Tangent space spatial filters for interpretable and efficient Riemannian classification

To cite this article before publication: Jiachen Xu *et al* 2020 *J. Neural Eng.* in press https://doi.org/10.1088/1741-2552/ab839e

Tangent space spatial filters for interpretable and efficient Riemannian classification

Jiachen Xu*, Moritz Grosse-Wentrup, and Vinay Jayaram

***Abstract*—*Objective*: Methods based on Riemannian geometry have proven themselves to be good models for decoding in brain-computer interfacing (BCI). However, these methods suffer from the curse of dimensionality and are not possible to deploy in high-density online BCI systems. In addition, the lack of interpretability of Riemannian methods leaves open the possibility that artifacts drive classification performance, which is problematic in the areas where artifactual control is crucial, e.g., neurofeedback and BCIs in patient populations. *Approach*: We rigorously proved the exact equivalence between any linear function on the tangent space and corresponding derived spatial filters. Upon which, we further proposed a set of dimension reduction solutions for Riemannian methods without intensive optimization steps. The proposed pipelines are validated against classic common spatial patterns and tangent space classification using an open-access BCI analysis framework, which contains over seven datasets and 200 subjects in total. At last, the robustness of our framework is verified via visualizing the corresponding spatial patterns. *Main results*: Proposed spatial filtering methods possess competitive, sometimes even slightly better, performances comparing to classic tangent space classification while reducing the time cost up to 97% in the testing stage. Importantly, the performances of proposed spatial filtering methods converge with using only four to six filter components regardless of the number of channels which is also cross validated by the visualized spatial patterns. These results reveal the possibility of underlying neuronal sources within each recording session. *Significance*: Our work promotes the theoretical understanding about Riemannian geometry based BCI classification and allows for more efficient classification as well as the removal of artifact sources from classifiers built on Riemannian methods.**

***Index Terms*—Brain-computer interface, Spatial filters, Riemannian geometry, Interpretability, Meta-analysis.**

I. Introduction

Brain-computer interfaces (BCIs), in particular imagery-based BCIs, are well known, if not infamous, for their sensitivity to noise and their low signal-to-noise ratio. Over the past decades, many methods have been invented in order to derive features from the raw signal data that are predictive of user intention. However, as the electroencephalogram (EEG) is highly sensitive to both neural and non-neural signals, optimizing setups for predictive accuracy was insufficient. Rather, it was necessary to be able to confirm that any classifier was both predictive and based purely on brain-derived features. These divergent requirements spurred the field to develop in two different directions: spatial filtering and Riemannian manifold techniques.

Spatial filters are linear combinations of channel activity that reconstruct a single (neural or non-neural) source with certain desired properties. Initially, these weightings were computed via physical or neurophysiological models [1]. However, it was quickly discovered that data-driven spatial filters could lead to features that reflect robust differences in brain activity. By optimizing for variance [2, 3] or independence [4, 5], or even searching for filters that maximize the difference between multiple types of intention [6, 7], many different sorts of spatial filters can be computed. In order to verify that the reconstructed signal comes from the brain, the spatial patterns can be plotted corresponding to those filters on the scalp.

Beginning with common spatial patterns (CSP), there has been a large body of literature dedicated to finding algorithms that optimally reconstruct source activity based on a given criterion. For differences between two classes, CSP has proven itself to be robust and easy to implement (for a more exhaustive review, see [8]). More recently, methods have been developed to find sources that track a continuous variable of interest [9]. One major difficulty that was recognized early on is that, while it is relatively simple to generate appropriate spatial filters for data that is already recorded, the application of these filters to new data is often confounded by the highly non-stationary nature of the EEG signal. Filters that persist across multiple recording sessions, or filters that work on multiple subjects, are both open areas of research. Some groups look at different criteria to derive robust filters [10, 11], others use more probabilistic techniques [12, 13], and still others consider options like sparsity [14, 15] or looking at patches of channels [16]. Each of the aforementioned techniques has shown its value in solving an aspect of the spatial filtering problem, but they often require very different approaches to solve the ensuing optimization problems, and it is hard to decide which one may be most appropriate for a given situation. A further issue is the inefficient use of data. Using the same data to compute optimal spatial features and then a classifier runs the risk of overfitting since the same data is used in both steps. However, if the training data is split into disjoint sets for spatial filtering computation and model fitting, then the amount that can be used to fit either model is necessarily reduced. Hence, in either case, the data cannot be efficiently utilized.

Nevertheless, spatial filtering remains a crucial method in applications where artifacts are of great concern. In particular,

Asterisk indicates corresponding author.

*J. Xu and M. Grosse-Wentrup are with the Faculty of Computer Science, University of Vienna, Hörlgasse 6, 1090 Vienna, Austria (email: jxu1809@gmail.com, moritzgw@ieee.org).

V. Jayaram is with the Department of Empirical Inference, Max Planck Institute for Intelligent Systems, Max-Planck-Ring 4, 72076 Tübingen, Germany (email: vjayaram@tuebingen.mpg.de).

The source code for this manuscript is available in *www.xujiachen.com* .

it is crucial for neurofeedback studies. When the goal is to give feedback on neural activity, there must be a way of ensuring that the model which reconstructs a source of interest from the original signal only uses brain data to do so. This requirement invalidates many black-box machine learning methods, such as random forests [17].

Outside of this sphere, methods based on Riemannian geometry have been gaining momentum as a model for robust classification for performance-optimized BCIs. Thanks to work in differential geometry, metrics for computing the distance between sensor covariance matrices have been discovered that are invariant to many common sorts of noise found in the electroencephalogram [18]. These methods can be translated into algorithms for finding classifiers that are far more robust to noise across a variety of contexts [19]. In particular, the approaches that use tangent space projection [20] have been shown to out-perform most other conventional methods in a recent meta-analysis [21]. Two major downsides, however, are their high computational complexity and their interpretation. Because these methods work in the space of sensor covariance matrices, their size scales quadratically with the number of sensors. To tackle this problem, which impedes application in high-density BCI systems, recent research has focused on investigating dimension reduction techniques by leveraging sparsity [22, 23], embedding adjacent samples [24, 25, 26], measures of manifold linearity [27], measures of affinity-weighted similarity [28], or based on an information criterion [29]. A generic framework can be referred to [30]. These methods can significantly reduce the computation necessary for online application, however they all require solving complex optimization problems.

Further, the issue of interpretation is a significant while unsolvable problem. As of now, it is not possible to determine what parts of a signal are being used to build a tangent space classifier, and therefore these can only be used in artifact-sensitive contexts when paired with an artifact detection pipeline or other artifact cleaning methods.

In our paper, we show the following contributions: that it is possible to find sets of spatial filters that describe a linear function in the Riemannian tangent space, and further that this space has a fundamental relationship to common spatial patterns. Via this approach, the full literature of linear machine learning methods can be used for spatial filtering, instead of requiring a different optimization for each regularization of interest. Using this connection, it is possible to visualize the sources that a tangent space classifier uses, and thereby to identify artifact sources used for classification and remove them via orthogonal projection. Finally, we show in offline comparison that using spatial filters derived via this approach significantly out-performs common spatial patterns and can even, in low-data situations, out-perform the tangent space function they are derived from. We improve upon the work presented in [31] by introducing rigorous proofs for the validity of the proposed techniques as well as an expanded set of experiments.

II. Background

Riemannian manifold-based classification methods (hereafter abbreviated Riemannian methods) can often seem difficult to understand. For convenience, we include this section that reviews our notation and the basic operations of Riemannian methods, as well as a short review of the mathematics behind spatial filtering.

A. Preliminary and Notations

We notate the the raw sensor data as $\mathbf{X} \in \Re^{C \times N \times T}$, where C, N and T represents the number of channels (electrodes), samples (length of each trial) and trials respectively. We represent the data of channel c (with $c \in \{1, \cdots, C\}$) as $\mathbf{X}_c$. In addition, the data from the t-th trial (with $t \in \{1, \cdots, T\}$) are expressed as $\mathbf{X}^t$. Similarly, we use $(\cdot)^t$ to express the variables derived from $\mathbf{X}^t$. Moreover, the covariance matrices computed from $\mathbf{X}$, i.e., the points lying on the manifold, are denoted as $\mathbf{C} \in \Re^{C \times C \times T}$. The Fréchet mean, a generalization of the standard arithmetic mean to other spaces, of the manifold points set $\mathbf{C}$ is expressed as $\mathbf{C}^m$. In the following section, we use $\mathbf{A}$ to denote any symmetric positive definite (SPD) matrix, for which the following property holds true: $\mathbf{v}^T \mathbf{A} \mathbf{v} > 0, \forall \mathbf{v} \neq \mathbf{0}$.

We next describe some common operations for manipulating points on the symmetric positive definite (SPD) manifold. Firstly, $\lambda(\mathbf{A})$ is used to express the vector of eigenvalues of $\mathbf{A}$. Next, the logarithm and exponential as well as the power of p for an SPD matrix are namely defined as:

$$\begin{aligned} \mathrm{Logm}(\mathbf{A}) &= \mathbf{V} \log(\mathbf{D}) \mathbf{V}^{\mathsf{T}} \\ \mathrm{Expm}(\mathbf{A}) &= \mathbf{V} \exp(\mathbf{D}) \mathbf{V}^{\mathsf{T}} \\ \mathbf{A}^p &= \mathbf{V} \mathbf{D}^p \mathbf{V}^{\mathsf{T}}, \ p \in \Re \text{ and } p \neq 0, \end{aligned} \tag{II.1}$$

where $\mathbf{D}$ is the diagonal eigenvalue matrix of $\mathbf{A}$, i.e., $\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^T$, $\log(\cdot)$ and $\exp(\cdot)$ as well as $(\cdot)^p$ represent taking the logarithm, exponential and power of p elementwise for a matrix, and $\mathbf{V}$ is the orthogonal matrix of eigenvectors. Please note that p can also be a fraction, e.g., $p = \frac{1}{2}$ means the square root and $p = -\frac{1}{2}$ denotes the inverse square root.

