mathcal{X}^{(1)} = \{\boldsymbol{x}_i^{(1)}(t)\}_{i=1}^{N_1}$ and $\mathcal{X}^{(2)} = \{\boldsymbol{x}_i^{(2)}(t)\}_{i=1}^{N_2}$ be two subsets of N_1 and N_2 high-dimensional time series, respectively, where $\boldsymbol{x}_i^{(k)}(t) \in \mathbb{R}^D$. Suppose each subset lives in a particular domain, which could be related to the acquisition modality, session, deployment, and set of environmental conditions. In our notation, the superscript k denotes the index of the subset, the subscript i denotes the index of the time-series within each subset, and t represents the time axis of each time-series.

Our exposition focuses only on two subsets, and the generalization for any number of subsets is discussed at the end of this section. In addition, we consider here time-series, but our derivation does not take the temporal order into account, and therefore, the extension to other types of data, where t is merely a sample index, e.g., images, is straight-forward.

Analyzing such data typically raises many challenges. For example, a long-standing problem is how to efficiently compare between high-dimensional point clouds, and particularly, time-series. When the data are measured signals, sample comparisons become even more challenging, since such high-dimensional measured data usually contain high levels of noise.

In particular, in our setting, we face an additional challenge, since the data is given in different domains; comparing time-series from the same subset is a difficult task by itself, even more so is comparing time-series from two subsets from different domains.

Our goal is to find a new *joint* representation of the two subsets in an unsupervised manner. Broadly, we aim to devise a low-dimensional representation in a Euclidean space that facilitates efficient and meaningful comparisons. For the purpose of evaluation, we associate the time-series $\boldsymbol{x}_i^{(k)}(t)$ with labels $y_i^{(k)}$ and define "meaningful" comparisons with respect to these labels. More concretely, we evaluate the joint representation by the Euclidean distance between the new representation of any two time-series with similar corresponding labels, independently of the time-series respective domain. We emphasize that the

proposed approach is unsupervised and it does not depend on the labels, which are only considered for the purpose of evaluation.

Devising such a new representation will facilitate efficient and accurate domain adaptation schemes. Specifically, given a subset $\{\boldsymbol{x}_i^{(1)}(t)\}_{i=1}^{N_1}$ with corresponding labels $\{y_i^{(1)}\}_{i=1}^{N_1}$, we could train a classifier based on the new derived representation of the subset. Then, when another unlabeled subset $\{\boldsymbol{x}_i^{(2)}(t)\}_{i=1}^{N_2}$ becomes available, we could apply the trained classifier to the derived (joint) representation of the latter subset.

B. Illustrative Example

To put the problem setting and our proposed solution in context, throughout the paper, we will follow an illustrative example, taken from the brain computer interface (BCI) competition IV (dataset IIa) [16]. Consider data from a BCI experiment of motor imagery comprising of recordings from $D = 22$ Electroencephalography (EEG) electrodes. The dataset contains several subjects, where each subject was asked repeatedly to perform one out of four motor imagery tasks: movement of the right hand, the left hand, both feet, and the tongue.

Let $\mathcal{X}^{(1)} = \{\boldsymbol{x}_i^{(1)}(t)\}_{i=1}^{N_1}$ be a subset of recordings acquired from a single subject, indexed (1), where the time-series $\boldsymbol{x}_i^{(1)}(t)$ consists of the signals, recorded simultaneously from the D EEG channels during the i-th repetition/trial. Each time series $\boldsymbol{x}_i^{(1)}(t)$ is attached with a label $y_i^{(1)}$, denoting the imagery task performed at the the i-th trial. Common practice is to train a classifier based on $\mathcal{X}^{(1)}$, so that the imagery task could be identified from new EEG recordings. This capability could then be the basis for devising brain computer interfaces, for example, to control prosthetics.

Suppose a new subset $\mathcal{X}^{(2)} = \{\boldsymbol{x}_i^{(2)}(t)\}_{i=1}^{N_2}$ of recordings acquired from another subject, indexed (2), becomes available. Applying the classifier, trained based on data from subject (1), to the new subset of recordings from subject (2) yields poor results, as we will demonstrate in Section IV-B. Indeed, most methods addressing this particular challenge, as well as related problems, exclusively analyze data from each individual subject separately. By constructing a joint representation for both $\mathcal{X}^{(1)}$ and $\mathcal{X}^{(2)}$, which is oblivious to the specific subject, we develop a classifier that is trained on data from one subject and applied to data from another subject without any calibration, i.e., without any labeled data from the new (test) subject.

C. Covariance Matrices as Data Features

As described before, we suggest looking at the data through the lens of covariance matrices. We denote the covariance matrices by:

$$\boldsymbol{P}_i^{(k)} = \mathbb{E}\left[\left(\boldsymbol{x}_i^{(k)}(t) - \boldsymbol{\mu}_i^{(k)}\right)\left(\boldsymbol{x}_i^{(k)}(t) - \boldsymbol{\mu}_i^{(k)}\right)^T\right]$$

where $\boldsymbol{\mu}_i^{(k)} = \mathbb{E}[\boldsymbol{x}_i^{(k)}(t)]$. Typically, since the statistics of the data is unknown, we use estimates of the covariance, such as the sample covariance. We note that our approach is applicable to any kind of input data given as SPD matrices. For example, in machine learning, common practice is to use kernels which represent an inner product between features after some nonlinear transformation [17].

By using covariance matrices as data features we enjoy a few key benefits. First, since covariance matrices are computed from data by averaging over time, they tend to be robust to noise. Second, covariance matrices can be seen as a low dimensional representation. Third, they have useful geometric properties and a well-developed Riemannian framework, as described in Section II. Particularly, they have a Riemannian metric (3), facilitating appropriate data samples comparisons, which is a basic ingredient of many analysis and learning techniques. In this work, we build on and extend the latter.

Recently, the usefulness of covariance matrices has been demonstrated in the context of the BCI problem [6]. There, Barachant et al. considered data from a single subject and proposed to project the covariance matrices $\{\boldsymbol{P}_i\}$ of the recordings from each trial (after some whitening) into the tangent plane of the Riemannian mean $\overline{\boldsymbol{P}}$, namely compute $\boldsymbol{S}_i = \mathrm{Log}_{\overline{\boldsymbol{P}}}(\boldsymbol{P}_i)$. Then, a classifier was trained on the set $\{\boldsymbol{S}_i\}$. Using this approach, state of the art results for motor imagery task classification were obtained. However, when considering several subsets from multiple domains, such as different sessions or subjects, as reported in [6], the covariance matrices convey a domain-specific content, which in turn poses limitations on task classification. For multiple sessions on different days, Barachant *et al.* proposed to subtract the Riemannian mean from each subset, namely, to project each subset $\mathcal{P}^{(k)} = \{\boldsymbol{P}_i^{(k)}\}$ to the tangent space at its own mean. Indeed, when the train set and the test set were obtained on different days, this mean normalization improved the task classification rate. However, in the case of multiple subjects, this approach is inadequate. As mentioned before, given recordings from one subject as a train set and recordings from another subject as a test set, the classification of the different mental tasks based on covariance matrices fails completely.

This illuminates the primary challenge addressed in this work – how to build a representation so that any two covariance matrices associated with the same mental task, but from possibly different sessions or subjects, will be given a similar representation. Importantly, since the task labels are unknown, this objective cannot be directly imposed. In the sequel, we exploit the Riemannian geometry of covariance matrices, and devise such a representation in an unsupervised manner by preserving local geometric structures.

D. Formulation

Consider two subsets $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$ from two different domains consisting of N_1 and N_2 covariance matrices, respectively. Let $\overline{\boldsymbol{P}}^{(1)}$ and $\overline{\boldsymbol{P}}^{(2)}$ be their respective Riemannian means. Let $\varphi(t)$, given explicitly in (2), denote the unique geodesic from $\overline{\boldsymbol{P}}^{(2)}$ to $\overline{\boldsymbol{P}}^{(1)}$ such that $\varphi(0) = \overline{\boldsymbol{P}}^{(2)}$ and $\varphi(1) = \overline{\boldsymbol{P}}^{(1)}$. Finally, let $\boldsymbol{S}_i^{(k)}$ be the symmetric matrix (or equivalently, the vector) in the tangent space $\mathcal{T}_{\overline{\boldsymbol{P}}^{(k)}}\mathcal{M}$, obtained by projecting $\boldsymbol{P}_i^{(k)}$ to $\mathcal{T}_{\overline{\boldsymbol{P}}^{(k)}}\mathcal{M}$:

$$\boldsymbol{S}_i^{(k)} = \mathrm{Log}_{\overline{\boldsymbol{P}}^{(k)}}\left(\boldsymbol{P}_i^{(k)}\right)$$

for $k \in \{1,2\}$ and $i \in \{1,2,\ldots,N_k\}$.

Our goal now is to derive a new representation $\Gamma(\boldsymbol{S}_i^{(2)})$ of $\boldsymbol{S}_i^{(2)}$ given by the map $\Gamma : \mathcal{T}_{\overline{\boldsymbol{P}}^{(2)}}\mathcal{M} \to \mathcal{T}_{\overline{\boldsymbol{P}}^{(1)}}\mathcal{M}$, such that $\{\boldsymbol{S}_i^{(1)}\}$ and $\{\Gamma(\boldsymbol{S}_i^{(2)})\}$ live in the same space. This allows us to relate samples from the two subsets, and compute quantities such as $\langle \boldsymbol{S}_i^{(1)}, \Gamma(\boldsymbol{S}_j^{(2)})\rangle_{\overline{\boldsymbol{P}}^{(1)}}$. In addition, we require that the new representation will fulfill the following properties:

1) Zero mean:
$$\tfrac{1}{N_2}\sum_{i=1}^{N_2}\Gamma\left(\boldsymbol{S}_i^{(2)}\right) = \tfrac{1}{N_1}\sum_{i=1}^{N_1}\boldsymbol{S}_i^{(1)} = 0$$
2) Inner product preservation:
$$\left\langle \Gamma\left(\boldsymbol{S}_i^{(2)}\right), \Gamma\left(\boldsymbol{S}_j^{(2)}\right)\right\rangle_{\overline{\boldsymbol{P}}^{(1)}} = \left\langle \boldsymbol{S}_i^{(2)}, \boldsymbol{S}_j^{(2)}\right\rangle_{\overline{\boldsymbol{P}}^{(2)}}$$
for all $i,j \in \{1,\ldots,N_2\}$.
3) Geodesic velocity preservation:
$$\Gamma\left(\varphi'(0)\right) = \varphi'(1) \tag{6}$$

Properties (1) and (2) imply that the new representation Γ preserves inter-sample relations, defined by the inner product. Note that a map Γ satisfying properties (1) and (2) is not unique; for any Γ admitting to properties (1) and (2), the composition $R \circ \Gamma$, where R is an arbitrary rotation within the subspace $\mathcal{T}_{\overline{\boldsymbol{P}}^{(1)}}\mathcal{M}$, satisfies properties (1) and (2) as well. To resolve this arbitrary degree of freedom, we use the geodesic between two points on the SPD manifold, which is unique [14]. Concretely, in property (3), the two intrinsic symmetric matrices (vectors) $\varphi'(0) \in \mathcal{T}_{\overline{\boldsymbol{P}}^{(2)}}\mathcal{M}$ and $\varphi'(1) \in \mathcal{T}_{\overline{\boldsymbol{P}}^{(1)}}\mathcal{M}$, induced by the velocity of the unique geodesic at the source and destination, are used to fix a rotation and to align the subset $\{\Gamma(\boldsymbol{S}_i^{(2)})\}$ with the subset $\{\boldsymbol{S}_i^{(1)}\}$.

We remark that the above properties imply that the subset $\{\Gamma(\boldsymbol{S}_i^{(2)})\}$ is embedded in the $\langle\cdot,\cdot\rangle_{\overline{\boldsymbol{P}}^{(1)}}$ inner product space. In the sequel, we will describe how to circumvent the dependence of the inner product space on $\overline{\boldsymbol{P}}^{(1)}$ and make the new representation truly Euclidean by pre-whitening the data. Additionally, note that the mean subtraction presented in [6] admits only properties (1) (2).

E. Domain Adaptation

First, we explicitly provide the expression for parallel transport on the SPD cone manifold, and then we use it to define the map Γ.

Lemma 1 (Parallel Transport): Let $\boldsymbol{A},\boldsymbol{B} \in \mathcal{M}$. The PT from $\boldsymbol{B}$ to $\boldsymbol{A}$ of any $\boldsymbol{S} \in \mathcal{T}_{\boldsymbol{B}}\mathcal{M}$ is given by:
$$\Gamma_{\boldsymbol{B}\to\boldsymbol{A}}(\boldsymbol{S}) \triangleq \boldsymbol{E}\boldsymbol{S}\boldsymbol{E}^T \tag{7}$$
where $\boldsymbol{E} = (\boldsymbol{A}\boldsymbol{B}^{-1})^{\frac{1}{2}}$.