At last, since the vectorization of an SPD matrix is also frequently employed to reduce the computational complexity, it is defined as below:

$$\begin{aligned} \mathrm{vec}(\mathbf{A}) &= [\alpha_{1,1}\mathbf{A}_{1,1}, \cdots, \alpha_{i,j}\mathbf{A}_{i,j}, \cdots, \alpha_{C,C}\mathbf{A}_{C,C}] \\ &\in \Re^{1 \times \frac{C(C+1)}{2}}, \text{ where } 1 \leq j \leq i \leq C. \\ \alpha_{i,j} &= \Big[1 \text{ if } i = j, \sqrt{2} \text{ else}\Big] \end{aligned} \tag{II.2}$$

An overview of the frequent notations is shown in Table II.1.

B. Riemannian Manifold based Methods

The Riemannian classification framework emerged in the BCI field around one decade ago [19]. Since then, it has attracted increasing attention due to its state-of-the-art performance [32, 33]. In this section, we briefly introduce the Riemannian methods from the practical viewpoint. For a more mathematically exhaustive treatment of Riemannian manifolds, please refer to [34].

TABLE II.1: The List of Frequent Notations

Data Related Variables
$\mathbf{X} \in \Re^{C\times N\times T}$ – Bandpass filtered trialwise data
$\mathbf{C} \in \Re^{C\times C\times T}$ – Covariance matrices on the manifold
$\mathbf{S} \in \Re^{C\times C\times T}$ – Points on the tangent space
$\mathbf{F} \in \Re^{C\times C}$ – Spatial filters with full rank
Operators
$(\cdot)^t$ – Variables from the data of t-th trial
$(\cdot)^m$ – Fréchet mean
$\overline{(\cdot)}$ – Arithmetic mean
$\text{vec}(\cdot)$ – Vectorizing SPD matrices
$\lambda(\cdot)$ – The eigenvalue vector of a matrix
$\log(\cdot)$ – Taking logrithm elementwise
$\text{Logm}(\cdot)$ /$\text{Expm}(\cdot)$ – Logrithm/Exponential for a matrix based on $\mathbf{I}$
$\text{Logm}_{\mathbf{A}}(\cdot)$ /$\text{Expm}_{\mathbf{A}}(\cdot)$ – Logrithm/Exponential for a matrix based on $\mathbf{A}$

1) Riemannian Metric and Distance: As the most common proxy for measuring the discriminability of data points, distances between points are usually defined by a preselected metric, normally the Euclidean one. While this metric can also be used with SPD matrices, it is incapable of adequately capturing the structure of SPD matrices, leading to certain undesirable effects such as the swelling effect [35]. It means that simply vectorizing covariance matrices and fitting them into a linear classifier often work poorly.

In order to take advantage of the structure inherent to covariance matrices, it is desirable to have a metric that generalizes the properties of the Euclidean metric in standard vector spaces to the SPD manifold. Therefore, the affine-invariant Riemannian metric is proposed [36] and based on this metric, the corresponding distance between two matrices is defined as:

$$d^2_{\text{AIRM}}(\mathbf{A},\mathbf{B}) = \left\|\log\left(\lambda\left(\mathbf{A}^{-\frac{1}{2}}\mathbf{B}\mathbf{A}^{-\frac{1}{2}}\right)\right)\right\|_2^2, \tag{II.3}$$

where $\|\cdot\|_2$ represents the L2 norm. Based on the chosen metric, the expression for the mean of a set of matrices is defined as:

$$\mathbf{C}^m = \arg\min_{\mathbf{A}\in\mathbf{C}} \sum_{t=1}^{T} d^2_{\text{AIRM}}\left(\mathbf{A},\mathbf{C}^t\right), \tag{II.4}$$

where $\mathbf{A} \in \Re^{C\times C}$ and $\mathbf{C} = \left[\mathbf{C}^1,\cdots,\mathbf{C}^T\right] \in \Re^{C\times C\times T}$.

If $\mathbf{C}^m$ is globally unique, then it is named as the Fréchet mean of the set of SPD matrices $\mathbf{C}$.

2) Tangent Space: One inconvenience introduced by the Riemannian metric is that the distance between two manifold points cannot be derived via simple subtraction and norm computation, as it can with the Euclidean metric. In order to treat SPD matrices in a manner identical to traditional feature vectors, we adopt the tangent space mapping constructed at a chosen reference point. After mapping onto the tangent space, these points can be treated as standard vectors.

To transform points from the manifold to the tangent space at a point and vice versa, the so-called logarithmic and exponential maps are used. Under the affine-invariant Riemannian metric, the logarithm and exponential function pair at a point $\mathbf{A}$ are formulated as following:

$$\begin{aligned}\mathbf{S}^t &= \text{Logm}_{\mathbf{A}}\left(\mathbf{C}^t\right)\\ \mathbf{C}^t &= \text{Expm}_{\mathbf{A}}\left(\mathbf{S}^t\right),\end{aligned} \tag{II.5}$$

where $\mathbf{S}^t$ is the projected point lying on the tangent space, $\mathbf{C}^t$ is the original manifold point and the operation of $\text{Logm}_{\mathbf{B}}(\mathbf{A})$ and $\text{Expm}_{\mathbf{B}}(\mathbf{A})$, i.e., the logarithm and exponential of $\mathbf{A}$ based on another SPD matrix $\mathbf{B}$, is defined as [19, 36]:

$$\begin{aligned}\text{Logm}_{\mathbf{B}}(\mathbf{A}) &= \mathbf{B}^{\frac{1}{2}}\text{Logm}\left(\mathbf{B}^{-\frac{1}{2}}\mathbf{A}\mathbf{B}^{-\frac{1}{2}}\right)\mathbf{B}^{\frac{1}{2}}\\ \text{Expm}_{\mathbf{B}}(\mathbf{A}) &= \mathbf{B}^{\frac{1}{2}}\text{Expm}\left(\mathbf{B}^{-\frac{1}{2}}\mathbf{A}\mathbf{B}^{-\frac{1}{2}}\right)\mathbf{B}^{\frac{1}{2}}\end{aligned} \tag{II.6}$$

To further simplify operations on the tangent space, the projected points are usually vectorized. Note that this procedure does not alter the location or norm of the points, it simply makes them easier to notate and use. We denote these vectors as tangent vectors and formulate them as follows:

$$\overrightarrow{s^t} = \text{vec}\left(\mathbf{S}^t\right) \in \Re^{\frac{C(C+1)}{2}\times 1}, \tag{II.7}$$

where $\overrightarrow{s_t}$ is the tangent vectors of t-th trial.

After obtaining the set of tangent vectors, standard machine learning algorithms can be applied.

3) Pros and Cons of Riemannian Methods: Riemannian methods is famous for their rich feature space and robustness to outliers. In particular, Jayaram *et al.* [21] have compared Riemannian methods and standard processing pipelines over more than 200 subjects and showed that Riemannian methods are, on average, superior to many other conventional methods.

One major pitfall of these methods, however, is their sensitivity to the number of channels. As shown in Eq. (II.7), the dimension of the tangent vectors increases quadratically with the number of channels C. In addition, the computational complexity of the eigenvalue decomposition for matrices grows cubically. Hence, it becomes infeasible to apply Riemannian methods on data sets with a large number of channels. In addition, since the full covariance matrix is utilized for classification, interpreting the contribution from each channel can be a challenge. Therefore, the application of Riemannian methods is still restricted to low-channel situations where interpretability and real-time are of lesser importance.

C. Spatial Filtering

Thanks to the novel metric function, Riemannian methods present the-state-of-the-art performances in the BCI field. But, the EEG signal is vulnerable to artifacts and noise which highly affect the data quality. To remove these artifacts and noise while reducing the computational complexity, spatial filtering techniques are often used. Since the projection of the underlying neuronal sources to the EEG electrodes can be modeled as a linear transformation [37], with the appropriate projection, it is possible to recover the activity of specific parts of the brain. This both increases signal quality and provides a convenient signal for neuro-feedback.

The various types of spatial filters has been intensively reviewed in [38], among which we would notably mention Common Spatial Patterns (CSP) [6, 7, 39], which are impactful in the BCIs fields over the past decades. CSP aim at extracting the signal sources by maximizing the variance ratio between two conditions. Thus, the filter components are extracted via the Generalized Eigenvalue Decomposition (GED) between

the within-class arithmetic mean, i.e., $\overline{\mathbf{C}^{(+)}}$ and $\overline{\mathbf{C}^{(-)}}$. Moreover, Blankertz *et al* also pointed out in [40] that CSP can also be interpreted as maxmizing the variance ratio between common and discriminative activity of both conditions, which are as defined below:

$$\begin{aligned}\mathbf{C}_d &= \overline{\mathbf{C}^{(+)}} - \overline{\mathbf{C}^{(-)}}\text{: discriminative activity}\\ \mathbf{C}_c &= \overline{\mathbf{C}^{(+)}} + \overline{\mathbf{C}^{(-)}}\text{: common activity}\end{aligned} \tag{II.8}$$

Thus, the spatial filter matrix of CSP, i.e., $\mathbf{F}_{\text{CSP}}$, can be extracted via GED($\mathbf{C}_d$,$\mathbf{C}_c$), which means:

$$\mathbf{F}_{\text{CSP}} = \text{GED}(\mathbf{C}_d,\mathbf{C}_c) \tag{II.9}$$

III. Methods

Utilizing the smallest dimension to achieve the highest discriminability is always the ideal when designing a feature extraction algorithm. Although the features extracted from standard Riemannian methods are of high quality, they are hamstrung by the curse of dimensionality and a lack of interpretability. It is striking that, when reviewing these two factors which impede the application of Riemannian methods, spatial filtering techniques seem to be the remedy. The arguments are two-fold: First, reducing the dimensionality of the covariance matrices decreases computation time drastically. Second, the associated spatial patterns of spatial filtering enables us to verify what aspects of the recorded signal are being used by the classifier. Hence, how to leverage the spatial filtering technique in the standard Riemannian methods becomes an interesting question.

Inspired by this idea, in this section, we first propose a novel spatial filter extraction algorithm in which we approximate a linear function on the Riemannian tangent space by a set of spatial filters, which render that function much less computationally intensive and also more understandable. We support the proposed algorithm by rigorous mathematical proofs (see supplementary). Moreover, by adopting this approximation idea, a simplified regression-like classification method is also proposed. Subsequently, CSP is proven to be a special case of the proposed tangent space spatial filtering. We validate our theoretical findings experimentally via the validation setup proposed in [21].

Mathematically-oriented readers are invited to begin below; readers more interested in a practical understanding may refer to Section III-D.

A. The Approximation of Standard Riemannian Methods via Spatial Filtering

For the tangent space based Riemannian methods, the decision function on the tangent space determines the classification accuracy. To unify spatial filtering and tangent space-based methods, one option is to find filters that can preserve this function. For simplicity, we consider linear functions in the tangent space:

$$\hat{y}_t = \overrightarrow{w}^{\mathsf{T}}\overrightarrow{s^t} + b \in \Re^{1\times 1},\ \forall t = 1,...,T, \tag{III.1}$$

where $\overrightarrow{w}$ is the weight vector on the tangent space, b is the corresponding intercept and $\hat{y}_t$ represents the predicted label from the decision function for t-th trial. Please note that the intercept term b in Eq. (III.1) will be omitted in the subsequent proof, as if we can show the equivalence of the projection, then we can simply use the same intercept as was in the original function. In addition, we would also like to emphasize that practically, this intercept can be safely ignored in the prediction for binary labels when using data set with balanced training data and a linear classifier on the tangent space, because b will be roughly equal to zero due to tangent vectors being centered when projecting to the AIRM tangent space around the Fréchet mean. Hence, based on the definition of the AIRM [36], the inner product on the tangent space can be expressed as the function of manifold points as derived below:

$$\hat{y}_t = \text{Tr}(\text{Logm}_{\mathbf{C}^m}(\mathbf{C}^w) \bullet \text{Logm}_{\mathbf{C}^m}(\mathbf{C}^t)),\ \forall t = 1,...,T, \tag{III.2}$$

where $\mathbf{C}^w$ is the weight covariance matrix re-projected onto the manifold via the exponential map $\text{Expm}_{\mathbf{C}^m}(\text{unvec}(\overrightarrow{w}))$.

Similarly, the approximated predicted labels from all the manifold points spatially filtered by $\mathbf{F}$ are as expressed below:

$$\begin{aligned}y_t^{\text{approx}}|_{\mathbf{F}} = \text{Tr}(&\text{Logm}_{\mathbf{F}^{\mathsf{T}}\mathbf{C}^m\mathbf{F}}(\mathbf{F}^{\mathsf{T}}\mathbf{C}^w\mathbf{F}) \bullet\\ &\text{Logm}_{\mathbf{F}^{\mathsf{T}}\mathbf{C}^m\mathbf{F}}(\mathbf{F}^{\mathsf{T}}\mathbf{C}^t\mathbf{F})),\end{aligned} \tag{III.3}$$

where $y_t^{\text{approx}}|_{\mathbf{F}}$ is denoted as y_t^{approx} thereafter for the convenience of notation and based on the property of AIRM we can easily know that the new Fréchet mean of filtered manifold points is the filtered Fréchet mean, i.e., $(\mathbf{C}^m)_{\text{new}} = \mathbf{F}^{\mathsf{T}}\mathbf{C}^m\mathbf{F}$.

The optimal scenario for extracting the spatial filter matrix $\mathbf{F}$ is that this spatial filter matrix $\mathbf{F}$ can perfectly reconstruct the decision function. Hence, in the next subsection, we provide the steps to find the optimal solution of $\mathbf{F}$.

B. Optimal Spatial Filter Extraction from the Tangent Space

Naively, the goal of spatial filter extraction is to find a filtering matrix that maximally reconstructs the tangent space function, which is shown as follows:

$$\mathbf{F}_K^* = \underset{\mathbf{F}_K\in\Re^{C\times K}}{\arg\min}\sum^{T}(\hat{y}_t - y_t^{\text{approx}}|_{\mathbf{F}_K})^2, \tag{III.4}$$

where $\mathbf{F}_K^*$ is the optimal filter matrix composed of K spatial filter components from the full filter matrix $\mathbf{F}$.

After substituting $\hat{y}_t$ (Eq. (III.2)) and y_t^{approx} (Eq. (III.3)) into Eq. (III.4), the objective function of the optimization becomes rather complicated. Considering that $\mathbf{F}_K$ is a subset of $\mathbf{F}$, we first focus on the structure of y_t^{approx} to see whether it can be simplified when $\mathbf{F}_K$ is full rank.

Assuming that $\mathbf{C}^m$ and $\mathbf{C}^w$ can be jointly diagonalized by a properly chosen $\mathbf{F}$, the matrix multiplications in Eq. (III.3) can be remarkably simplified. Without loss of generality, we further assume that this properly chosen $\mathbf{F}$ diagonalizes one of the two matrices. If we choose that matrix to be $\mathbf{C}^m$, the approximation becomes much simpler:

$$y_t^{\text{approx}} = \text{Tr}(\text{Logm}_{\mathbf{I}}(\mathbf{D}^w) \bullet \text{Logm}_{\mathbf{I}}(\mathbf{F}^{\mathsf{T}}\mathbf{C}^t\mathbf{F})), \tag{III.5}$$

where $\mathbf{D}^w$ is the filtered weight matrix.

From here, we make one major assumption that the filtering matrix $\mathbf{F}$ approximately diagonalizes all $\mathbf{C}^t$. If this assumption holds , i.e., $\mathbf{F}^\mathsf{T}\mathbf{C}^t\mathbf{F}$ is a diagonally dominant matrix for all t, based on the Gershgorin circle theorem [41] we know that

$$\lambda\left(\mathbf{F}^\mathsf{T}\mathbf{C}^t\mathbf{F}\right) \approx \text{diag}\left(\mathbf{F}^\mathsf{T}\mathbf{C}^t\mathbf{F}\right) = \mathbf{D}^t, \tag{III.6}$$

where $\mathbf{D}^t$ represents the diagonal matrix which only contains the diagonal elements of $\mathbf{F}^\mathsf{T}\mathbf{C}^t\mathbf{F}$. Moreover, since $\mathbf{F}^\mathsf{T}\mathbf{C}^t\mathbf{F}$ is diagonally dominant, then following approximation can be inferred:

$$\mathbf{F}^\mathsf{T}\mathbf{C}^t\mathbf{F} \approx \mathbf{D}^t \Rightarrow \text{Logm}(\mathbf{F}^\mathsf{T}\mathbf{C}^t\mathbf{F}) \approx \text{Logm}(\mathbf{D}^t) \tag{III.7}$$

After applying the approximation in Eq. (III.7) into Eq. (III.5), y_t^{approx} can be simplified as:

$$y_t^{\text{approx}} \approx \text{Tr}\left(\text{Logm}(\mathbf{D}^w)\text{Logm}(\mathbf{D}^t)\right) = \log(\overrightarrow{d^w})^\mathsf{T}\log(\overrightarrow{d^t}), \tag{III.8}$$

where $\overrightarrow{d^{(\cdot)}}$ represents the diagonal vector of $\mathbf{D}^{(\cdot)}$.

We reiterate that one primary assumption in the above simplification is that all $\mathbf{C}^t$ are roughly jointly diagonal, which is a very strong assumption. However, there is evidence for this in the fact that the projection of the physiological sources in the EEG signal to the electrodes is linear: Since the head moves very little with respect to the electrodes within a session, we can assume that the mixing (and hence unmixing) matrices stay relatively constant, even if the actual variances are non-stationary. In the supplementary materials, we also provide a simulation analysis to show that for small numbers of sources, approximate joint diagonalization is a reasonable assumption.

The key step that enables the simplification from Eq. (III.3) to Eq. (III.8) is the simultaneous diagonalization of $\mathbf{C}^w$ and the whitening of $\mathbf{C}^m$. The generalized eigenvalue decomposition (GED) conveniently solves both goals:

$$\mathbf{C}^w\mathbf{F} = \mathbf{C}^m\mathbf{F}\mathbf{D} \Leftrightarrow \begin{cases} \mathbf{F}^\mathsf{T}\mathbf{C}^m\mathbf{F} = \mathbf{I} \\ \mathbf{F}^\mathsf{T}\mathbf{C}^w\mathbf{F} = \mathbf{D} \end{cases}, \tag{III.9}$$

where the $\mathbf{F}$ is named as tangent space spatial filter (TSSF) and $\mathbf{D}$ is the corresponding eigenvalues. Importantly, to ensure that Eq. (III.9) holds, the order of $\mathbf{C}^m$ and $\mathbf{C}^w$ in the GED equation cannot be switched.

Now, since y_t^{approx} can be drastically simplified as long as $\mathbf{F}$ are extracted with the GED manner, when looking back to the objective function for extracting optimal filters, i.e., $\mathbf{F}_K^* = \underset{\mathbf{F}_K \in \Re^{C\times K}}{\arg\max} \sum^T (\hat{y}_t - y_t^{\text{approx}}|_{\mathbf{F}_K})^2$, the last remaining obstacle is the true predicted label $\hat{y}_t$. We next assert that the equivalence between $\hat{y}_t$ and y_t^{approx} holds under mild conditions, as seen in *Theorem 1*. The full proof can be found in the supplementary materials.