This lemma was presented in [1, Eq. 3.4]. The proof of the lemma is given in Appendix A and it is based on [18]. An illustration of the PT on the SPD manifold is presented in Figure 2. Note that the inner products between the three vectors in the figure are preserved under the parallel transport $\Gamma_{\boldsymbol{B}\to\boldsymbol{A}}$ and the appearance could be misleading since the space is not Euclidean.

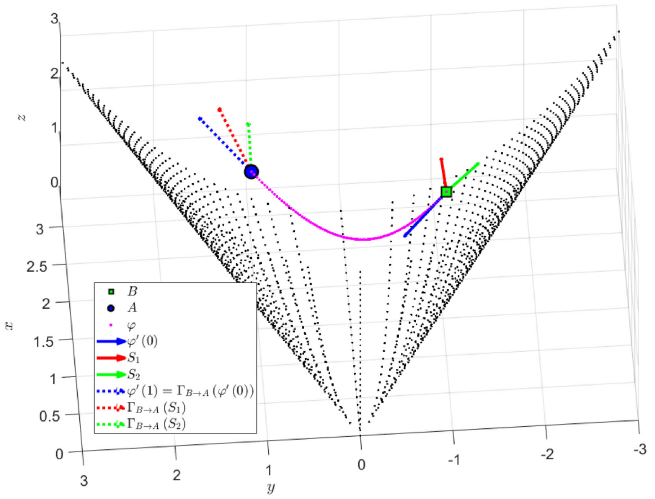

Fig. 2. Illustration of the PT on the SPD manifold. $\boldsymbol{A}$ and $\boldsymbol{B}$ are two SPD matrices, and φ is the unique geodesic between them. We plot three vectors in $\mathcal{T}_{\boldsymbol{B}}\mathcal{M}$: $\varphi'(0)$, S_1 and S_2 along with their corresponding parallel transported vectors to $\mathcal{T}_{\boldsymbol{A}}\mathcal{M}$ using $\Gamma_{\boldsymbol{B}\to\boldsymbol{A}}$.

Theorem 1: The representation $\Gamma_{\overline{\boldsymbol{P}}^{(2)}\to\overline{\boldsymbol{P}}^{(1)}}(\boldsymbol{S}_i^{(2)})$ given by the unique PT of $\boldsymbol{S}_i^{(2)}$ from $\overline{\boldsymbol{P}}^{(2)}$ to $\overline{\boldsymbol{P}}^{(1)}$ is well defined and satisfies properties (1)–(3).

The proof is given in Appendix B.

Theorem 1 sets the stage for domain adaptation. We propose a map $\Psi : \mathcal{M} \to \mathcal{M}$ that adapts the domain of the subset of SPD matrices $\mathcal{P}^{(2)}$ to the domain of the subset $\mathcal{P}^{(1)}$. For any $\boldsymbol{P}_i^{(2)} \in \mathcal{P}^{(2)}$, the map $\Psi(\boldsymbol{P}_i^{(2)})$ is given by
$$\Psi\left(\boldsymbol{P}_i^{(2)}\right) = \mathrm{Exp}_{\overline{\boldsymbol{P}}^{(1)}}\left(\Gamma_{\overline{\boldsymbol{P}}^{(2)}\to\overline{\boldsymbol{P}}^{(1)}}\left(\mathrm{Log}_{\overline{\boldsymbol{P}}^{(2)}}\left(\boldsymbol{P}_i^{(2)}\right)\right)\right) \tag{8}$$
To enhance the geometric insight, we explicitly describe the three steps comprising the construction of Ψ:

1) Project the SPD matrix $\boldsymbol{P}_i^{(2)}$ to the tangent plane $\mathcal{T}_{\overline{\boldsymbol{P}}^{(2)}}\mathcal{M}$ by $\boldsymbol{S}_i^{(2)} = \mathrm{Log}_{\overline{\boldsymbol{P}}^{(2)}}(\boldsymbol{P}_i^{(2)})$.
2) Parallel transport $\boldsymbol{S}_i^{(2)}$ from $\overline{\boldsymbol{P}}^{(2)}$ to $\overline{\boldsymbol{P}}^{(1)}$ by computing $\boldsymbol{S}_i^{(2)\to(1)} = \Gamma_{\overline{\boldsymbol{P}}^{(2)}\to\overline{\boldsymbol{P}}^{(1)}}(\boldsymbol{S}_i^{(2)})$.
3) Project the symmetric matrix $\boldsymbol{S}_i^{(2)\to(1)} \in \mathcal{T}_{\overline{\boldsymbol{P}}^{(1)}}\mathcal{M}$ back to the manifold using $\mathrm{Exp}_{\overline{\boldsymbol{P}}^{(1)}}(\boldsymbol{S}_i^{(2)\to(1)})$.

The implementation of Ψ can be simplified and made more efficient by using the following theorem.

Theorem 2: Let $\boldsymbol{A},\boldsymbol{B},\boldsymbol{P} \in \mathcal{M}$ and let $\boldsymbol{S} = \mathrm{Log}_{\boldsymbol{B}}(\boldsymbol{P}) \in \mathcal{T}_{\boldsymbol{B}}\mathcal{M}$. Then,
$$\mathrm{Exp}_{\boldsymbol{A}}\left(\Gamma_{\boldsymbol{B}\to\boldsymbol{A}}(\boldsymbol{S})\right) = \boldsymbol{E}\boldsymbol{P}\boldsymbol{E}^T$$
where $\boldsymbol{E} = (\boldsymbol{A}\boldsymbol{B}^{-1})^{\frac{1}{2}}$.

In words, the "parallel transport" of an SPD matrix $\boldsymbol{P} \in \mathcal{M}$ from $\boldsymbol{B}$ to $\boldsymbol{A}$ is given the same transformation applied to $\boldsymbol{S} = \mathrm{Log}_{\boldsymbol{B}}(\boldsymbol{P})$. Namely, the "parallel transport" of the SPD matrix $\boldsymbol{P}$ from $\boldsymbol{B}$ to $\boldsymbol{A}$ is equal to projecting $\boldsymbol{P}$ to the tangent plane at $\boldsymbol{B}$, parallel transporting the projection to the tangent plane at $\boldsymbol{A}$, and then projecting back to the SPD manifold. As a consequence, we show in the sequel that the map Ψ in (8) can be written simply in terms of Γ. The proof of Theorem 2 is given in Appendix C. We note that we present the theorem in a general context, since

we did not find such a result in the literature and believe it might be of independent interest.

Theorem 2 enables us to efficiently compute $\Psi(\boldsymbol{P}_i^{(2)})$, since it circumvents the computation of the Logarithm and Exponential maps of the SPD matrix in steps 1 and 3 above. Instead, the transformation defined by $\boldsymbol{E}$ is computed only once for the entire set, and (8) can be recast as:

$$\Psi\left(\boldsymbol{P}_i^{(2)}\right) = \Gamma_{\overline{\boldsymbol{P}}^{(2)}\to\overline{\boldsymbol{P}}^{(1)}}\left(\boldsymbol{P}_i^{(2)}\right) = \boldsymbol{E}\boldsymbol{P}_i^{(2)}\boldsymbol{E}^T \tag{9}$$

where $\boldsymbol{E} \triangleq (\overline{\boldsymbol{P}}^{(1)}(\overline{\boldsymbol{P}}^{(2)})^{-1})^{\frac{1}{2}}$. Note that this equality is well defined since any tangent plane to the SPD manifold $\mathcal{M}$ is the entire space of symmetric matrices [18].

Thus far in the exposition, only the uniqueness of the geodesic curve on the manifold of SPD matrices was exploited, such that the PT along the geodesic admits the property in (6), namely: $\Gamma(\varphi'(0)) = \varphi'(1)$. Importantly, PT specifically along the unique geodesic curve exhibits important invariance to the "relative" location on the manifold.

Definition 1 (Equivalent Pairs): Two pairs $(\boldsymbol{A}_1, \boldsymbol{B}_1)$ and $(\boldsymbol{A}_2, \boldsymbol{B}_2)$, such that $\boldsymbol{A}_1, \boldsymbol{B}_1, \boldsymbol{A}_2, \boldsymbol{B}_2 \in \mathcal{M}$, are *equivalent* if there exists an invertible matrix $\boldsymbol{E}$ such that

$$\boldsymbol{A}_2 = \Gamma(\boldsymbol{A}_1) = \boldsymbol{E}\boldsymbol{A}_1\boldsymbol{E}^T$$

$$\boldsymbol{B}_2 = \Gamma(\boldsymbol{B}_1) = \boldsymbol{E}\boldsymbol{B}_1\boldsymbol{E}^T$$

We denote this relation by

$$(\boldsymbol{A}_1, \boldsymbol{B}_1) \sim (\boldsymbol{A}_2, \boldsymbol{B}_2)$$

Lemma 2: The relation $\sim$ is an equivalence relation.

The proof is straight-forward as we show in the following.

- *Reflexivity* is satisfied by setting $\boldsymbol{E}$ to be the identity matrix.
- *Symmetry*: if $\boldsymbol{A}_2 = \boldsymbol{E}\boldsymbol{A}_1\boldsymbol{E}^T$ then $\boldsymbol{A}_1 = \boldsymbol{E}^{-1}\boldsymbol{A}_2\boldsymbol{E}^{-T}$ and analogously for $\boldsymbol{B}_1, \boldsymbol{B}_2$.
- *Transitivity*: if $\boldsymbol{A}_2 = \boldsymbol{E}_1\boldsymbol{A}_1\boldsymbol{E}_1^T$ and $\boldsymbol{A}_3 = \boldsymbol{E}_2\boldsymbol{A}_2\boldsymbol{E}_2^T$ then $\boldsymbol{A}_3 = \boldsymbol{E}\boldsymbol{A}_1\boldsymbol{E}^T$ where $\boldsymbol{E} = \boldsymbol{E}_2\boldsymbol{E}_1$ and analogously for $\boldsymbol{B}_1, \boldsymbol{B}_2$.

In other words, two pairs are equivalent if the relation of the two matrices in the pair is given by the same transformation Γ. We interpret such equivalent pairs as matrices with equivalent intra-relations (e.g., if $(\boldsymbol{A}_1, \boldsymbol{B}_1) \sim (\boldsymbol{A}_2, \boldsymbol{B}_2)$, then $d_R(\boldsymbol{A}_1, \boldsymbol{B}_1) = d_R(\boldsymbol{A}_2, \boldsymbol{B}_2)$), but with a different global position on the manifold. For example, each two pairs in Figure 1 are equivalent pairs.

Proposition 1: Let $(\boldsymbol{A}_1, \boldsymbol{B}_1)$ be a pair of SPD matrices $\boldsymbol{A}_1, \boldsymbol{B}_1 \in \mathcal{M}$, and let $[(\boldsymbol{A}_1, \boldsymbol{B}_1)]$ denote the equivalence class $[(\boldsymbol{A}_1, \boldsymbol{B}_1)] = \{(\boldsymbol{A}_2, \boldsymbol{B}_2) \in \mathcal{M} \times \mathcal{M} | (\boldsymbol{A}_2, \boldsymbol{B}_2) \sim (\boldsymbol{A}_1, \boldsymbol{B}_1)\}$, of all matrix pairs that are equivalent to $(\boldsymbol{A}_1, \boldsymbol{B}_1)$. Then, for any $(\boldsymbol{A}_2, \boldsymbol{B}_2) \in [(\boldsymbol{A}_1, \boldsymbol{B}_1)]$:

$$\Gamma \circ \Gamma_{\boldsymbol{B}_1\to\boldsymbol{A}_1} = \Gamma_{\boldsymbol{B}_2\to\boldsymbol{A}_2} \circ \Gamma$$

where $\Gamma(\boldsymbol{P}) = \boldsymbol{E}\boldsymbol{P}\boldsymbol{E}^T$ and $\boldsymbol{E}$ is the transformation defined in Definition 1.

The proof is given in Appendix D. Note that in this context, we restrict the operator Γ to the SPD manifold. This restriction is based on Theorem 2 and is well-defined since the tangent plane is the entire space of symmetric matrices and therefore contains the manifold.

An immediate consequence of Proposition 1 is that the domain adaptation via the representation Ψ is invariant to the relative position of $\overline{\boldsymbol{P}}^{(1)}$ and $\overline{\boldsymbol{P}}^{(2)}$ on the manifold, and is constructed equivalently for every pair in the equivalence class $[(\overline{\boldsymbol{P}}^{(1)}, \overline{\boldsymbol{P}}^{(2)})]$.