Theorem 1: Equivalence between true and approximated decision function

$$\hat{y}_t \equiv y_t^{\text{approx}}, \text{ iff } \mathbf{F} \text{ is extacted via GED}(\mathbf{C}^w, \mathbf{C}^m) \text{ and with full rank.} \tag{III.10}$$

By leveraging this equivalence, the objective function in Eq. (III.4) can be reformulated as:

$$\mathbf{F}_K^* = \underset{\mathbf{F}_K \in \Re^{C\times K}}{\arg\min} \sum^T (y_t^{\text{approx}}|_{\mathbf{F}} - y_t^{\text{approx}}|_{\mathbf{F}_K})^2 \tag{III.11}$$

Unfortunately, assuming no prior knowledge is known about $\mathbf{F}_K^*$, Eq. (III.11) still seems to have no closed-form solution, as the operation Logm () is heavily involved in the expression $y_t^{\text{approx}}|_{\mathbf{F}_K}$. Instead of adopting sophisticated algorithms to tackle this intricate problem as in other related research, we decide to leverage the power of Theorem 1 to solve this problem with only slightly sacrificing the optimality. Our solution is assuming that $\mathbf{F}_K^*$ is in the subspace of the extracted full-rank spatial filter $\mathbf{F}$ such that the equivalence can persist. Hence, this optimal set of spatial filters, i.e., $\mathbf{F}_K^*$ will only be a conditional optimal solution for the optimization problem Eq. (III.11). Further, since the $\mathbf{F}_K$ is now known as the subset of $\mathbf{F}$ which is extracted from the GED($\mathbf{C}^w, \mathbf{C}^m$), the optimization problem in Eq. (III.11) is then equivalent to the problem of ordering the columns of $\mathbf{F}$.

When observing the result in Eq. (III.8) which states that as long as the filtered input data $\mathbf{F}^\mathsf{T}\mathbf{C}^t\mathbf{F}$ is roughly diagonal, the linear functions in the Riemannian tangent space can be approximated by linear functions of the log-variances of the filtered data. More importantly, the coefficients of this approximated linear function are simply the log-eigenvalues after the GED is solved, i.e., $\log(\overrightarrow{d}^w)$. Thus, standard techniques for determining the most important variables in a linear regression problem can be used. For simplicity's sake, we use the absolute values of the regression coefficients as markers of their importance to the function.

a) Intuitive Explanation: One common and effective technique across domains is whitening data. By decorrelating the different channels, constructed features are often more distinct and predictive. However, whitening has a fundamental flaw, in that there are arbitrarily many whitening matrices that are possible since the covariance of whitened data is invariant to rotations. One explanation for the finding above is that the GED can be decomposed into a whitening transform and a subsequent rotation. The whitening is with respect to the data, and the rotation is chosen based on the weight matrix. Therefore this technique can be considered a particular choice of data whitening that simultaneously preserves the information of a function in the tangent space.

C. Classification based on the TSSF

As a feature extraction method, spatial filtering always requires a classifier to deal with the processed features, which often requires an extra optimization step. TSSF can also be utilized with such a conventional manner but thanks to the simplification as shown in Eq. (III.8), this secondary training can also be skipped for TSSF, which further accelerates the decoding.

From Eq. (III.8), we notice that this function is actually a linear regressor using the log-eigenvalues of the GED as the regression weights. Therefore, we can directly input the filtered data into this regressor to obtain the predicted value.

This method is named as one-step classification in our paper, and the ordinary way to classify the data is named as two-steps classification, i.e., filtering and classifying.

Example 1: Tangent Space Spatial Filter - The Generalization of CSP

During the proof for the justification of TSSF, we also find a strong relationship between TSSF and CSP, which is that CSP is a simplified TSSF with using LDA as classifier on the tangent space. First of all, TSSF are obtained via solving the $\text{GED}(\mathbf{C}^w, \mathbf{C}^m)$ as described in Section III-B. Moreover, the equivalent solution of eigenvectors can also be extracted by solving $\text{GED}(\mathbf{S}^w, \mathbf{C}^m)$, i.e.,:

$$\begin{aligned} \mathbf{F}, \mathbf{D} &= \text{GED}(\mathbf{C}^w, \mathbf{C}^m) \\ \mathbf{F}, \text{Logm}\,(\mathbf{D}) &= \text{GED}(\mathbf{S}^w, \mathbf{C}^m), \end{aligned} \tag{III.12}$$

where $\mathbf{S}^w = \text{Logm}_{\mathbf{C}^m}(\mathbf{C}^w)$ and the proof of Eq. (III.12) can be referred in the supplementary materials.

Furthermore, when the classifier on the tangent space is specified as the Fisher linear LDA classifier [42], the weight vector on the tangent space is as expressed in Eq.(III.13) [42].

$$\begin{aligned} \overrightarrow{w}_{\text{LDA}} &= \mathbf{S}_{within}^{-1}(\mu^{(+)} - \mu^{(-)}) \\ \mathbf{S}_{within} &= \sum_{a\in\{+,-\}} \sum_{t\in^{(a)}} \left(\overrightarrow{s^t} - \mu^{(a)}\right)\left(\overrightarrow{s^t} - \mu^{(a)}\right)^{\mathsf{T}}, \end{aligned} \tag{III.13}$$

where $\mu^{(a)}, a \in \{+,-\}$ are the within class mean for the tangent vectors and $\mathbf{S}_{within}$ is the within scatter matrix.

Under the special case that the $\mathbf{S}_{within}$ is equal to the identity matrix $\mathbf{I}$, the weight vector of LDA classifier is expressed as:

$$\overrightarrow{w}_{\text{LDA}} = \mu^{(+)} - \mu^{(-)} \in \Re^{\frac{C(C+1)}{2}\times 1} \tag{III.14}$$

Based on the reverse operation of $\text{vec}\,(\cdot)$ (Eq. (II.2)), the equivalent formulation of Eq. (III.14) in matrix format is:

$$\mathbf{S}^{w_{\text{LDA}}} = \overline{\mathbf{S}^{(+)}} - \overline{\mathbf{S}^{(-)}}, \tag{III.15}$$

where $\overline{\mathbf{S}^{(\cdot)}}$ is the arithmetic within-class mean for projected points on the tangent space. Moreover, assuming the special situation holds in which the between-class Euclidean mean difference of the covariances is the exponential transform of the between-class Euclidean mean difference of tangent space points, i.e.,

$$\overline{\mathbf{C}^{(+)}} - \overline{\mathbf{C}^{(-)}} = \text{Expm}_{\mathbf{C}^m}\left(\overline{\mathbf{S}^{(+)}} - \overline{\mathbf{S}^{(-)}}\right) \tag{III.16}$$

Combining the special LDA classifier with the conclusion drawn from Eq. (III.12), the solution of TSSF becomes:

$$\begin{aligned} \text{GED}(\mathbf{C}^{w_{\text{LDA}}}, \mathbf{C}^m) &\overset{\text{Eq. (III.12)}}{=} \text{GED}(\mathbf{S}^{w_{\text{LDA}}}, \mathbf{C}^m) \\ &\overset{\text{Eq. (III.15)}}{=} \text{GED}(\overline{\mathbf{S}^{(+)}} - \overline{\mathbf{S}^{(-)}}, \mathbf{C}^m) \\ &\overset{\text{Eq. (III.12\& III.16)}}{=} \text{GED}(\overline{\mathbf{C}^{(+)}} - \overline{\mathbf{C}^{(-)}}, \mathbf{C}^m), \end{aligned} \tag{III.17}$$

where $\text{GED}(\mathbf{A}, \mathbf{B})$ in above equations represents the corresponding eigenvectors, i.e., $\mathbf{V} = \text{GED}(\mathbf{A}, \mathbf{B})$ and $\mathbf{AV} = \mathbf{BV\Gamma}$ ($\mathbf{\Gamma}$ is the matrix of generalized eigenvalues).

In addition, if we further replace the Fréchet mean $\mathbf{C}^m$ in Eq. (III.17) with arithmetic mean $\overline{\mathbf{C}^m}$, we will have:

$$\begin{aligned} \text{GED}(\mathbf{C}^{w_{\text{LDA}}}, \mathbf{C}^m) &\overset{\text{Eq. (III.17)}}{\Rightarrow} \text{GED}(\overline{\mathbf{C}^{(+)}} - \overline{\mathbf{C}^{(-)}}, \mathbf{C}^m) \\ &\Rightarrow \text{GED}(\overline{\mathbf{C}^{(+)}} - \overline{\mathbf{C}^{(-)}}, \overline{\mathbf{C}^m}) \\ &\overset{\substack{\text{Scaling} \\ \text{invariance}}}{\Rightarrow} \text{GED}(\overline{\mathbf{C}^{(+)}} - \overline{\mathbf{C}^{(-)}}, \overline{\mathbf{C}^{(+)}} + \overline{\mathbf{C}^{(-)}}) \end{aligned} \tag{III.18}$$

By combining the definition of CSP from the discriminative perspective as described in Eq. (II.8) and the equivalence as shown in Eq. (III.18), we are able to conclude the relationship between CSP and TSSF as:

$$\begin{aligned} \mathbf{F}_{\text{TSSF}} &\overset{\text{Def.}}{\Rightarrow} \text{GED}(\mathbf{C}^{w_{\text{LDA}}}, \mathbf{C}^m) \\ &\overset{\text{Eq. (III.18)}}{\Rightarrow} \text{GED}(\overline{\mathbf{C}^{(+)}} - \overline{\mathbf{C}^{(-)}}, \overline{\mathbf{C}^{(+)}} + \overline{\mathbf{C}^{(-)}}) \\ &\overset{\text{Eq. (II.8)}}{=} \text{GED}(\mathbf{C}_d, \mathbf{C}_c) \\ &\overset{\text{Eq. (II.9)}}{\Rightarrow} \mathbf{F}_{\text{CSP}} \end{aligned} \tag{III.19}$$

Namely, CSP is the representation of TSSF when LDA is chosen as the classifier on the tangent space, and the within-class scatter matrix is assumed to be the identity. One important caveat is the exponential relationship of class mean subtraction, as shown in the Eq. (III.17), which is not necessarily true.

One related work is [43], in which Barachant *et al* replaced the arithmetic mean with the Fréchet mean in CSP. Since the Fréchet mean is a much better proxy of common activities across trials, the proposed Riemannian CSP is a far better approximation of LDA in the tangent space, and Barachant *et al* also show increased performance and robustness with this alteration [43].