To demonstrate the importance of the property above, we revisit the illustrating BCI problem. Suppose $(\overline{\boldsymbol{P}}_{A1}, \overline{\boldsymbol{P}}_{B1})$ are the Riemannian means of the covariance matrices of Subject A and Subject B recorded in Session 1, and suppose $(\overline{\boldsymbol{P}}_{A2}, \overline{\boldsymbol{P}}_{B2})$ are the Riemannian means of the covariance matrices of Subject A and Subject B recorded in Session 2. If $(\overline{\boldsymbol{P}}_{A1}, \overline{\boldsymbol{P}}_{B1}) \sim (\overline{\boldsymbol{P}}_{A2}, \overline{\boldsymbol{P}}_{B2})$, then there exists a transformation Γ such that Γ encodes the relation between Session 1 and Session 2 whereas the relation of the two subjects is encoded by $\Gamma_{\overline{\boldsymbol{P}}_{B1}\to\overline{\boldsymbol{P}}_{A1}}$ or by $\Gamma_{\overline{\boldsymbol{P}}_{B2}\to\overline{\boldsymbol{P}}_{A2}}$, depending on the session. Proposition 1 guarantees the consistence of the relation between Subject A and Subject B. Namely, the Riemannian mean of Subject B in Session 2 can be related to the Riemannian mean of Subject A in Session 1 using the relation between the sessions (given by Γ) and the relation between the two subjects (given either by $\Gamma_{\overline{\boldsymbol{P}}_{B1}\to\overline{\boldsymbol{P}}_{A1}}$ or by $\Gamma_{\overline{\boldsymbol{P}}_{B2}\to\overline{\boldsymbol{P}}_{A2}}$), independently of the relative location of the means on the manifold.

F. Extension to K Subsets

Overall, by Theorem 2, for a general number of subsets $K \geq 2$, we can apply PT using Ψ (9) directly to the SPD matrices $\mathcal{P}^{(k)} = \{\boldsymbol{P}_i^{(k)}\}$ without projections to and from the tangent plane. Let $\hat{\boldsymbol{P}}$ denote the Riemannian mean of Riemannian means (centroids) $\{\overline{\boldsymbol{P}}^{(k)}\}_{k=1}^K$ of the subsets, namely,

$$\hat{\boldsymbol{P}} = \arg\min_{\boldsymbol{P}} \sum_{k=1}^{K} d_R^2\left(\boldsymbol{P}, \overline{\boldsymbol{P}}^{(k)}\right)$$

Each subset $\mathcal{P}^{(k)}$ is then parallel transported from its corresponding centroid $\overline{\boldsymbol{P}}^{(k)}$ to $\hat{\boldsymbol{P}}$. Formally, let $\boldsymbol{\Gamma}_i^{(k)}$ denote $\boldsymbol{P}_i^{(k)}$ after applying PT, which is given by

$$\boldsymbol{\Gamma}_i^{(k)} = \Gamma_{\overline{\boldsymbol{P}}^{(k)}\to\hat{\boldsymbol{P}}}\left(\boldsymbol{P}_i^{(k)}\right), \quad \forall i,\ k$$

and let $\tilde{\boldsymbol{S}}_i^{(k)}$ be the projection of $\boldsymbol{\Gamma}_i^{(k)}$ to the Euclidean tangent space (5):

$$\tilde{\boldsymbol{S}}_i^{(k)} = \log\left(\hat{\boldsymbol{P}}^{-\frac{1}{2}}\boldsymbol{\Gamma}_i^{(k)}\hat{\boldsymbol{P}}^{-\frac{1}{2}}\right)$$

This projection, which is further discussed in [6], can be interpreted as (i) data whitening by $\tilde{\boldsymbol{\Gamma}}_i^{(k)} = \hat{\boldsymbol{P}}^{-\frac{1}{2}}\boldsymbol{\Gamma}_i^{(k)}\hat{\boldsymbol{P}}^{-\frac{1}{2}}$, and (ii) projection to $\mathcal{T}_{\boldsymbol{I}}\mathcal{M}$ where $\boldsymbol{I}$ is the identity matrix. The projected symmetric matrices (vectors) $\tilde{\boldsymbol{S}}_i^{(k)}$ indeed reside in a Euclidean space. The proposed algorithm is given in Algorithm 1.

We conclude this section with two remarks. First, since the matrices $\tilde{\boldsymbol{S}}_i^{(k)}$ are symmetric, only their upper (or lower) triangular part with a gain factor of $\sqrt{2}$ applied to all non-diagonal elements could be taken into account. Second, alternative choices of $\hat{\boldsymbol{P}}$ could also be used, for example, the identity matrix. Indeed, recently [12] proposed to align datasets for transfer learning in a similar context using the identity matrix as $\hat{\boldsymbol{P}}$. However in [12], the alignment appeared as an empirical affine transformation, whereas in this work, we provide the geometrical justification and rigorous mathematical analysis. In case of two domains,

Algorithm 1: Domain Adaptation Using Parallel Transport for SPD Matrices.

Input: $\{\boldsymbol{P}_i^{(1)}\}_{i=1}^{N_1}, \{\boldsymbol{P}_i^{(2)}\}_{i=1}^{N_2}, \ldots, \{\boldsymbol{P}_i^{(K)}\}_{i=1}^{N_K}$ where $\boldsymbol{P}_i^{(k)}$ is the SPD matrix associated with the i-th element (e.g., high-dimensional time-series) in the k-th subset.

Output: $\{\tilde{\boldsymbol{S}}_i^{(1)}\}_{i=1}^{N_1}, \{\tilde{\boldsymbol{S}}_i^{(2)}\}_{i=1}^{N_2}, \ldots, \{\tilde{\boldsymbol{S}}_i^{(K)}\}_{i=1}^{N_K}$ where $\tilde{\boldsymbol{S}}_i^{(k)}$ is the new representation of $\boldsymbol{P}_i^{(k)}$ in a Euclidean space.

1) **For** each $k \in \{1, 2, \ldots, K\}$, compute $\overline{\boldsymbol{P}}^{(k)}$ the Riemannian mean of the subset $\{\boldsymbol{P}_i^{(k)}\}$.
2) Compute $\hat{\boldsymbol{P}}$, the Riemannian mean of $\{\overline{\boldsymbol{P}}^{(k)}\}_{k=1}^{K}$.
3) **For** all k and all i, apply Parallel Transport using (7):
$$\boldsymbol{\Gamma}_i^{(k)} = \Gamma_{\overline{\boldsymbol{P}}^{(k)} \to \hat{\boldsymbol{P}}}\left(\boldsymbol{P}_i^{(k)}\right).$$
4) **For** all k and all i, project the transported matrix to the tangent space via:
$$\tilde{\boldsymbol{S}}_i^{(k)} = \log\left(\hat{\boldsymbol{P}}^{-\frac{1}{2}} \boldsymbol{\Gamma}_i^{(k)} \hat{\boldsymbol{P}}^{-\frac{1}{2}}\right).$$

$\hat{\boldsymbol{P}}$ can be any point on the geodesic between the centroids of the two domains. Setting such target points leads to the same representation.

In case of more than two domains, we set $\hat{\boldsymbol{P}}$ to be the mean of the centroids for two reasons. First, applying PT to the mean point leads to the minimal transportation along the SPD cone manifold. Second, this choice is invariant to the sizes of subsets, and therefore, can better support unbalanced datasets.

We remark that, in the case of two subsets with means $\overline{\boldsymbol{P}}_A \in \mathcal{M}$ and $\overline{\boldsymbol{P}}_B \in \mathcal{M}$, the affine transformation presented in [12] can be interpreted as two consecutive applications of PT: from $\overline{\boldsymbol{P}}_B$ to $\boldsymbol{I}$ and then from $\boldsymbol{I}$ to $\overline{\boldsymbol{P}}_A$. The arbitrary choice of $\boldsymbol{I}$ as an intermediate point introduces dependence of the algorithm on the global position on the manifold. Indeed, such a procedure, which can be expressed by $\Gamma_{\boldsymbol{I} \to \overline{\boldsymbol{P}}_A} \circ \Gamma_{\overline{\boldsymbol{P}}_B \to \boldsymbol{I}}$ does not admit the invariance property specified in Proposition 1.

Interestingly, the method proposed in [12] coincides with the present work, namely, $\Gamma_{\overline{\boldsymbol{P}}_B \to \overline{\boldsymbol{P}}_A} = \Gamma_{\boldsymbol{I} \to \overline{\boldsymbol{P}}_A} \circ \Gamma_{\overline{\boldsymbol{P}}_B \to \boldsymbol{I}}$ when the identity matrix $\boldsymbol{I}$ is on the geodesic φ between $\overline{\boldsymbol{P}}_B$ and $\overline{\boldsymbol{P}}_A$. In this case, the matrices $\overline{\boldsymbol{P}}_A$ and $\overline{\boldsymbol{P}}_B$ commute and they have the same eigenvectors (see Appendix E). From a data analysis perspective, when $\overline{\boldsymbol{P}}_A$ and $\overline{\boldsymbol{P}}_B$ are two covariance matrices of two subsets, this implies that the subsets have the same principal components.

We set $\hat{\boldsymbol{P}}$ as the Riemannian mean of the centroids so that the overall transport applied to the covariance matrices is minimal. This choice is motivated by the assumption that transporting accumulates distortions. This is a straight-forward generalization of the two subsets case, where the parallel transport is carried out along the shortest path (unique geodesic curve).

IV. Experimental Results

In this section we show the results of Algorithm 1 for both a synthetic example and for real data. The code for the synthetic example is available online at https://github.com/oryair/ParallelTransportSPDManifold.

A. Toy Problem

We generate time series in $\mathbb{R}^2$, so that their covariance matrices are in $\mathbb{R}^{2\times 2}$. Since the covariance matrices are symmetric, this particular choice enables us to visualize them in $\mathbb{R}^3$. Concretely, any 2×2 symmetric matrix $\boldsymbol{A} = \left(\begin{smallmatrix} x & y \\ y & z \end{smallmatrix}\right)$ can be visualized in $\mathbb{R}^3$ using $(x, y, z) \in \mathbb{R}^3$. $\boldsymbol{A}$ is positive-definite if and only if: $x, z > 0$ and $y^2 < xz$. These conditions establish the cone manifold of 2×2 SPD matrices.

Consider the set of hidden multi-dimensional times series $\{\boldsymbol{s}_i[n]\}_{i=1}^{100}$, given by:

$$\boldsymbol{s}_i[n] = \begin{bmatrix} \sin(2\pi f_0 n/T) \\ \cos(2\pi f_0 n/T + \phi_i) \end{bmatrix}, \quad n = 0, \ldots, T-1$$

where $f_0 = 10$, $T = 500$, and ϕ_i is uniformly drawn from $[-\pi/2, 0]$. Namely, each time-series $\boldsymbol{s}_i[n]$ consists of two oscillatory signals and is governed by a 1-dimensional hidden variable ϕ_i, the initial phase of the oscillations. Indeed, the population covariance of $\boldsymbol{s}_i[n]$ is

$$\frac{1}{2}\begin{bmatrix} 1 & -\sin(\phi_i) \\ -\sin(\phi_i) & 1 \end{bmatrix}$$

which depends only on ϕ_i, and therefore, when presenting the population covariances of the time-series $\boldsymbol{s}_i[n]$ in $\mathbb{R}^3$, two coordinates are fixed and only one varies with i.

We generate two observable subsets, $\mathcal{X}^{(k)} = \{\boldsymbol{x}_i^{(k)}[n]\}_{i=1}^{100}$, $k = 1, 2$ such that:

$$\boldsymbol{x}_i^{(k)}[n] = \boldsymbol{M}^{(k)} \boldsymbol{s}_i[n]$$

where $\boldsymbol{M}^{(1)}$ is randomly chosen, and $\boldsymbol{M}^{(2)} = 1.5\left(\begin{smallmatrix} -1 & 0 \\ 0 & 1 \end{smallmatrix}\right)\boldsymbol{M}^{(1)}$. The two subsets $\mathcal{X}^{(1)}$ and $\mathcal{X}^{(2)}$ can be viewed as two different observations of $\{\boldsymbol{s}_i[n]\}$ through two unknown observation functions $\boldsymbol{M}^{(1)}$ and $\boldsymbol{M}^{(2)}$. For example, $\mathcal{X}^{(1)}$ and $\mathcal{X}^{(2)}$ can represent two different batches, and $\boldsymbol{M}^{(1)}$ and $\boldsymbol{M}^{(2)}$ can represent the discrepancy between two different sessions of a particular experiment. For each $\boldsymbol{x}_i^{(k)}[n]$, we compute its sample covariance matrix by

$$\boldsymbol{P}_i^{(k)} = \frac{1}{T}\sum_n \boldsymbol{x}_i^{(k)}[n]\left(\boldsymbol{x}_i^{(k)}[n]\right)^T = \boldsymbol{M}^{(k)} \boldsymbol{P}_{s_i} \left(\boldsymbol{M}^{(k)}\right)^T$$

where $\boldsymbol{P}_{s_i}$ denotes the inaccessible sample covariance of $\boldsymbol{s}_i[n]$, which is given by:

$$\boldsymbol{P}_{s_i} = \frac{1}{T}\sum_{n=0}^{T-1} \boldsymbol{s}_i[n]\left(\boldsymbol{s}_i[n]\right)^T$$

Our goal is to obtain a new representation of the observed data both in $\mathcal{X}^{(1)}$ and $\mathcal{X}^{(2)}$, which circumvents the effect of $\boldsymbol{M}^{(1)}$ and $\boldsymbol{M}^{(2)}$. Moreover, in the new representation, we aspire to associate two observations from possibly different subsets which have a similar initial phase ϕ_i.

In Figure 3 we plot the 2×2 SPD matrices in $\mathbb{R}^3$, where the black points mark the boundaries of the cone manifold. The red line marks the center of the cone, given by $\alpha \boldsymbol{I}$ for $\alpha \in [0, 2]$, and the blue point on the red line indicates the identity matrix $\boldsymbol{I}$, namely, where $\alpha = 1$.

Figure 3(a) presents the two subsets $\mathcal{P}^{(k)} = \{\boldsymbol{P}_i^{(k)}\}$, $k = 1, 2$ of accessible sample covariance matrices, colored by ϕ_i (left) and by k (right). We observe that the two subsets $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$ are completely separated, while each subset has a similar

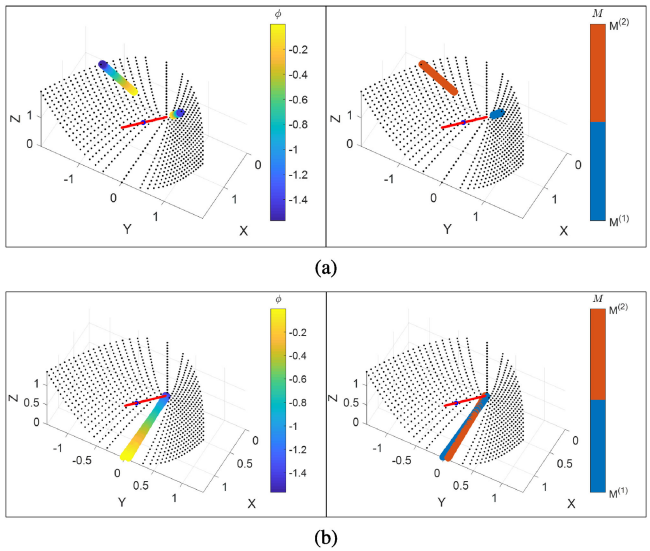

Fig. 3. Synthetic example, applying Steps (1)–(3) of Algorithm 1. (a) Scatter plot of $\mathcal{P}^{(k)}$ colored by ϕ_i (left) and by k (right). (b) Scatter plot of $\{\mathbf{\Gamma}_i^{(k)}\}$ obtained by Algorithm 1 colored by ϕ_i (left) and by k (right). Note that in the new representation (b), the two subsets are aligned, namely the discrepancy caused by $\boldsymbol{M}^{(1)}$ and $\boldsymbol{M}^{(2)}$ is removed while the intrinsic structure given by ϕ_i is preserved.

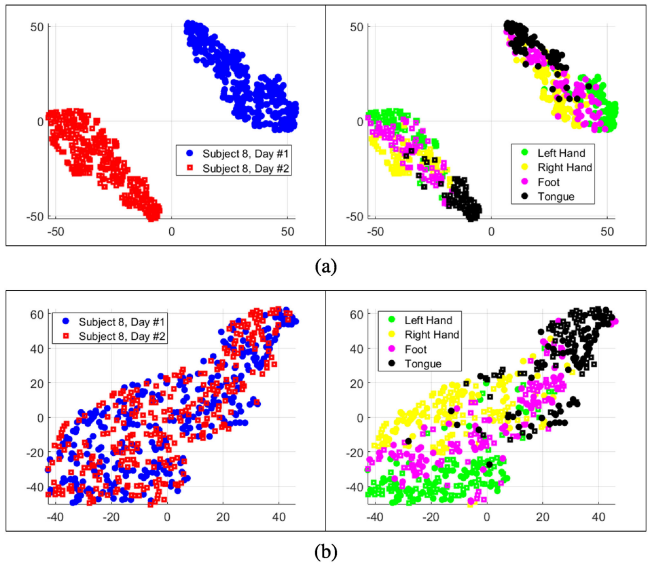

Fig. 4. Representation of a single subject's (#8) recordings from different days. (a) Scatter plot of the "baseline" $\{\boldsymbol{B}_i^{(k)}\}$ colored by the different day (left) and by the mental task (right). (b) Scatter plot of $\{\tilde{\boldsymbol{S}}_i^{(k)}\}$ obtained by Algorithm 1 colored by the different day (left) and by the mental task (right). Note the difference between the different days of experiments is completely removed. More importantly, we further observe that the new representations of the two subsets are aligned, i.e., we obtain similar representations of two recordings associated with the same mental task, regardless of their respective sessions (days of experiments).

structure governed by the values of ϕ_i. We apply Steps (1)–(3) of Algorithm 1 to the subsets $\mathcal{P}^{(k)}$ to obtain $\{\mathbf{\Gamma}_i^{(k)}\}$.

Figure 3(b) presents $\{\mathbf{\Gamma}_i^{(k)}\}$, colored by ϕ_i (left) and k (right). Now we observe that in the new representation, the two subsets are aligned, namely the discrepancy caused by $\boldsymbol{M}^{(1)}$ and $\boldsymbol{M}^{(2)}$ is removed while the intrinsic structure given by ϕ_i is preserved. As a result, we can associate covariance matrices from different batches but with similar underlying ϕ_i values. Note that this was accomplished by Algorithm 1 in a completely unsupervised manner, without access to the hidden "labels" ϕ_i.

B. BCI - Motor Imagery

As described in Section III, we use data from the BCI competition IV [19]. The dataset contains EEG recordings acquired by 22 EEG electrodes from 9 subjects, where the data from each subject was recorded on 2 different days of experiments. The experiment protocol consists of repeated trials, where in each trial the subject was asked to imagine performing one out of four possible movements: (i) right hand, (ii) left hand, (iii) both feet, and (iv) tongue. Overall, in a single day, each movement was repeated 72 times by each subject, and therefore, the dataset contains 288 trials from each subject in each day of experiments.

We remark that all the algorithms participating in the competition reported on poor classification results for particular four subjects. Since our goal is not to improve the classification of the data from each subject, we excluded these four subjects (indexed 2, 4, 5, 6).

Initially, focusing on the data from a single subject, we show that Algorithm 1 builds a representation of the data which enables us to train a classifier with data from one day of experiments and apply it to data from the other day of experiments. Then, we further show that Algorithm 1 builds a representation that allows us to train a classifier based on data from one subject and apply it to data from a different subject without any additional labeled trials. Finally, we extend the latter result and show the performance on multiple subjects. For visualization purposes, we apply tSNE [20] to the obtained data from the different methods. In all the tSNE applications, we set the perplexity parameter to 20.

1) Single Subject – Different Days: In the first experiment, we process the recordings of Subject 8 (arbitrarily chosen) from the two days of experiments. We report that the results for the other 4 subjects were similar. We denote the subsets of trial recordings from day $k = 1, 2$ by $\mathcal{X}^{(k)} = \{\boldsymbol{x}_i^{(k)}\}_{i=1}^{288}$. From the recordings of each trial i, we compute the sample covariance matrix $\boldsymbol{P}_i^{(k)} \in \mathbb{R}^{22\times 22}$, and denote $\mathcal{P}^{(k)} = \{\boldsymbol{P}_i^{(k)}\}$.

To highlight the challenge, we first compute $\hat{\boldsymbol{P}}$, the Riemannian mean of all covariance matrices (from both subsets). Then, we project the matrices onto $\mathcal{T}_{\hat{\boldsymbol{P}}}\mathcal{M}$ by computing $\boldsymbol{B}_i^{(k)} = \mathrm{Log}_{\hat{\boldsymbol{P}}}(\boldsymbol{P}_i^{(k)})$. For visualization purpose, we apply tSNE to the vectors $\{\boldsymbol{B}_i^{(k)}\}$. Figure 4(a) presents the two dimensional representation of the vectors obtained by the tSNE algorithm. Namely, each point in the figure is the representation of a vector $\boldsymbol{B}_i^{(k)}$. On the left, the points are colored according to the different days (indexed by $k = 1, 2$), and on the right, the points are colored according to the mental task. We observe that, similarly to the toy problem, the recordings from the different days are completely separated. This implies that one cannot train a classifier from the recordings from day 1 and apply it to the recordings from day 2.

We apply Algorithm 1 to the subsets $\mathcal{P}^{(k)}$ covariance matrices and obtain the subsets $\tilde{\mathcal{S}}^{(k)} = \{\tilde{\boldsymbol{S}}_i^{(k)}\}$. The matrices $\tilde{\boldsymbol{S}}_i^{(k)}$ are the new representation after applying domain adaptation (Algorithm 1). These matrices could be transformed into feature vectors that consist only of elements from the upper triangular part of the matrices, exploiting the symmetry of the matrices. By (5), these feature vectors lie in a Euclidean space, where the Euclidean distance approximates the Riemannian distance between the corresponding SPD matrices after Parallel Transport. Since our feature vectors lie in a Euclidean space, in all experiments we use a linear SVM classifier in order to quantify the quality of the representation. Figure 4(b) presents the two dimensional representation of the vectors $\tilde{\mathcal{S}}^{(k)}$ obtained by the tSNE algorithm. On the left, the points are colored according to the different days, and on the right, the points are colored according to the mental task. We observe that the difference between the different days of experiments is completely removed. More importantly, we further observe that the new representations of the two subsets are aligned, i.e., we obtain similar representations of two recordings associated with the same mental task, regardless of their respective sessions (days of experiments).

2) Two Subjects: We repeat the evaluation, but now with the two subsets $\mathcal{X}^{(k)}$, $k = 1, 2$ which were recorded from two subjects, specifically, Subject 3 and Subject 8. We repeat the steps from the previous examination. We compute $\hat{\boldsymbol{P}}$, the Riemannian mean of all covariance matrices (from both subsets). Then, we project the covariance matrices onto $\mathcal{T}_{\hat{\boldsymbol{P}}}\mathcal{M}$ and apply tSNE to obtain two dimensional representations. Figure 5(a) presents the two dimensional representations obtained by the tSNE algorithm. On the left, the points are colored by the subject index, and on the right the points are colored according to the mental task. Similarly to the single-subject two-sessions case, we observe that the recordings from different subjects are completely separated in the obtained representation.

We also apply the mean transport approach presented in [6]. The mean transport is obtained by projecting each subset of covariance matrices $\mathcal{P}^{(k)}$ to its own tangent plane $\mathcal{T}_{\overline{\boldsymbol{P}}^{(k)}}\mathcal{M}$, where $\overline{\boldsymbol{P}}^{(k)}$ is the Riemannian mean of the k-th subset. In other words, we compute $\boldsymbol{S}_i^{(k)} = \mathrm{Log}_{\overline{\boldsymbol{P}}^{(k)}}(\boldsymbol{P}_i^{(k)})$. Figure 5(b) presents the two dimensional representation obtained by the tSNE algorithm. We observe that indeed the two subsets are not separated as in Figure 5(a), however, the inner structure of each subset was not preserved. Thus, this scheme is insufficient and does not support training a classifier based on data from one subject and applying it to data from another subject.

Finally, we apply Algorithm 1 to the subsets $\mathcal{P}^{(k)}$, and obtain the subsets $\tilde{\mathcal{S}}^{(k)} = \{\tilde{\boldsymbol{S}}_i^{(k)}\}$. Figure 5(c) presents the two dimensional representation of the subsets $\tilde{\mathcal{S}}^{(k)}$ obtained by the tSNE algorithm. We observe that in this representation, the subsets are not separated. Moreover, the two subsets are aligned according to the mental tasks, and indeed points that correspond to the same mental task assumed a similar value in the new representation. This new representation allows us to train a classifier using recordings from one subject and apply it to recordings from another subject.