D. Summary of the extraction and application of TSSF

For practitioners interested in using the proposed TSSF framework, in this subsection, we summarize its procedures which can be categorized as the extraction and application of TSSF. To link each algorithm's description with its pseudocode, we adopt the abbreviation that A1-1 denotes the Step-1 of Algorithm 1.

1) Extraction of TSSF: To extract the TSSF, the input data should be bandpass filtered and epoched into trials. Moreover, the choice of the linear model on the tangent space should also be defined beforehand. Subsequently, the covariance matrices are estimated based on the input trialwise EEG signal and their Fréchet mean is computed to use as the reference point for the tangent space projection (A1-1). After finding the Fréchet mean, all covariance matrices are projected onto the tangent space and vectorized into tangent vectors (A1-2). Afterward, the chosen linear model is fitted by them and the weight vector can be obtained (A1-3). Furthermore, it is reshaped into a symmetric matrix and mapped back onto the manifold via the exponential transform (A1-4). Next, the full-rank filter matrix of TSSF and the regression coefficients for one-step classification are obtained by solving the GED problem (A1-5) and both are sorted based on the absolute value of the logarithm of the eigenvalues in the descending order (A1-6 and A1-7). At last, based on the predefined number of filter

components, the first K components of the sorted filter matrix with full-rank are extracted, and same applies to the regression coefficients (A1-8).

Algorithm 1 Extraction of TSSF

Data: Bandpass filtered trialwise data $\mathbf{X} \in \Re^{C\times N\times T}$, loss function for linear model L

Result: TSSF and regression coefficients with K components: $\mathbf{F}_K \in \Re^{C\times K}$, $\overrightarrow{\beta}_K \in \Re^{k\times 1}$

begin

1. Compute the covariance matrices and the Fréchet mean: $\mathbf{C}^t = \mathbf{X}^t(\mathbf{X}^t)^\mathsf{T}, \forall t \in [1,\cdots,T]$, $\mathbf{C}^m = \underset{\mathbf{A}\in\mathbf{C}}{\arg\min} \sum_{t=1}^{T} d^2_{\text{AIRM}}(\mathbf{A}, \mathbf{C}^t)$
2. Compute tangent vectors: $\overrightarrow{s^t} = \text{vec}\left(\text{Logm}_{\mathbf{C}^m}(\mathbf{C}^t)\right), \forall t \in [1,\cdots,T]$ and $\mathbf{s} = [\overrightarrow{s^1},\cdots,\overrightarrow{s^T}] \in \Re^{\frac{C(C+1)}{2}\times T}$
3. Fit linear model: $\overrightarrow{w} = \text{argmin}_{\overrightarrow{w}} L(\mathbf{s}, \overrightarrow{w}) \in \Re^{\frac{C(C+1)}{2}\times 1}$
4. Project weights onto manifold: $\overrightarrow{w} \overset{\text{unvec}(\cdot)}{\Rightarrow} \mathbf{S}^w \overset{\text{Expm}_{\mathbf{C}^m}(\cdot)}{\Rightarrow} \mathbf{C}^w \in \Re^{C\times C}$
5. Solve the Generalized Eigenvalue Decomposition (GED) problem: $\overrightarrow{d}, \mathbf{V} = \text{GED}(\mathbf{C}^w, \mathbf{C}^m)$
6. Get the sorted index based on the value of $\overrightarrow{d}$: inds = sort $\left(\left|\log(\overrightarrow{d})\right|\right)$
7. Obtain the sorted TSSF and regression coefficients: $\mathbf{F} = \mathbf{V}[:, \text{inds}] \in \Re^{C\times C}$, $\overrightarrow{\beta} = \log(\overrightarrow{d}[\text{inds}]) \in \Re^{C\times 1}$
8. Extract the first K components: $\mathbf{F}_K = \mathbf{F}[:, :K]$, $\overrightarrow{\beta}_K = \overrightarrow{\beta}[:K]$

end

2) Application of TSSF: Once the TSSF are extracted, the next step is to apply these filters to the trialwise data (A2-1). Subsequently, there are three types of features which can be generated from the filtered data: the log-variance of filtered data (A2-2.a)), the diagonal vector of the logarithm of the filtered covariance matrices (A2-2.b)) and the tangent vector of the filtered covariance matrices (A2-2.c)). These three types of features and their descriptions, as well as the corresponding abbreviations, are summarized in Table III.1.

Formulation	Description	Abbreviation
$\log\left(\text{diag}(\widetilde{\mathbf{C}}^t_{\perp\mathbf{F}})\right)$	Log-variance	Log-var
$\text{diag}\left(\text{Logm}_{\mathbf{I}}(\widetilde{\mathbf{C}}^t_{\perp\mathbf{F}})\right)$	Diagonal of logarithm of covariance matrices	Diag. log-cov
$\text{vec}\left(\text{Logm}_{\widetilde{\mathbf{C}}^m_{\perp\mathbf{F}}}(\widetilde{\mathbf{C}}^t_{\perp\mathbf{F}})\right)$	Logarithm of covariance matrices	Log-cov

TABLE III.1: Summary of classifiable features

After obtaining the features, as described in Section III-C, two possible classification algorithms can be applied: one-step classification and two-steps classification. One thing that should be noted is that one-step classification is only applicable for the diagonal elements based features, namely features from (A2-2.b)) and (A2-2.c)). For the one-step classification, the inner product between regression coefficients and features are computed, and the label is taken as the sign of the result in binary classification problems (A2-2.a).i)). For the two-steps classification, a second classifier is chosen and fitted with the features from the training set (A2-2.b).i)). As such, test data can be classified by this second classifier.

Algorithm 2 Feature generation and classification

Data: Test trialwise data $\mathbf{X} \in \Re^{C\times N\times T}$, Second classifier Clf_2 if needed

Result: TSSF and regression coefficients: $\mathbf{F}_K \in \Re^{C\times K}$, $\overrightarrow{\beta}_K \in \Re^{K\times 1}$

begin

1. Filter the test data: $\widetilde{\mathbf{X}}_{\perp\mathbf{F}_K} = \mathbf{F}_K^\mathsf{T}\mathbf{X} \in \Re^{K\times N}$
2. Compute features (several options are provided, only choose one):
   a). $\overrightarrow{e} = \log\left(\text{var}\left(\widetilde{\mathbf{X}}_{\perp\mathbf{F}_K}\right)\right) \in \Re^{K\times 1}$
   b). $\overrightarrow{e} = \text{diag}\left(\text{Logm}\left(\text{Cov}(\widetilde{\mathbf{X}}_{\perp\mathbf{F}_K})\right)\right) \in \Re^{K\times 1}$
   c). $\overrightarrow{e} = \text{vec}\left(\text{Logm}\left(\text{Cov}(\widetilde{\mathbf{X}}_{\perp\mathbf{F}_K})\right)\right) \in \Re^{K\times 1}$
3. Return label (several options are provided, only choose one):
   a). One-step classificaiton (only applicaple for features from 2.a) or 2.b)):
      i). $\hat{y} = \text{sgn}(\overrightarrow{\beta}_K^\mathsf{T}\overrightarrow{e})$
   b). Two-steps classificaiton (applicaple for all features):
      i). Use a set of $\overrightarrow{e}$ from training datasets to fit a second classifier Clf_2
      ii). Use the fitted classifier to classify the testing datasets and obtain the predicted label.

return predicted label

end

E. Experimental Setup

Now that we have shown the theoretical validity of TSSF, we move on to our experimental results. We base our experimental setup on an open-source benchmark – *Mother of all BCI Benchmark (MOABB)* [21]. After that, we first fix the experimental paradigm as left- versus right-hand motor imagery because the corresponding neurophysiological knowledge, as well as the activated neuronal sources, are well studied. Furthermore, the analysis is restricted to the α- and β-bands (8Hz $\sim$ 32Hz) based on neurophysiological knowledge. Also, all channels are utilized except for the electrooculography (EOG) channel. Based the chosen paradigm, we tried to adopt all eight available datasets in the MOABB, as summarized in Table III.2; however, the dataset *BNCI 2014-004* is excluded from the analysis (marked in red in Table III.2) due to having only three electrodes.

After the bandpass filtering, the covariance matrices are first estimated from the trial-wise data via the empirical covariance estimator. Subsequently, three algorithms are employed to generate feature: CSP, TSSF, and standard Riemannian tangent space methods. TSSF based features are further subdivided into three types depending on the degree of approximation as summarized in Table III.1, and two methods, namely one-step and two-steps classification as described in Section III-C. The difference between them is the choice of the second classifier: either fitting a new classifier after spatial filter generation (two-steps) or employing the log-eigenvalues from the GED

TABLE III.2: Overview of all adopted datasets with left- versus right-hand motor imagery (MI) paradigm. The dataset marked in red color has only 3 channels and is hence excluded from this analysis.

Dataset Name	#Channels	#Subjects	#Sessions	Citations
BNCI 2014-001	22	9	2	[44]
BNCI 2014-004	3	9	5	[45]
Cho et al. 2017	64	49	1	[46]
Munich MI	128	10	1	[2]
Physionet MI	64	109	1	[47]
Shin et al. 2017	25	29	3	[48]
Weibo et al. 2014	60	10	1	[49]
Zhou et al. 2016	14	4	3	[50]

solution as linear regression coefficients. These classification methods are summarized in Table III.3. For CSP and standard Riemannian features, the L2-regularized SVM classifier is adopted.

Name	First classifier	Second Classifier
One-step	L2 Regularized SVM	N/A
Two-steps	L2 Regularized SVM	L2 Regularized SVM

TABLE III.3: Summary of classifiers. For all regularized SVM listed above, the parameters are found by grid search [51].

The motivation of selecting a regularized SVM as the first classifier to generate weight vectors on the tangent space is inspired by the results from [21], in which the combination of regularized SVM and Riemannian methods has been validated as the best among all benchmarked pipelines. For choosing hyperparameters, a grid search [51] is employed to find the optimal value within the range from 0.01 to 100.