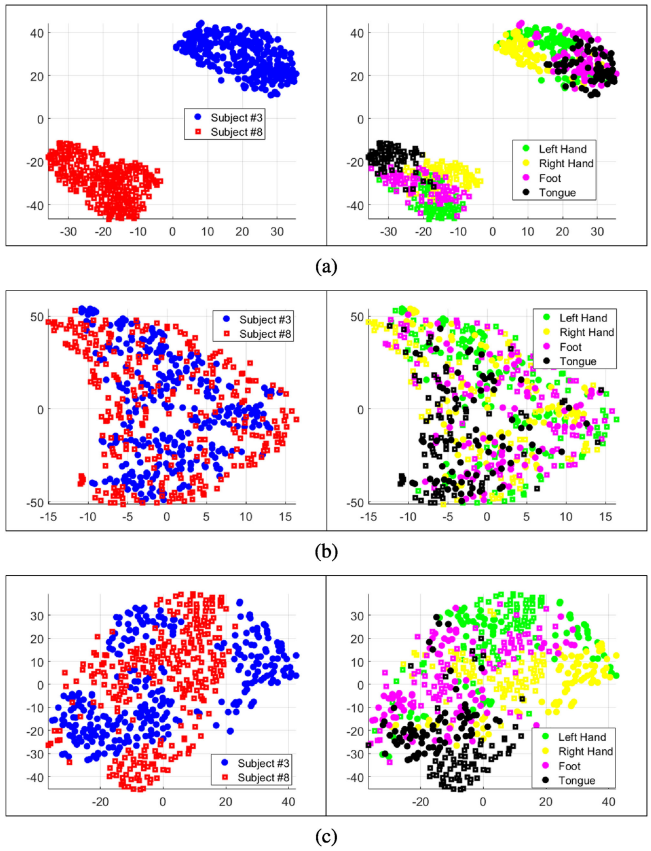

Fig. 5. Representation of recordings from two subjects (#3 and #8). (a) Scatter plot of the "baseline" $\{\boldsymbol{B}_i^{(k)}\}$ colored by the subject (left) and by the mental task (right). (b) Scatter plot of the representation of $\{\boldsymbol{S}_i^{(k)}\}$ obtained by the "mean transport", colored by the subject (left) and by the mental task (right). (c) Scatter plot of $\{\tilde{\boldsymbol{S}}_i^{(k)}\}$ obtained by Algorithm 1 colored by the subject (left) and by the mental task (right). See the text for details.

3) Multiple Subjects: In the third experiment, we apply Algorithm 1 to multiple subjects. We processed data from five subjects (indexed 1, 3, 7, 8 and 9) and from all trials of the mental tasks: left hand, right hand, foot, and tongue. We denote the subsets of the recordings by $\mathcal{X}^{(k)} = \{\boldsymbol{x}_i^{(k)}\}$ for $k = 1, 3, 7, 8, 9$. As before, we compute the covariance matrices $\mathcal{P}^{(k)} = \{\boldsymbol{P}_i^{(k)}\}$. We compute $\hat{\boldsymbol{P}}$, the Riemannian mean of all covariance matrices. Then, we project the covariance matrices onto $\mathcal{T}_{\hat{\boldsymbol{P}}}\mathcal{M}$ and apply tSNE to obtain a two dimensional representation. Figure 6(a) presents the two dimensional representation obtained by the tSNE algorithm. On the left, the points are colored by the subject index, and on the right the points are colored according to the mental task. As before, we observe that the recordings from different subjects are completely separated.

Next, we apply Algorithm 1 to the five subsets of covariance matrices $\mathcal{P}^{(k)}$ and obtain five subsets of new representations $\tilde{\mathcal{S}}^{(k)} = \{\tilde{\boldsymbol{S}}_i^{(k)}\}$. Figure 5(b) presents the two dimensional representation of the subsets $\tilde{\mathcal{S}}^{(k)}$ obtained by the tSNE algorithm. We observe that also in the multiple subjects scenario, Algorithm 1 was able to center and align the subsets.

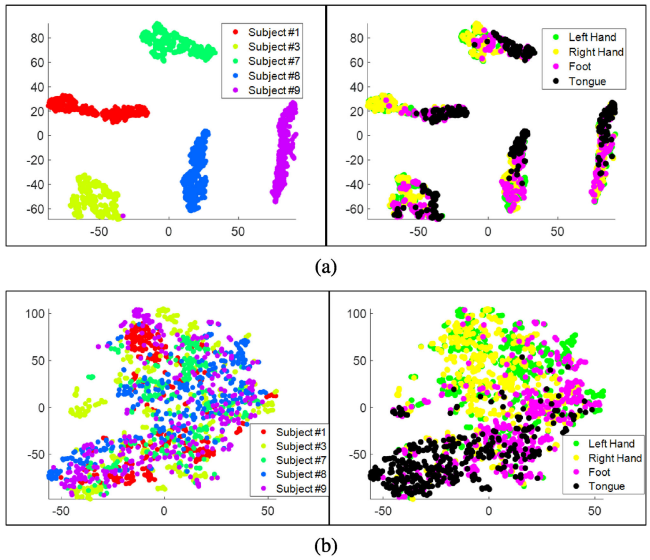

Fig. 6. Representation of recordings from multiple subjects. (a) Scatter plot of the "baseline" $\{\boldsymbol{B}_i^{(k)}\}$ colored by the subject (left) and by the mental task (right). (b) Scatter plot of $\{\tilde{\boldsymbol{S}}_i^{(k)}\}$ obtain by Algorithm 1 colored by the subject (left) and by the mental task (right). See the text for details.

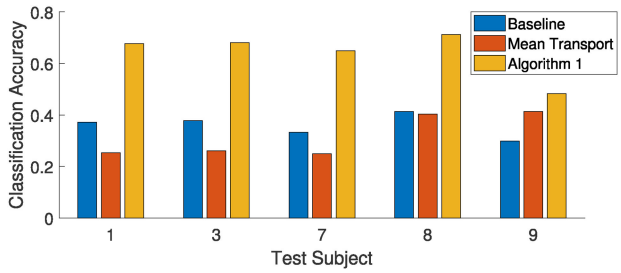

Fig. 7. The classification accuracy obtained by the three competing methods.

To provide quantitative results, we apply a leave-one-subject-out cross validation, namely, we trained a linear SVM classifier based on 4 out of the 5 subjects and evaluated the classification accuracy for each one of the three methods mentioned in Section IV-B2. We compare the classification accuracy of Algorithm 1 to the two other approaches denoted by: (i) "Baseline (No Transport)", and (ii) the "Mean Transport" approach proposed in [6].

Figure 7 presents the classification accuracy obtained using the three competing methods as a function of the evaluated subject. In all cases, applying Algorithm 1 to the data dramatically improved the classification accuracy. In addition, Figure 8 presents the confusion matrices per subject. For example, the second row presents the performance of the three algorithms when subject #3 was tested. Figure 8 (left) presents the confusion matrices obtained from the data without applying any transportation ("Baseline"). Figure 8 (center) presents the confusion matrices obtained by the "Mean Transport" approach. Figure 8 (right) presents the confusion matrices obtained by Algorithm 1. We observe that Algorithm 1 obtains significantly better classification results compared with the "Baseline" and "Mean Transport" algorithms. In addition, the confusion matrices highlight the challenge in training a classifier from multiple subject data. For example, in subject #7, the confusion matrices in Figure 8 (left) and (center), the "foot" mental task falsely dominated the prediction (it was predicted 227 and 288 times, respectively, whereas it was performed only 72 times). This implies that the decision regions of the classifiers are completely misaligned with the data from a new unseen subject.

C. Sleep Stage Identification

Here, we demonstrate the applicability of Algorithm 1 to real medical signals. Specifically, we address the problem of sleep stage identification. Typically, for this purpose, data are collected in sleep clinics with multiple multimodal sensors, and then, analyzed by a human expert. There are six different sleep stages: awake, REM, and sleep stages 1–4, indicating shallow to deep sleep.

The data we used is available online in [21] and described in detail in [22]. A single patient's night recording contains several measurements including two EEG channels and one electrooculography (EOG) channel sampled at 100[Hz]. We used recordings from three subjects. We split each subject's night into non-overlapping 30 seconds windows. We omit the awake and sleep stage 4 windows due to too few occurrences. For visual purposes, we kept only windows corresponding to REM and stage 3.

We denote the i-th window of the k-th subject by $\boldsymbol{x}_i^{(k)}(t)$ with its corresponding covariance matrix $\boldsymbol{P}_i^{(k)} \in \mathbb{R}^{3\times 3}$. We first compute $\hat{\boldsymbol{P}}$, the Riemannian mean of all covariance matrices (from the three subsets). Then, we project the matrices onto $\mathcal{T}_{\hat{\boldsymbol{P}}}\mathcal{M}$ by computing $\boldsymbol{S}_i^{(k)} = \mathrm{Log}_{\hat{\boldsymbol{P}}}(\boldsymbol{P}_i^{(k)})$. For visualization purposes, we apply PCA to the vectors $\{\boldsymbol{S}_i^{(k)}\}$ and present the first three principle components. Since the covariance matrices in this experiment are of size 3×3, dimension reduction using PCA was sufficient. It was preferred here over tSNE since it better preserves the global geometry of the data.

Figure 9(a) presents the three dimensional representation of the vectors obtained by PCA. Namely, each point in the figure is the representation of a vector $\boldsymbol{S}_i^{(k)}$. On the left, the points are colored according to the different subjects, and on the right, the points are colored according to the sleep stage. We observe that the points are clustered according to the different subjects. We apply Algorithm 1 to the three subsets $\{\boldsymbol{P}_i^{(k)}\}$ of covariance matrices and obtain the subsets $\{\tilde{\boldsymbol{S}}_i^{(k)}\}$. Figure 9(b) presents the two dimensional representation of the vectors $\{\tilde{\boldsymbol{S}}_i^{k)}\}$ obtained by PCA. On the left, the points are colored according to the different subjects, and on the right, the points are colored according to the sleep stage. Now we observe that the data is clustered according to the sleep stage while the difference between the three subjects is completely removed.

As in the Subsection IV-B3, to provide quantitative results, we train a linear SVM classifier based on Subject 1

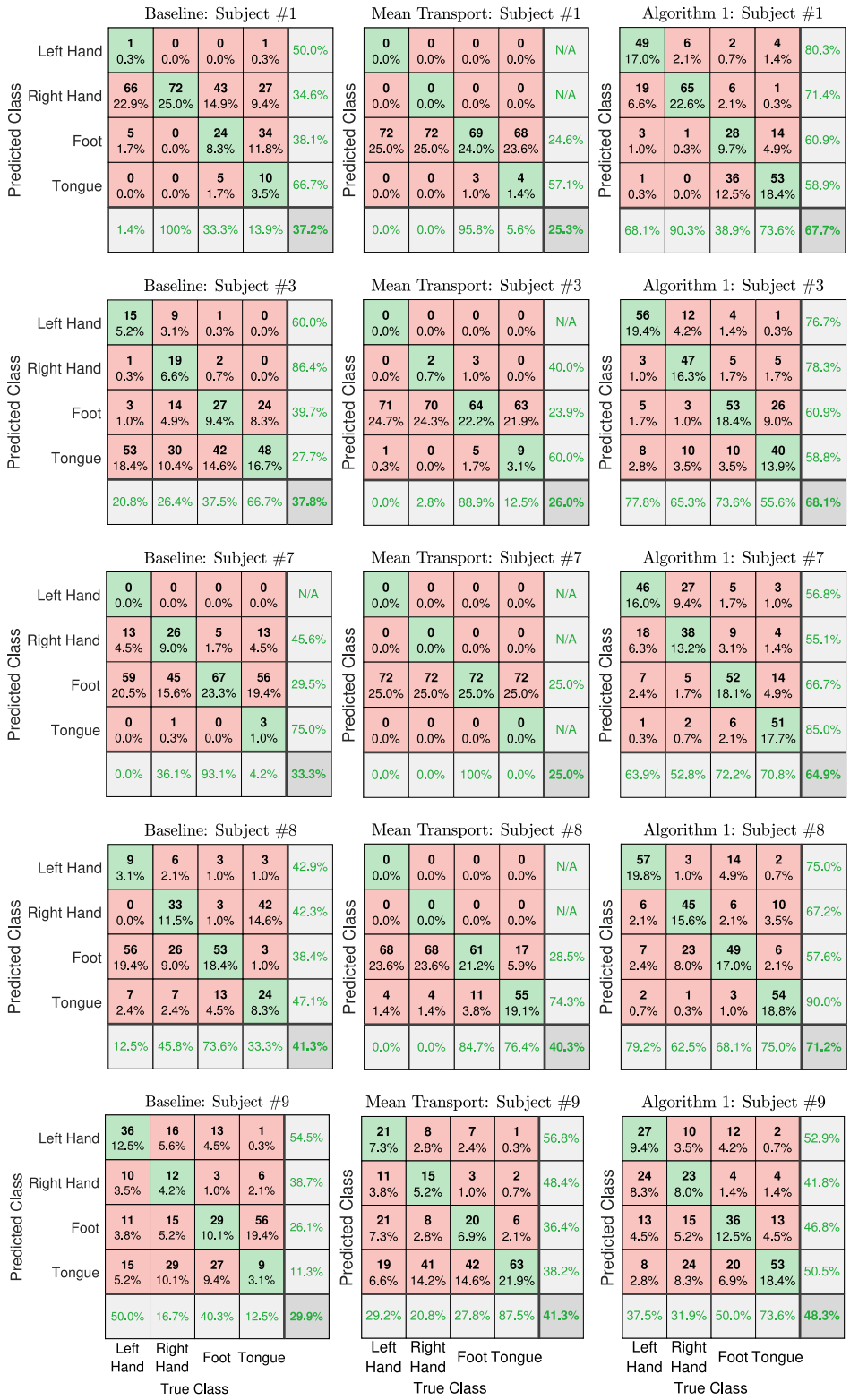

Fig. 8. The confusion matrices of the BCI task classification of data. (left) Baseline. (center) Mean Transport, (right) Algorithm 1.

and Subject 2 and evaluate the classification accuracy on Subject 3. Figure 10 presents the obtained confusion matrices. Figure 10 (left) presents the confusion matrix obtained from the data without applying any adaptation ("Baseline"). Figure 10 (center) presents the confusion matrix obtained by the "Mean Transport" approach. Figure 10 (right) presents the confusion matrix obtained by Algorithm 1. We observe that using Algorithm 1 demonstrates better classification results compared with the "Baseline" and the "Mean Transport" algorithms.