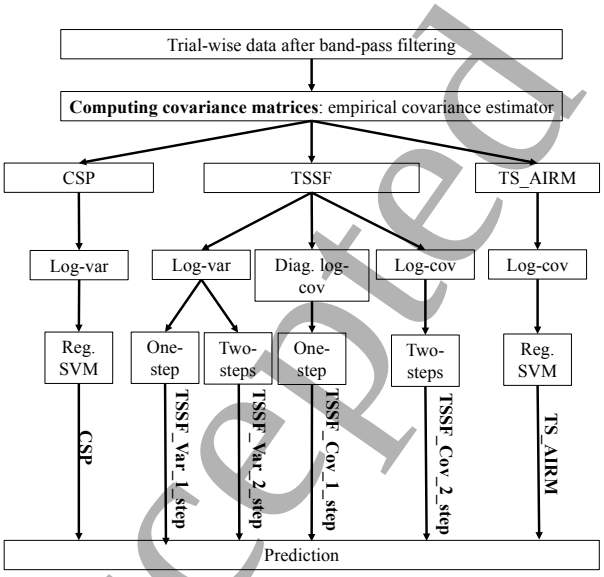


Fig. III.1: All tested pipelines in this paper. The annotated text above the line linked between classifiers and predictions is the abbreviation of the corresponding pipeline and Reg. SVM is the abbreviation of L2 regularized SVM.

For better understanding the difference among the multiple variants of TSSF based methods, CSP, and standard Riemannian methods, we summarize all the above steps into a flowchart (Fig. III.1). After the prediction, the scoring metric chosen by us is the ROC-AUC (receiver operating characteristic - area under the curve) metric, and these scores are computed via five-fold cross-validation within each session of every data set.

After obtaining scores from different pipelines, we analyze their statistical performance. In our work, two statistics, the p-value and the effect size, are adopted to compare the proposed TSSF against CSP as well as the full Riemannian approach. The p-value for the one-sided test is computed across sessions and subjects but within each data set, the null hypothesis of which is that the median accuracy of using one pipeline is not larger than using another pipeline. The effect size is measured by the standardized mean difference (SMD) between the accuracies of the two compared methods. Further details about these statistical tests can be found in [21].

IV. Results

To comprehensively assess the performance of the proposed TSSF, three aspects are considered in this paper: the quality of the filtered feature, the interpretability revealed from associated spatial patterns, and the computational time.

Of these three perspectives, feature quality is the only indicator which can be analyzed in a purely quantitative manner through the classification accuracy. Therefore, in this section, we exclusively analyze the performance of feature quality. Although it can be argued that the results of computational time can be analyzed in a quantitive way, i.e., by exhaustively comparing the simulation results of computational time, we more concerned with the theoretical, computational complexity analysis since the latter is more general than the simulation results. Therefore, interpretability and computational time are both left until the discussion, as the results are more qualitative and require more context to be properly interpreted.

As a typical indicator of feature quality, the classification accuracies are chosen to be compared as a way of assessing which features are most informative. In subsequent subsections, we begin with a comparison of all proposed classification pipelines over all the datasets, to see whether any of them consistently outperform the rest. The results are shown in Figure IV.1.

A. Statistical performance across datasets

We select three typical cases of applying spatial filters: two, six, or twelve spatial filters. By observing Fig. IV.1a we can notice that even when only applying two filters, the p-value of the comparison between all TSSF-based pipelines and *CSP* are highly significant, and the effect sizes are moderate. Moreover, in the comparisons with the full Riemannian method, the *TSSF_Cov_2_step* even significantly outperforms the full Riemannian method, albeit with a small effect size (0.23).

When increasing the filter number to 6, as shown in Fig. IV.1b, the performance of all TSSF-based pipelines continues to surpass *CSP*. Surprisingly, *TSSF_Var_2_step* also shows significantly better results than the full Riemannian method *TS_AIRM*, though again with a tiny effect size (0.08). In addition, performance begins to differ among the various

(a) Applying the first two filters

(b) Applying the first six filters

(c) Applying the first twelve filters

Fig. IV.1: Statistical comparison of the classification accuracies from different pipelines. Parameters: effect size t (standardized mean difference) and p-value p are computed within each dataset. In each block, the statistics are computed based on the null-hypothesis that the median accuracy of row method is not larger than the column method. The green block means there exists an overall significant result across all datasets. The red block means there exist contradictory results, i.e., the overall one-tailed results show significance, but the effect size is not positive. Furthermore, the number in parentheses next to the p-values represents that the percentage of datasets in which significance is reported. The meaning of each label can be referred from Table. III.1 and Fig. III.1

TSSF-based pipelines. After increasing the number to 12 (Fig. IV.1c), although one more TSSF-based pipeline significantly outperforms *TS_AIRM*, the differences among the TSSF-based pipelines also further enlarge.

Observing and comparing these figures from a macro perspective, we can discover several trends: First, CSP is constantly outperformed by all Riemannian based methods. Second, the performances of TSSF-based methods tend to differ from each other, only at large numbers of filters. Third, the difference in performance between one-step and two-steps methods also enlarges as the filter number increases.

B. Performance w.r.t. the number of applied filters in a single data set

In addition, we look at how the performance changes as a function of the number of applied filters. As a meta-analysis here results in an enormous number of statistical tests on not very much data, we focus on this section of the analysis on a single dataset. For better reflecting the relationship between accuracy and number of applied filters, we choose the data set *Munich Motor Imagery*, which has the highest channels numbers (128). From Fig. IV.2 we first notice that the accuracies of all TSSF-based features converge to the performance of the standard Riemannian method with merely four filters while CSP needs around 20 filters to reach a stable plateau. Second, except for *TSSF_Var_1_step*, all other TSSF-based methods significantly outperform *CSP* whatever number of filters is used, as shown in Fig. IV.3. Lastly, for all log-variance based TSSF pipelines, their accuracy usually decreases when the number of filters continues to increase. Moreover, this fact can also be observed in all rest datasets, as shown in the supplementary materials.

In this section, we have comprehensively compared the quality of the features extracted from various ways, and confirmed two things: that Riemannian methods reliably outperform CSP, and that TSSF can approximate and sometimes even outperform standard Riemannian methods. As a spatial filtering method, however, the interpretability is always of the highest significance, especially for online purposes, because it is the only way that we can know whether reasonable signal sources are utilized. Moreover, the computational efficiency of the spatial filtering method is also vital because the online BCI system usually has a strict real-time requirement. Therefore, in the next section, we further discuss these two aspects.

V. Discussion

We have shown that spatial filters can be extracted from linear functions in the Riemannian tangent space, rendering Riemannian methods suitable for online use even in cases of over 100 channels. Moreover, we validate our approaches using over 220 subjects via an open-access toolkit [21] and show that our method is statistically indistinguishable from the full tangent space approach on average, and in some cases can significantly improve on it. Lastly, the idea of using

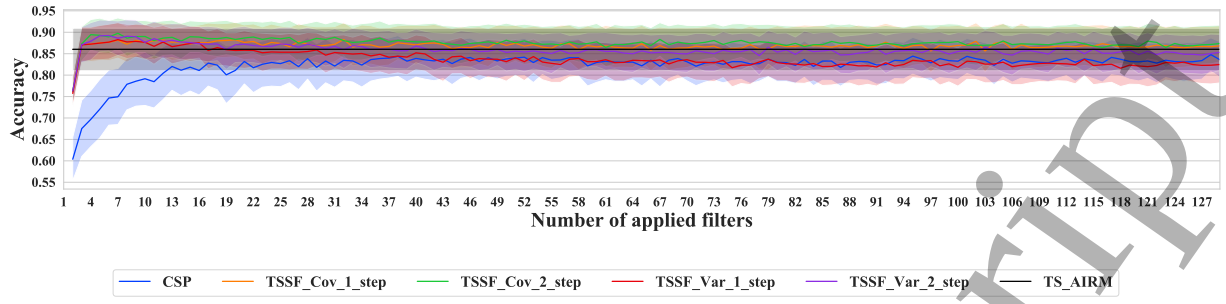


Fig. IV.2: Classification accuracy w.r.t. the number of applied filters within Munich Motor Imagery data set and the accuracies are computed across all subjects. The central line is the mean accuracy and the error band shows 68% confidence interval.

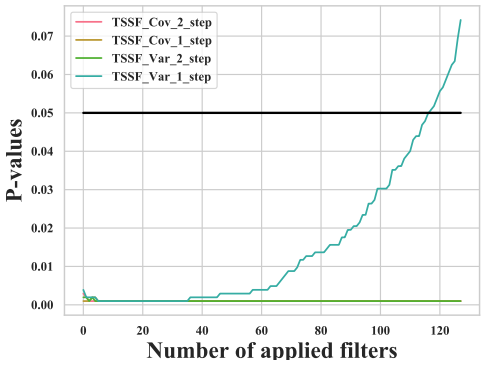


Fig. IV.3: The p-values from the statistical test between all TSSF-based pipelines and CSP w.r.t. to the number of applied filters. The chosen data set is *Munich Motor Imagery* and the null-hypothesis is that the median accuracy of TSSF-based methods is not larger than CSP. Significance threshold is set as 0.05, as indicated by the black straight line.

approximation to generate spatial filters eliminates the need for complicated optimization frameworks.

Another notable contribution of this paper is the proposal of one-step classification, which further reduces the computational time in the testing stage significantly. Subsequently, we analyze the associated spatial patterns of CSP and TSSF. Afterward, we discuss the signal sources as well as the robustness reflected from these patterns. We end our discussion with several suggestions regarding its usage and finally, a look towards the future work this result implies.

A. One-step classification

While one-step classification relies strongly on the assumption that the input points are roughly jointly diagonalizable, and hence that the proposed approximation holds, we have shown in practice that this appears to be the case for sufficiently small numbers of filters. What this suggests is that certain underlying sources can be extracted by static generalized spatial filters, while others do not correspond to static eigenvectors of the covariance matrices. If few enough filters are chosen, the resulting classifier is very close to the tangent space function, but as more are added, the approximation quality degrades. This explains the results in Fig. IV.2 in which the only classifier whose quality degraded as a function of filter number was the single-step log-variance classifier.