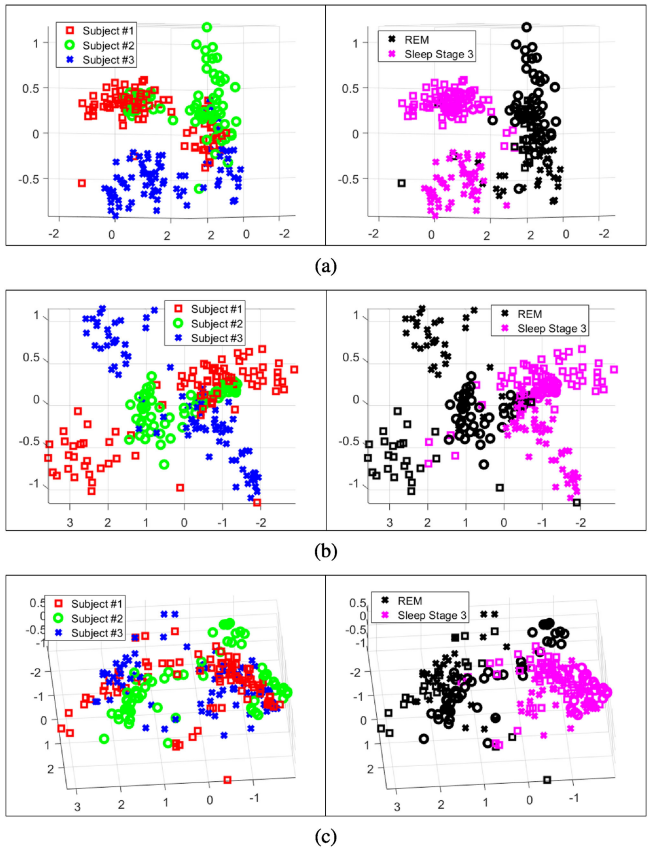

Fig. 9. Representation of recordings for sleep stage identification. (a) Scatter plot of the "baseline" $\{\boldsymbol{B}_i^{(k)}\}$ (after PCA) colored by subject (left) and by sleep stage (right). (b) Scatter plot of $\{\boldsymbol{S}_i^{(k)}\}$ (after PCA) obtained by the "mean transport", colored by subject (left) and by sleep stage (right). (c) Scatter plot of $\{\tilde{\boldsymbol{S}}_i^{(k)}\}$ (after PCA) obtained by Algorithm 1 colored by subject (left) and by sleep stage (right).

D. Mental Arithmetic Identification From EEG

In this subsection, we demonstrate the proposed domain adaptation on a large dataset consisting of EEG recordings from 29 subjects for the purpose of mental arithmetic identification [23] (Dataset B). The EEG data were recorded using 30 electrodes at 1000 Hz. The EEG recordings were obtained while the subjects were in two mental states. In the first mental state, the subjects were instructed to perform repeated simple arithmetic calculations. In the second, baseline, state, the subjects were instructed to rest. The experiment consisted of three sessions. Each session comprised 20 repetitions of the above tasks. Thus, overall, the dataset contains 60 trials per subject.

Similarly to the previous experiments, we computed the covariance matrix of each trial and applied Algorithm 1 and the competing algorithms for domain adaptation in order to account for the inherent difference between the three sessions. First we visually demonstrate the performance on a single subject. Figure 11(a) presents the two dimensional tSNE representation obtained by the "baseline" algorithm. One can notice a clear shift in the representation of trials from Session 1 compared to trials from Session 2 and 3. Figure 11(b) presents the two dimensional tSNE representation obtained by Algorithm 1. After applying our domain adaptation, we can observe that the shift between the sessions is no longer apparent and that the trials are clustered according to the mental state, as desired.

We repeated this test for all 29 subjects and report that a similar difference between the sessions is present in most of the subjects. To provide a quantitative assessment of the adaptation, we applied a leave-one-session-out cross-validation using a linear SVM classifier to the representations obtained from the 3 competing methods. The average classification accuracy obtained for all 29 subjects is depicted in the following table:

Baseline	MT	PT
74%	73%	78%

V. Conclusions

Analyzing complex data in high-dimension is challenging, since such data do not live in a Euclidean space. Therefore, basic operations such as comparisons, additions, and subtractions, which are the basis of any analysis and learning technique, do not necessarily exist and are not appropriately defined. In this work, we propose to view the complex data through the lens of SPD matrices, which reside on an analytic Riemannian manifold. Using the Riemannian geometry of SPD matrices, we presented an approach for multi-domain data representation. Based on this new representation, we proposed an algorithm for domain adaptation. We extend the existing results in the Riemannian geometry of SPD matrices and establish a framework for the justification and analysis of the proposed solution. We demonstrated the usefulness of the presented domain adaptation method in applications to simulation and real recorded data.

Appendix A
Proof of Lemma 1

Proof: The PT of $\boldsymbol{S}$ along the geodesic between $\boldsymbol{B}$ and $\boldsymbol{A}$ is given by [18]:

$$\Gamma_{\boldsymbol{B}\to\boldsymbol{A}}(\boldsymbol{S}) = \boldsymbol{M}\boldsymbol{S}\boldsymbol{M}^T$$

where $\boldsymbol{M} = \boldsymbol{B}^{\frac{1}{2}} \exp\left(\boldsymbol{B}^{-\frac{1}{2}} \frac{1}{2} \mathrm{Log}_{\boldsymbol{B}}(\boldsymbol{A})\, \boldsymbol{B}^{-\frac{1}{2}}\right) \boldsymbol{B}^{-\frac{1}{2}}$. Now we will show that $\boldsymbol{M}$ can be written more simply, proving a more efficient way to compute it. We have

$$\begin{aligned}\boldsymbol{M} &= \boldsymbol{B}^{\frac{1}{2}} \exp\left(\boldsymbol{B}^{-\frac{1}{2}} \frac{1}{2} \mathrm{Log}_{\boldsymbol{B}}(\boldsymbol{A})\, \boldsymbol{B}^{-\frac{1}{2}}\right) \boldsymbol{B}^{-\frac{1}{2}} \\ &= \boldsymbol{B}^{\frac{1}{2}} \exp\left(\frac{1}{2} \log\left(\boldsymbol{B}^{-\frac{1}{2}} \boldsymbol{A} \boldsymbol{B}^{-\frac{1}{2}}\right)\right) \boldsymbol{B}^{-\frac{1}{2}} \\ &= \boldsymbol{B}^{\frac{1}{2}} \left(\boldsymbol{B}^{-\frac{1}{2}} \boldsymbol{A} \boldsymbol{B}^{-\frac{1}{2}}\right)^{\frac{1}{2}} \boldsymbol{B}^{-\frac{1}{2}}\end{aligned}$$

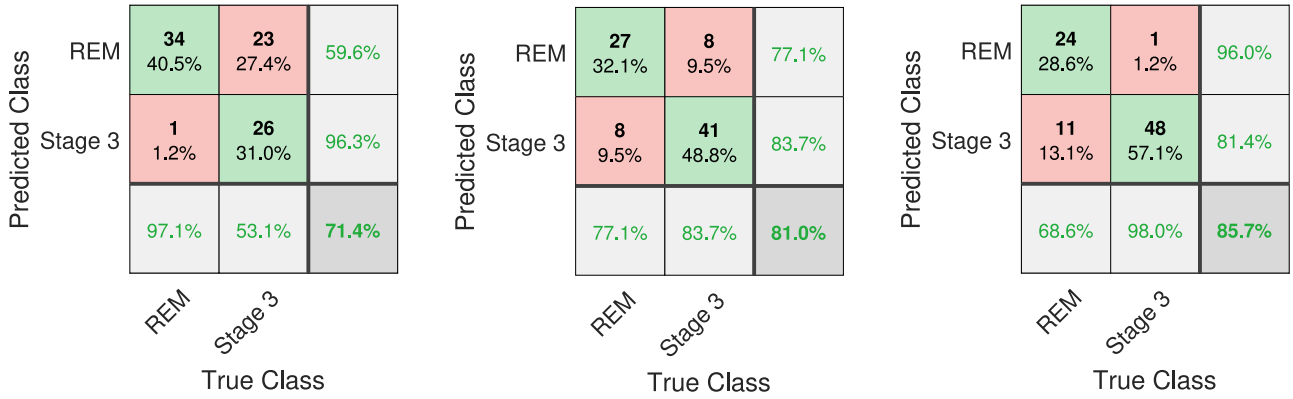

Fig. 10. Confusion matrices of sleep stage identification based on recordings from Subject 3. (left) Baseline, (center) Mean Transport, (right) Algorithm 1.

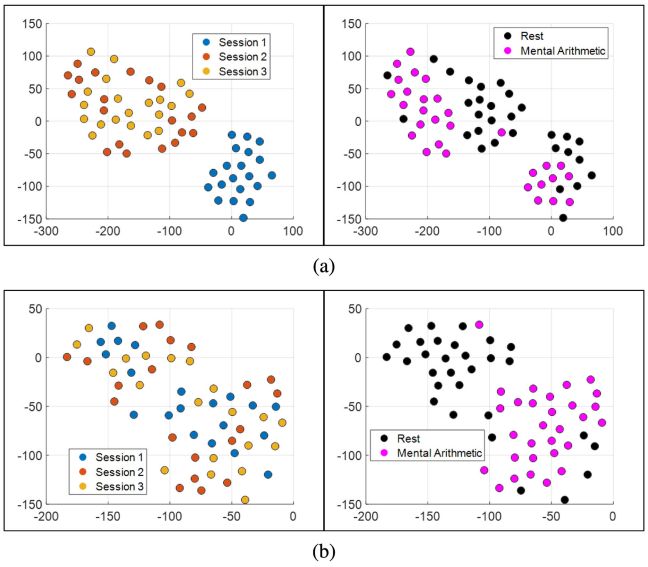

Fig. 11. Representation of recordings from subject #1. (a) Scatter plot obtained by the "baseline" algorithm colored by the session (left) and by the mental state (right). (b) Scatter plot obtained by Algorithm 1 colored by the session index (left) and by the mental state (right). See the text for details.

and also

$$\boldsymbol{M}^2 = \left(\boldsymbol{B}^{\frac{1}{2}}\left(\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{-\frac{1}{2}}\right)^{\frac{1}{2}}\boldsymbol{B}^{-\frac{1}{2}}\right)^2 = \boldsymbol{A}\boldsymbol{B}^{-1} = \boldsymbol{E}^2$$

and since $\boldsymbol{A}\boldsymbol{B}^{-1}$ is similar to $\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{-\frac{1}{2}} > 0$, it has only positive eigenvalues and the square root is unique, namely $\boldsymbol{E} = \boldsymbol{M}$. ■

APPENDIX B
PROOF OF THEOREM 1

For better readability we denote $\boldsymbol{A} = \overline{\boldsymbol{P}}^{(1)}$ and $\boldsymbol{B} = \overline{\boldsymbol{P}}^{(2)}$. First we remark that $\Gamma_{\boldsymbol{B}\to\boldsymbol{A}}$ is well defined since $\mathcal{T}_{\boldsymbol{P}}\mathcal{M}$ is the space of all symmetric matrices regardless the matrix $\boldsymbol{P}$,[1] and if the input $\boldsymbol{S}$ is symmetric then by definition $\Gamma_{\boldsymbol{B}\to\boldsymbol{A}}(\boldsymbol{S})$ is symmetric as well.