Another major benefit to using one-step classification is that it is a better use of training data. Current spatial filtering-based approaches to classifications need to either re-use data for both spatial filter and classifier fitting, or partition training data into disjoint sets, which reduces the quality of both solutions. When the approximation holds, one-step classification is a much more data-efficient solution.

B. One-step classification: Computational complexity analysis

<table>
<tr><th colspan="2">TSSF_Var_1_step (One-step)</th><th colspan="2">TS_AIRM (Two-steps)</th></tr>
<tr><th>Procedure</th><th>Computational complexity</th><th>Computational complexity</th><th>Procedure</th></tr>
<tr><td colspan="4">Preprocessing</td></tr>
<tr><td rowspan="3">$\widetilde{\mathbf{C}}^t_{\perp\mathbf{F}} = \mathbf{F}_K\mathbf{C}^t(\mathbf{F}_K)^T$
, where $\mathbf{F}_K \in \Re^{C\times K}$</td><td rowspan="3">$O(KC^2)$</td><td>$O(C^3)$</td><td>$\mathrm{ED}(\mathbf{C}^t)$
$\Updownarrow$
$\mathbf{V}^t\mathbf{D}^t(\mathbf{V}^t)^T = \mathbf{C}^t$
, where $\mathbf{V}^t \in \Re^{C\times C}, \mathbf{D}^t \in \Re^{C\times C}$</td></tr>
<tr><td>$O(C)$</td><td>$\mathrm{Logm}(\mathbf{D}^t)$</td></tr>
<tr><td>$O(C^3)$</td><td>$\mathbf{S}^t = \mathbf{V}^t\mathrm{Logm}(\mathbf{D}^t)(\mathbf{V}^t)^T$
, where $\mathbf{S}^t \in \Re^{C\times C}$</td></tr>
<tr><td colspan="4">Features</td></tr>
<tr><td>$f_t = \mathrm{diag}\left(\mathrm{Logm}_{\mathbf{1}}(\widetilde{\mathbf{C}}^t_{\perp\mathbf{F}})\right)$</td><td>$O(K^3)$</td><td rowspan="3">$O(\frac{C(C+1)}{2})$</td><td rowspan="3">$f_t = \mathrm{vec}(\mathbf{S}^t) \in \Re^{\frac{C(C+1)}{2}}$</td></tr>
<tr><td>or</td><td></td></tr>
<tr><td>$f_t = \log\left(\mathrm{diag}(\widetilde{\mathbf{C}}^t_{\perp\mathbf{F}})\right)$</td><td>$O(K)$</td></tr>
<tr><td colspan="4">Feature matrix for all trials</td></tr>
<tr><td>$[f_1, f_2, \cdots, f_T] \in \Re^{K\times T}$</td><td></td><td></td><td>$[f_1, f_2, \cdots, f_T] \in \Re^{\frac{C(C+1)}{2}\times T}$</td></tr>
<tr><td colspan="4">Classification</td></tr>
<tr><td colspan="2">Linear regression (testing)</td><td colspan="2">Regularized SVM (testing)</td></tr>
<tr><td>$y_t = \mathrm{diag}(\mathbf{D})[:K] \times f_t$
, where $\mathrm{diag}(\mathbf{D})[:K] \in \Re^K$</td><td>$O(K)$</td><td>$O(\frac{C(C+1)}{2})$</td><td>$y_t = w^T_{\mathrm{SVM}} \times f_t$
, where $w_{\mathrm{SVM}} \in \Re^{\frac{C(C+1)}{2}\times 1}$</td></tr>
</table>

Fig. V.1: Theorectical computational complexity analysis and comparison between one-step and two-step classification in the testing stage. Note: the practical complexity are usually smaller than the listed value due to the adoption of efficient algorithm. For instance, the matrix multiplication complexity is theoretically equal to $O(C^3)$ but usually between $O(C^{2.376})$ [52] and $O(C^3)$ in practice.

Considering that one major critique of Riemannian methods is their inability to scale to high numbers of channels, the computational complexity comparisons between the one-step classification framework and the full Riemannian tangent

space method are provided from both theoretical and experimental aspects

Data set	Method	Number of Applied Filters				#Channels
		4		6		
		Time (s)	Accuracy	Time (s)	Accuracy	
BNCI-2014001	CSP	0.0083±0.0013	0.83±0.18	0.0088±0.0016	0.85±0.15	22
	TSSF_Var_1_step	0.0036±0.0021	0.85±0.17	0.0038±0.0003	0.85±0.15	
	TS_AIRM	0.0435±0.0091	0.86±0.15	0.0435±0.0091	0.86±0.15	
Cho	CSP	0.0189±0.0032	0.68±0.19	0.0217±0.0033	0.70±0.17	64
	TSSF_Var_1_step	0.0120±0.0022	0.73±0.16	0.0159±0.0037	0.73±0.15	
	TS_AIRM	0.1353±0.0162	0.72±0.16	0.1353±0.0162	0.72±0.16	
Munich	CSP	0.0938±0.0119	**0.72±0.23**	0.0881±0.0139	**0.75±0.24**	128
	TSSF_Var_1_step	**0.1335±0.0215**	**0.87±0.13**	**0.1235±0.0094**	**0.88±0.13**	
	TS_AIRM	**3.1503±0.8547**	0.86±0.16	**3.1503±0.8547**	0.86±0.16	
Phy.	CSP	0.0201±0.0097	0.65±0.23	0.0152±0.0070	0.65±0.24	64
	TSSF_Var_1_step	0.0020±0.0015	0.68±0.24	0.0016±0.0002	0.67±0.24	
	TS_AIRM	0.0517±0.0121	0.65±0.26	0.0517±0.0121	0.65±0.26	
Shin	CSP	0.0155±0.0054	0.68±0.32	0.0184±0.0072	0.68±0.33	25
	TSSF_Var_1_step	0.0017±0.0007	0.67±0.34	0.0021±0.0002	0.66±0.35	
	TS_AIRM	0.0168±0.0085	0.65±0.35	0.0168±0.0085	0.65±0.35	
Weibo	CSP	0.0101±0.0018	0.81±0.16	0.0100±0.0008	0.82±0.18	60
	TSSF_Var_1_step	0.0054±0.0014	0.82±0.16	0.0056±0.0007	0.82±0.17	
	TS_AIRM	0.0696±0.0092	0.83±0.17	0.0696±0.0092	0.83±0.17	
Zhou	CSP	0.0096±0.0020	0.91±0.11	0.0101±0.0017	0.92±0.10	14
	TSSF_Var_1_step	0.0040±0.0009	0.91±0.09	0.0044±0.0005	0.90±0.10	
	TS_AIRM	0.0116±0.0011	0.92±0.09	0.0116±0.0011	0.92±0.09	

Fig. V.2: Comparison of classification accuracy and running time in the testing stage for three pipelines: *CSP*, *TSSF_Var_1_step* and *TS_AIRM*. The values for both accuracy and time are with the format of the mean $\pm$ the standard deviation, which is computed across all sessions within each data set. The comparisons with the largest contrast are noted as bold. The above numbers are obtained from computers with 64GB RAM and an 8-core CPU.

For a better understanding of experimental results, we first start with the theoretical analysis. As seen from Fig. V.1, standard Riemannian methods require operations with a computational complexity of either $O(C^3)$ or $O(\frac{C(C+1)}{2})$. For high numbers of channels, this can be difficult to do for real-time feedback. To verify that we ran a runtime analysis for all datasets including the *Munich Motor Imagery* data set which has over 100 channels. The results are shown in Fig. V.2 which indicates that standard Riemannian methods are slower than both CSP and TSSF based methods. In particular, the full Riemannian methods is 25 times slower than TSSF based methods with similar performance when observing the results from the data set with 128 channels. As for the accuracy comparison, the superiority of TSSF based methods is already validated in Fig. IV.3, in which *TSSF_Var_1_step* significantly outperforms *CSP* when using four or six filters.

In summary, by adopting the one-classification, it is no longer impossible to enjoy the robustness and excellent performance of Riemannian methods in an online BCI system with high-dimensional data. One additional advantage is: by using fewer features, the model suffers less risk from overfitting.

C. How robust is spatial filter order to artifacts?

Another important aspect of our work is the observation that this procedure allows one to easily validate the relevance of the features that a Riemannian classifier is using. By visualizing the spatial filters, it is easy to ensure that artifactual sources are not included in the classifier, which is of crucial importance when a BCI is used for neurofeedback.

As shown in Fig. V.3, in which only two filters are applied, the accuracies of TSSF based methods are clearly better than CSP based, especially for S1, S2, S7, S8, S9, and S10. Correspondingly, the CSP based patterns seem more patchy than TSSF based. As for the subjects that both methods have tied performance, i.e., S4 and S6, their spatial patterns are almost identical to each other.

We would, however, like to better understand where and how components that are not brain-related enter the spatial filters in these two methods. Therefore, we increase the filter number to ten and compare the spatial patterns from two contrasting subjects in order to verify how the methods are robust to artifacts, i.e., S2 and S4 who are with 55% and 6.3% contaminated trials respectively.

Observing the Fig. V.4, S2 Comp 0 and S2 Comp 1 of the spatial pattern of TSSF seems similar to S2 Comp 4 and S2 Comp 6 of CSP. As for the rest CSP patterns of S2, most of them appear to be artifacts while for TSSF patterns of S2, only Comp 8 and Comp 9 look slightly patchy while the rest of the patterns show strong activity around the sensorimotor cortex. Moreover, in S4's patterns, from Comp 0 to Comp 6, the results of CSP and TSSF reflect similar neuronal sources with a slightly different order. However, when looking at the last three patterns of both filters, artifactual sources appear. Overall, the ordering in TSSF is much more informative than in CSP, although in very low artifact scenarios they are similar.

Therefore, our conclusion is that the associated spatial patterns reflect more neurophysiologically explainable neural sources in TSSF. In contrast, CSP often gets distracted by artifacts when processing highly contaminated data. Moreover, when considering the patterns from the low-artifact subject S4, the patterns from both spatial filtering methods are almost identical, in particular for the first several patterns.