Proof of Theorem 1: Condition (1) is immediate since Γ is a linear operation, and therefore we have

$$\sum_{i=1}^{N_2}\Gamma\left(\boldsymbol{S}_i^{(2)}\right) = \sum_{i=1}^{N_2}\boldsymbol{E}\boldsymbol{S}_i^{(2)}\boldsymbol{E}^T = \boldsymbol{E}\underbrace{\sum_{i=1}^{N_2}\boldsymbol{S}_i^{(2)}}_{=0}\boldsymbol{E}^T = 0$$

(10) shown at the bottom of this page, Conditions (2) and (3) are derived from Lemma 1, since these are properties of PT. For completeness, we provide their explicit proofs. Proof of condition (2): Let $\boldsymbol{A}, \boldsymbol{B} \in \mathcal{M}$ and $\boldsymbol{S}_1, \boldsymbol{S}_2 \in \mathcal{T}_{\boldsymbol{B}}\mathcal{M}$ and denote

[1] Any tangent plane to the SPD manifold $\mathcal{M}$ is the entire space of symmetric matrices [18].

$$\begin{aligned}
\boldsymbol{B}_2^{\frac{1}{2}}\left(\boldsymbol{B}_2^{-\frac{1}{2}}\boldsymbol{A}_2\boldsymbol{B}_2^{-\frac{1}{2}}\right)^{\frac{1}{2}}\boldsymbol{B}_2^{\frac{1}{2}} &= \boldsymbol{B}_2^{\frac{1}{2}}\left(\left(\boldsymbol{E}\boldsymbol{B}_1\boldsymbol{E}^T\right)^{-\frac{1}{2}}\boldsymbol{E}\boldsymbol{A}_1\boldsymbol{E}^T\left(\boldsymbol{E}\boldsymbol{B}_1\boldsymbol{E}^T\right)^{-\frac{1}{2}}\right)^{\frac{1}{2}}\boldsymbol{B}_2^{\frac{1}{2}} \\
&= \boldsymbol{B}_2^{\frac{1}{2}}\left(\underbrace{\left(\boldsymbol{E}\boldsymbol{B}_1\boldsymbol{E}^T\right)^{\frac{1}{2}}\boldsymbol{E}^{-T}\boldsymbol{B}_1^{-\frac{1}{2}}}_{\boldsymbol{K}}\boldsymbol{B}_1^{-\frac{1}{2}}\boldsymbol{A}_1\boldsymbol{B}_1^{-\frac{1}{2}}\underbrace{\boldsymbol{B}_1^{-\frac{1}{2}}\boldsymbol{E}^{-1}\left(\boldsymbol{E}\boldsymbol{B}_1\boldsymbol{E}^T\right)^{\frac{1}{2}}}_{\boldsymbol{K}^T}\right)^{\frac{1}{2}}\boldsymbol{B}_2^{\frac{1}{2}} \\
&= \boldsymbol{B}_2^{\frac{1}{2}}\left(\boldsymbol{E}\boldsymbol{B}_1\boldsymbol{E}^T\right)^{\frac{1}{2}}\boldsymbol{E}^{-T}\boldsymbol{B}_1^{-\frac{1}{2}}\left(\boldsymbol{B}_1^{-\frac{1}{2}}\boldsymbol{A}_1\boldsymbol{B}_1^{-\frac{1}{2}}\right)^{\frac{1}{2}}\boldsymbol{B}_1^{-\frac{1}{2}}\boldsymbol{E}^{-1}\left(\boldsymbol{E}\boldsymbol{B}_1\boldsymbol{E}^T\right)^{\frac{1}{2}}\boldsymbol{B}_2^{\frac{1}{2}} \\
&= \boldsymbol{E}\boldsymbol{B}_1\boldsymbol{E}^T\boldsymbol{E}^{-T}\boldsymbol{B}_1^{-\frac{1}{2}}\left(\boldsymbol{B}_1^{-\frac{1}{2}}\boldsymbol{A}_1\boldsymbol{B}_1^{-\frac{1}{2}}\right)^{\frac{1}{2}}\boldsymbol{B}_1^{-\frac{1}{2}}\boldsymbol{E}^{-1}\boldsymbol{E}\boldsymbol{B}_1\boldsymbol{E}^T \\
&= \boldsymbol{E}\boldsymbol{B}_1^{\frac{1}{2}}\left(\boldsymbol{B}_1^{-\frac{1}{2}}\boldsymbol{A}_1\boldsymbol{B}_1^{-\frac{1}{2}}\right)^{\frac{1}{2}}\boldsymbol{B}_1^{\frac{1}{2}}\boldsymbol{E}^T
\end{aligned} \tag{10}$$

$\boldsymbol{E} = (\boldsymbol{A}\boldsymbol{B}^{-1})^{\frac{1}{2}} = \boldsymbol{B}^{\frac{1}{2}}(\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{-\frac{1}{2}})^{\frac{1}{2}}\boldsymbol{B}^{-\frac{1}{2}}$. We have

$$\begin{aligned}\boldsymbol{A}^{-1}\boldsymbol{E} &= \boldsymbol{A}^{-1}\boldsymbol{B}^{\frac{1}{2}}\left(\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{-\frac{1}{2}}\right)^{\frac{1}{2}}\boldsymbol{B}^{-\frac{1}{2}} \\ &= \boldsymbol{A}^{-1}\boldsymbol{B}^{\frac{1}{2}}\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{-\frac{1}{2}}\left(\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{-\frac{1}{2}}\right)^{-\frac{1}{2}}\boldsymbol{B}^{-\frac{1}{2}} \\ &= \boldsymbol{B}^{-\frac{1}{2}}\left(\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{-\frac{1}{2}}\right)^{-\frac{1}{2}}\boldsymbol{B}^{-\frac{1}{2}}\end{aligned}$$

Namely, $\boldsymbol{A}^{-1}\boldsymbol{E}$ is a symmetric matrix. Thus, we get that

$$\begin{aligned}\boldsymbol{E}^T\boldsymbol{A}^{-1}\boldsymbol{E} &= \boldsymbol{E}^T\boldsymbol{E}^T\boldsymbol{A}^{-1} = \left(\boldsymbol{A}\boldsymbol{B}^{-1}\right)^T\boldsymbol{A}^{-1} \\ &= \left(\boldsymbol{B}^{-1}\boldsymbol{A}\right)\boldsymbol{A}^{-1} = \boldsymbol{B}^{-1}\end{aligned}$$

and finally, we obtain

$$\begin{aligned}\left\langle \boldsymbol{E}\boldsymbol{S}_1\boldsymbol{E}^T, \boldsymbol{E}\boldsymbol{S}_2\boldsymbol{E}^T\right\rangle_{\boldsymbol{A}} &= \left\langle \boldsymbol{E}\boldsymbol{S}_1\boldsymbol{E}^T\boldsymbol{A}^{-1}, \boldsymbol{A}^{-1}\boldsymbol{E}\boldsymbol{S}_2\boldsymbol{E}^T\right\rangle \\ &= \mathrm{Tr}\left\{\boldsymbol{E}\boldsymbol{S}_1\boldsymbol{E}^T\boldsymbol{A}^{-1}\boldsymbol{E}\boldsymbol{S}_2\boldsymbol{E}^T\boldsymbol{A}^{-1}\right\} \\ &= \mathrm{Tr}\left\{\boldsymbol{S}_1\boldsymbol{E}^T\boldsymbol{A}^{-1}\boldsymbol{E}\boldsymbol{S}_2\boldsymbol{E}^T\boldsymbol{A}^{-1}\boldsymbol{E}\right\} \\ &= \mathrm{Tr}\left\{\boldsymbol{S}_1\boldsymbol{B}^{-1}\boldsymbol{S}_2\boldsymbol{B}^{-1}\right\} \\ &= \left\langle \boldsymbol{S}_1, \boldsymbol{S}_2\right\rangle_{\boldsymbol{B}}\end{aligned}$$

Proof of condition (3): Let $\boldsymbol{B} \in \mathcal{M}$ be an SPD matrix with the following spectral decomposition: $\boldsymbol{B} = \boldsymbol{M}\boldsymbol{\Lambda}\boldsymbol{M}^T$. Then, we have

$$\begin{aligned}\frac{d}{dt}\boldsymbol{B}^t &= \frac{d}{dt}\boldsymbol{M}\boldsymbol{\Lambda}^t\boldsymbol{M}^T = \boldsymbol{M}\boldsymbol{\Lambda}^t\log(\boldsymbol{\Lambda})\boldsymbol{M}^T \\ &= \boldsymbol{M}\boldsymbol{\Lambda}^t\boldsymbol{M}^T\boldsymbol{M}\log(\boldsymbol{\Lambda})\boldsymbol{M}^T = \boldsymbol{B}^t\log(\boldsymbol{B})\end{aligned}$$

Consider the geodesic $\varphi(t)$ from $\boldsymbol{B}$ to $\boldsymbol{A}$: $\varphi(t) = \boldsymbol{B}^{\frac{1}{2}}(\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{-\frac{1}{2}})^t\boldsymbol{B}^{\frac{1}{2}}$ Thus, its velocity at $t = 0$ is given by

$$\varphi'(0) = \boldsymbol{B}^{\frac{1}{2}}\log\left(\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{\frac{1}{2}}\right)\boldsymbol{B}^{\frac{1}{2}} = \mathrm{Log}_{\boldsymbol{B}}(\boldsymbol{A}) \tag{11}$$

and similarly, the velocity at $t = 1$ is given by

$$\begin{aligned}\varphi'(1) &= \boldsymbol{B}^{\frac{1}{2}}\left(\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{-\frac{1}{2}}\right)^1\log\left(\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{-\frac{1}{2}}\right)\boldsymbol{B}^{\frac{1}{2}} \\ &= -\boldsymbol{A}\boldsymbol{B}^{-\frac{1}{2}}\log\left(\boldsymbol{B}^{\frac{1}{2}}\boldsymbol{A}^{-1}\boldsymbol{B}^{\frac{1}{2}}\right)\boldsymbol{B}^{\frac{1}{2}} \\ &= -\boldsymbol{A}\boldsymbol{B}^{-\frac{1}{2}}\log\left(\boldsymbol{B}^{\frac{1}{2}}\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{B}\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{A}^{\frac{1}{2}}\boldsymbol{B}^{-\frac{1}{2}}\right)\boldsymbol{B}^{\frac{1}{2}} \\ &\underbrace{=}_{(*)} -\boldsymbol{A}\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{B}^{\frac{1}{2}}\boldsymbol{A}^{-\frac{1}{2}}\log\left(\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{B}\boldsymbol{A}^{-\frac{1}{2}}\right)\boldsymbol{A}^{\frac{1}{2}}\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{B}^{\frac{1}{2}} \\ &= -\boldsymbol{A}^{\frac{1}{2}}\log\left(\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{B}\boldsymbol{A}^{-\frac{1}{2}}\right)\boldsymbol{A}^{\frac{1}{2}} = -\mathrm{Log}_{\boldsymbol{A}}(\boldsymbol{B})\end{aligned} \tag{12}$$

where in $(*)$ we pull out $\boldsymbol{V} = \boldsymbol{B}^{\frac{1}{2}}\boldsymbol{A}^{-\frac{1}{2}}$ and $\boldsymbol{V}^{-1} = \boldsymbol{A}^{\frac{1}{2}}\boldsymbol{B}^{-\frac{1}{2}}$ from the log, since it is a scalar function: $\log\left(\boldsymbol{V}\boldsymbol{P}\boldsymbol{V}^{-1}\right) = \boldsymbol{V}\log(\boldsymbol{P})\boldsymbol{V}^{-1}$.

Let $\boldsymbol{U}$ be the following unitary matrix

$$\boldsymbol{U} = \boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{B}^{\frac{1}{2}}\left(\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{-\frac{1}{2}}\right)^{\frac{1}{2}}$$

Using $\boldsymbol{U}$, we can rewrite $\boldsymbol{E}$ as:

$$\boldsymbol{E} = \left(\boldsymbol{A}\boldsymbol{B}^{-1}\right)^{\frac{1}{2}} = \boldsymbol{B}^{\frac{1}{2}}\left(\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{-\frac{1}{2}}\right)^{\frac{1}{2}}\boldsymbol{B}^{-\frac{1}{2}} = \boldsymbol{A}^{\frac{1}{2}}\boldsymbol{U}\boldsymbol{B}^{-\frac{1}{2}} \tag{13}$$

Finally, by combining (11), (12) and 13, we have

$$\begin{aligned}\boldsymbol{E}\varphi'(0)\boldsymbol{E}^T &= \boldsymbol{A}^{\frac{1}{2}}\log\left(\boldsymbol{U}\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{U}^T\right)\boldsymbol{A}^{\frac{1}{2}} \\ &= -\boldsymbol{A}^{\frac{1}{2}}\log\left(\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{B}\boldsymbol{A}^{-\frac{1}{2}}\right)\boldsymbol{A}^{\frac{1}{2}} \\ &= -\mathrm{Log}_{\boldsymbol{A}}(\boldsymbol{B}) = \varphi'(1)\end{aligned}$$

■

APPENDIX C
PROOF OF THEOREM 2

Proof:

$$\begin{aligned}\Psi(\boldsymbol{P}) &= \mathrm{Exp}_{\boldsymbol{A}}\left(\Gamma_{\boldsymbol{B}\to\boldsymbol{A}}(\boldsymbol{S})\right) \\ &= \mathrm{Exp}_{\boldsymbol{A}}\left(\boldsymbol{E}\mathrm{Log}_{\boldsymbol{B}}(\boldsymbol{P})\boldsymbol{E}^T\right) \\ &= \boldsymbol{A}^{\frac{1}{2}}\exp\left(\boldsymbol{U}\log\left(\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{P}\boldsymbol{B}^{-\frac{1}{2}}\right)\boldsymbol{U}^T\right)\boldsymbol{A}^{\frac{1}{2}} \\ &= \boldsymbol{A}^{\frac{1}{2}}\boldsymbol{U}\exp\left(\log\left(\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{P}\boldsymbol{B}^{-\frac{1}{2}}\right)\right)\boldsymbol{U}^T\boldsymbol{A}^{\frac{1}{2}} \\ &= \boldsymbol{A}^{\frac{1}{2}}\boldsymbol{U}\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{P}\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{U}^T\boldsymbol{A}^{\frac{1}{2}} \\ &= \boldsymbol{A}^{\frac{1}{2}}\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{E}\boldsymbol{B}^{\frac{1}{2}}\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{P}\boldsymbol{B}^{-\frac{1}{2}}\boldsymbol{B}^{\frac{1}{2}}\boldsymbol{E}^T\boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{A}^{\frac{1}{2}} \\ &= \boldsymbol{E}\boldsymbol{P}\boldsymbol{E}^T\end{aligned}$$

where $\boldsymbol{U} = \boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{E}\boldsymbol{B}^{\frac{1}{2}}$ is a unitary matrix, and therefore can be pulled out of the scalar exp function. ■

APPENDIX D
PROOF OF PROPOSITION 1

Proof: Let $\boldsymbol{E}_1 = (\boldsymbol{A}_1\boldsymbol{B}_1^{-1})^{\frac{1}{2}} = \boldsymbol{B}_1^{\frac{1}{2}}(\boldsymbol{B}_1^{-\frac{1}{2}}\boldsymbol{A}_1\boldsymbol{B}_1^{-\frac{1}{2}})\boldsymbol{B}_1^{-\frac{1}{2}}$ and $\boldsymbol{E}_2 = (\boldsymbol{A}_2\boldsymbol{B}_2^{-1})^{\frac{1}{2}} = \boldsymbol{B}_2^{\frac{1}{2}}(\boldsymbol{B}_2^{-\frac{1}{2}}\boldsymbol{A}_2\boldsymbol{B}_2^{-\frac{1}{2}})\boldsymbol{B}_2^{-\frac{1}{2}}$. Since

$$\Gamma\left(\Gamma_{\boldsymbol{B}_1\to\boldsymbol{A}_1}(\boldsymbol{S})\right) = \boldsymbol{E}\boldsymbol{E}_1\boldsymbol{S}\boldsymbol{E}_1^T\boldsymbol{E}^T,$$

and

$$\Gamma_{\boldsymbol{B}_2\to\boldsymbol{A}_2}\left(\Gamma(\boldsymbol{S})\right) = \boldsymbol{E}_2\boldsymbol{E}\boldsymbol{S}\boldsymbol{E}^T\boldsymbol{E}_2^T,$$

it is enough to show that

$$\boldsymbol{E}_2\boldsymbol{E} = \boldsymbol{E}\boldsymbol{E}_1.$$

First, note that $\boldsymbol{K} = \boldsymbol{B}_2^{\frac{1}{2}}\boldsymbol{E}^{-T}\boldsymbol{B}_1^{-\frac{1}{2}}$ is unitary. Thus, we have (10). Finally, we have

$$\begin{aligned}\boldsymbol{E}_2\boldsymbol{E} &= \boldsymbol{B}_2^{\frac{1}{2}}\left(\boldsymbol{B}_2^{-\frac{1}{2}}\boldsymbol{A}_2\boldsymbol{B}_2^{-\frac{1}{2}}\right)^{\frac{1}{2}}\boldsymbol{B}_2^{-\frac{1}{2}}\boldsymbol{E} \\ &= \boldsymbol{B}_2^{\frac{1}{2}}\left(\boldsymbol{B}_2^{-\frac{1}{2}}\boldsymbol{A}_2\boldsymbol{B}_2^{-\frac{1}{2}}\right)^{\frac{1}{2}}\boldsymbol{B}_2^{\frac{1}{2}}\boldsymbol{B}_2^{-1}\boldsymbol{E} \\ &= \boldsymbol{E}\boldsymbol{B}_1^{\frac{1}{2}}\left(\boldsymbol{B}_1^{-\frac{1}{2}}\boldsymbol{A}_1\boldsymbol{B}_1^{-\frac{1}{2}}\right)^{\frac{1}{2}}\boldsymbol{B}_1^{\frac{1}{2}}\boldsymbol{E}^T\boldsymbol{B}_2^{-1}\boldsymbol{E} \\ &= \boldsymbol{E}\boldsymbol{B}_1^{\frac{1}{2}}\left(\boldsymbol{B}_1^{-\frac{1}{2}}\boldsymbol{A}_1\boldsymbol{B}_1^{-\frac{1}{2}}\right)^{\frac{1}{2}}\boldsymbol{B}_1^{-\frac{1}{2}} = \boldsymbol{E}\boldsymbol{E}_1.\end{aligned}$$

■

APPENDIX E

In this appendix we proof that if the identity matrix $\boldsymbol{I}$ is on the geodesic φ between $\boldsymbol{A}$ and $\boldsymbol{B}$, then, the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ commute and they have the same eigenvectors.

Proof: if $\boldsymbol{I}$ is on the geodesic $\varphi(t)$, then there exist some $t_0 \in (0,1)$ (if $t_0 = 0$ or $t_0 = 1$ the result is trivial) such that:

$$\varphi(t_0) = \boldsymbol{A}^{\frac{1}{2}} \left(\boldsymbol{A}^{-\frac{1}{2}} \boldsymbol{B} \boldsymbol{A}^{-\frac{1}{2}}\right)^{t_0} \boldsymbol{A}^{\frac{1}{2}} = \boldsymbol{I}$$

Now, consider $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$ the eigenvalue-decomposition of $\boldsymbol{A}$. By multiplying both from the right and the left, we have

$$\boldsymbol{A}^{\frac{1}{2}} \left(\boldsymbol{A}^{-\frac{1}{2}} \boldsymbol{B} \boldsymbol{A}^{-\frac{1}{2}}\right)^{t_0} \boldsymbol{A}^{\frac{1}{2}} = \boldsymbol{I}$$

$$\left(\boldsymbol{A}^{-\frac{1}{2}} \boldsymbol{B} \boldsymbol{A}^{-\frac{1}{2}}\right)^{t_0} = \boldsymbol{A}^{-1}$$

By raising to the power of $\frac{1}{t_0}$, and then multiplying both from the right and the left again, we have

$$\boldsymbol{A}^{-\frac{1}{2}} \boldsymbol{B} \boldsymbol{A}^{-\frac{1}{2}} = \boldsymbol{A}^{-\frac{1}{t_0}}$$

$$\Rightarrow \boldsymbol{B} = \boldsymbol{A}^{1-\frac{1}{t_0}} = \boldsymbol{V}\boldsymbol{\Lambda}^{1-\frac{1}{t_0}}\boldsymbol{V}^T$$

Thus, $\boldsymbol{B}$ has the same eigenvectors as $\boldsymbol{A}$ and they commute

$$\boldsymbol{A}\boldsymbol{B} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{\Lambda}^{1-\frac{1}{t_0}}\boldsymbol{V}^T = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{\Lambda}^{1-\frac{1}{t_0}}\boldsymbol{V}^T$$

$$= \boldsymbol{V}\boldsymbol{\Lambda}^{1-\frac{1}{t_0}}\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T = \boldsymbol{B}\boldsymbol{A}$$

■

APPENDIX F
RIEMANNIAN MEAN ALGORITHM

Algorithm 2: Riemannian Mean for SPD Matrices as Presented in [6].

Input: a set of SPD matrices $\{\boldsymbol{P}_i \in \mathcal{M}\}_{i=1}^N$.

Output: the Riemannian mean matrix $\overline{\boldsymbol{P}}$.

1) Compute the initial term $\overline{\boldsymbol{P}} = \frac{1}{N}\sum_{i=1}^N \boldsymbol{P}_i$
2) **do**
   a) Compute the Euclidean mean in the tangent space: $\overline{\boldsymbol{S}} = \frac{1}{N}\sum_{i=1}^N \mathrm{Log}_{\overline{P}}(\boldsymbol{P}_i)$
   b) Update $\overline{\boldsymbol{P}} = \mathrm{Exp}_{\overline{P}}\left(\overline{\boldsymbol{S}}\right)$
   c) **while** $\left\|\overline{\boldsymbol{S}}\right\|_F > \epsilon$ where $\|\cdot\|_F$ is the Frobenius norm.

REFERENCES

[1] S. Sra and R. Hosseini, "Conic geometric optimization on the manifold of positive definite matrices," *SIAM J. Optim.*, vol. 25, no. 1, pp. 713–739, 2015.

[2] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on Riemannian manifolds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1713–1727, Oct. 2008.

[3] O. Freifeld, S. Hauberg, and M. J. Black, "Model transport: Towards scalable transfer learning on manifolds," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2014, pp. 1378–1385.

[4] R. Bergmann, J. H. Fitschen, J. Persch, and G. Steidl, "Priors with coupled first and second order differences for manifold-valued image processing," *J. Math. Imag. Vision*, vol. 60, no. 9, pp. 1459–1481, 2018.

[5] X. Pennec, P. Fillard, and N. Ayache, "A Riemannian framework for tensor computing," *Int. J. Comput. Vis.*, vol. 66, no. 1, pp. 41–66, 2006.

[6] 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.

[7] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 137–144.

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

[9] M. Lorenzi, N. Ayache, and X. Pennec, "Schilds ladder for the parallel transport of deformations in time series of images," in *Proc. Biennial Int. Conf. Inf. Process. Med. Imag.*, 2011, pp. 463–474.

[10] H. J. Kim, N. Adluru, B. B. Bendlin, S. C. Johnson, B. C. Vemuri, and V. Singh, "Canonical correlation analysis on Riemannian manifolds and its applications," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 251–267.

[11] S. Bonnabel and R. Sepulchre, "Riemannian metric and geometric mean for positive semidefinite matrices of fixed rank," *SIAM J. Matrix Anal. Appl.*, vol. 31, no. 3, pp. 1055–1070, 2009.

[12] P. Zanini, M. Congedo, C. Jutten, S. Said, and Y. Berthoumieu, "Transfer learning: A Riemannian geometry framework with applications to brain-computer interfaces," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 5, pp. 1107–1116, May 2018.

[13] A. Karbalayghareh, X. Qian, and E. R. Dougherty, "Optimal Bayesian transfer learning," *IEEE Trans. Signal Process.*, vol. 66, no. 14, pp. 3724–3739, Jul. 2018.

[14] R. Bhatia, *Positive Definite Matrices*. Princeton, NJ, USA: Princeton Univ. Press, 2009.

[15] M. Moakher, "A differential geometric approach to the geometric mean of symmetric positive-definite matrices," *SIAM J. Matrix Anal. Appl.*, vol. 26, no. 3, pp. 735–747, 2005.

[16] A. Schlogl, J. Kronegg, J. Huggins, and S. Mason, "19 evaluation criteria for BCI research," *Toward Brain-Computer Interfacing*. Boca Raton, FL, USA: CRC Press 2007.

[17] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *Ann. Statist.*, vol. 36, pp. 1171–1220, 2008.

[18] R. Ferreira, J. Xavier, J. P. Costeira, and V. Barroso, "Newton method for Riemannian centroid computation in naturally reductive homogeneous spaces," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2006.

[19] M. Naeem, C. Brunner, R. Leeb, B. Graimann, and G. Pfurtscheller, "Seperability of four-class motor imagery data using independent components analysis," *J. Neural Eng.*, vol. 3, no. 3, pp. 208–216, 2006.

[20] L. van der Maaten and G. Hinton, "Visualizing data using T-SNE," *J. Mach. Learn. Res.*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[21] A. L. Goldberger *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000, doi: 10.1161/01.CIR.101.23.e215. [Online]. Available: http://circ.ahajournals.org/content/101/23/e215.full PMID:1085218

[22] B. Kemp, A. H. Zwinderman, B. Tuk, H. A. Kamphuisen, and J. J. Oberye, "Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG," *IEEE Trans. Biomed. Eng.*, vol. 47, no. 9, pp. 1185–1194, Sep. 2000.

[23] J. Shin *et al.*, "Open access dataset for EEG+ NIRS single-trial classification," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 10, pp. 1735–1745, Oct. 2017.

Or Yair (S'15), photograph and biography not available at the time of publication.

Mirela Ben-Chen, photograph and biography not available at the time of publication.

Ronen Talmon (M'11), photograph and biography not available at the time of publication.