D. How many filters lead to optimal performance?

By enlarging the feature space, classification accuracy only increases when useful information is encoded within the additional features. For the case of spatial filtering, the most informative features are usually from the first several spatial filters and afterward, the features are no longer as informative as before, as indicated by the spatial patterns shown in Fig. V.4. Therefore, when applying only a few spatial filters, there always exists a positive relationship between the performance and number of applied filters. However, this positive relationship turns into a plateau when further increasing the filter numbers because the additional features are no longer as informative as before, and can even turn negative in cases of overfitting.

This initial positive relationship and the subsequent plateau always raises the question of the optimal number of spatial filters i.e., using the least number of filters to achieve a given level of performance. We argue that TSSF reliably requires less spatial filters than CSP in order to achieve the same level of performance based on the observed seven datasets. Moreover, the optimal filter number for TSSF seems independent from the number of channels which reflects a true biological set of sources conserved across all the data while it does not hold for

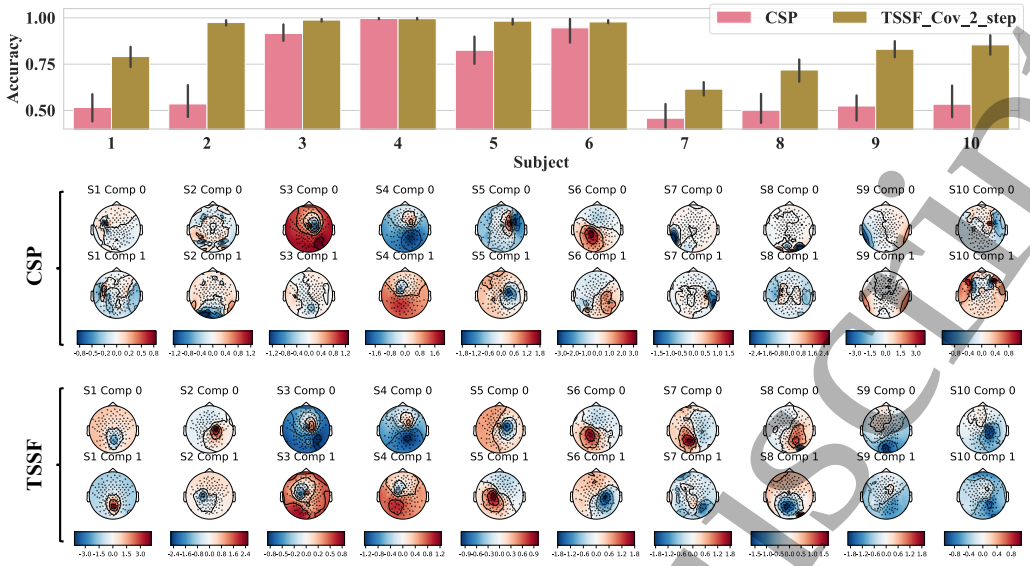
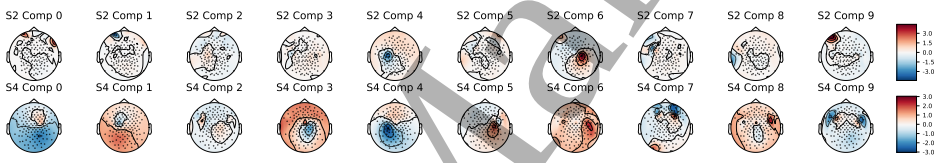
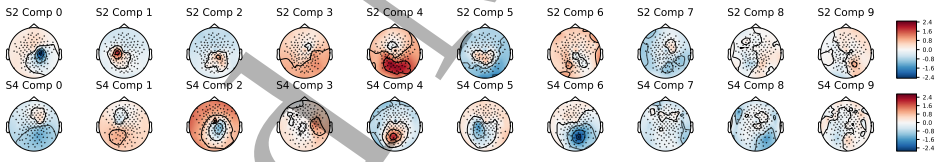


Fig. V.3: The classification accuracy and the associated spatial patterns of data set *Munich Motor Imagery* when applying the first two spatial filters for all subjects.



(a) Spatial patterns from CSP



(b) Spatial patterns from TSSF

Fig. V.4: The associated spatial patterns of data set *Munich Motor Imagery* when applying the first ten spatial filters for S2 and S4. The major conclusion drawn from these two figures is that the TSSF based patterns present a better ordering comparing to the CSP which means the CSP tends to be affected by artifacts.

CSP. In addition, these patterns also reflect the robustness of TSSF against the influence from ocular or muscular artifacts which CSP do not possess.

E. Suggestions for the usage of TSSF

After exhaustively benchmarking the TSSF based methods against conventional algorithms, we provide several suggestions to the reader who would like to use the TSSF method:

1) *Use the empirical covariance estimator when possible*: Since diagonal loading cannot be added during online use (as the filtered variances are used), high regularization runs the risk of degrading the approximation.
2) *Choose the Riemannian metric carefully*: Theoretical analysis is based on the assumption that the AIRM is utilized. A similar property remains to be validated for other Riemannian metrics.
3) *Choice of features*: Three types of features to be adopted as seen in Table III.1. As there is a trade-off between computational complexity and feature quality, the choice of features highly depends on the experimental environment. Nonetheless, based on our experience, $\operatorname{diag}\left(\operatorname{Logm}(\mathbf{F}^{\mathsf{T}}\mathbf{C}^{t}\mathbf{F})\right)$ is a good candidate when demanding high accuracy, while $\log\left(\operatorname{diag}(\mathbf{F}^{\mathsf{T}}\mathbf{C}^{t}\mathbf{F})\right)$ is a better choice for a strict real-time requirement.

F. Future Work

This paper covers the fundamental concept and proof of spatial filtering via the tangent space. However, there are still many interesting directions worthy of being explored later on. We discuss these possible directions from two levels, the extension of the scientific idea and of these proposed algorithms:

1) *Unsupervised dimensionality reduction and multi-class TSSF*: The nice performance of TSSF based methods with few components implies the existence of a low dimensional subspace where the brain projects. Therefore, it will be interesting to know how to find this subspace in an unsupervised fashion. Moreover, inspired by [53, 54], it will also be fascinating to investigate spatial filters for multiclass classification based on the TSSF.
2) *TSSF in comodulation manner*: Since the proposed TSSF is currently extracted in a regression-like manner, it will also be very worthwhile to explore whether we can also leverage continuous information encoded within the target variables, just like the source power comodulation (SPoC) method [9].
3) *Other choices of the first classifier*: The SVM based TSSF methods achieved a satisfying performance in this paper, but it will be very interesting to explore the influence of the classifier on the tangent space.
4) *Multiple frequency bands*: In this paper, the features are extracted from the joint μ and β band. It remains a mystery whether the ampler information induced by the filter bank TSSF will outperform current methods.

VI. Conclusion

Thanks to its impressive performance, the Riemannian manifold classification framework has seen an upsurge in interest in recent years. Historically, it has been hampered by various issues, namely that Riemannian methods scale poorly to high-density setups and are somewhat difficult to introspect.

To tackle these obstacles, we have proposed a set of methods based on the combination of spatial filtering techniques with Riemannian methods, because the former possess nice properties which Riemannian methods are lacking, such as low dimensionality and the visualization of signal sources (or associated spatial patterns). In order to further simplify the computation of the proposed idea, we have proved the rationality behind several variants of Riemannian features based on the approximation of the decision function on the tangent space. Moreover, we have also put forward one-step classification in order to simultaneously find a classifier and spatial filters. In addition, we would like to address again that the optimal subset of spatial filters that can be extracted by leveraging the proposed framework is not necessarily the globally optimal solution for the dimension reduction problem on the manifold of SPD matrices. Nevertheless, it is striking that these reduced spatial filters can converge to the performance of classic Riemannian methods robustly with using only four to six filters. We hope that this work will allow for the expansion of Riemannian geometry-based methods into more BCI applications, and that it might spur further development in both application and theory for this sort of interface.

Acknowledgment

We would like to thank Alexandre Barachant for his input in conceiving of this project.

References

[1] P. L. Nunez and K. L. Pilgreen. “The spline-Laplacian in clinical neurophysiology: a method to improve EEG spatial resolution.” In: *Journal of Clinical Neurophysiology: Official Publication of the American Electroencephalographic Society* 8.4 (1991), pp. 397–413.

[2] M. Grosse-Wentrup et al. “Beamforming in noninvasive brain-computer interfaces”. In: *IEEE Transactions on Biomedical Engineering* 56.4 (2009), pp. 1209–1219.

[3] A. Subasi and M. I. Gursoy. “EEG signal classification using PCA, ICA, LDA and support vector machines”. In: *Expert Systems with Applications* 37.12 (2010), pp. 8659–8666.

[4] R. Vigário et al. “Independent component approach to the analysis of EEG and MEG recordings”. In: *IEEE Transactions on Biomedical Engineering* 47.5 (2000), pp. 589–593.

[5] S. Makeig et al. “Blind separation of auditory event-related brain responses into independent components”. In: *Proceedings of the National Academy of Sciences* 94.20 (1997), pp. 10979–10984.

[6] Z. J. Koles, M. S. Lazar, and S. Z. Zhou. “Spatial patterns underlying population differences in the background EEG.” In: *Brain Topography* 2.4 (1990), pp. 275–84.

[7] H. Ramoser, J. Müller-Gerking, and G. Pfurtscheller. “Optimal spatial filtering of single trial EEG during imagined hand movement”. In: *IEEE Transactions on Rehabilitation Engineering* 8.4 (2000), pp. 441–446.

[8] F. Lotte and C. Guan. “Regularizing common spatial patterns to improve BCI designs: Unified theory and new algorithms”. In: *IEEE Transactions on Biomedical Engineering* 58.2 (2011), pp. 355–362.

[9] S. Dähne et al. “SPoC: A novel framework for relating the amplitude of neuronal oscillations to behaviorally relevant parameters”. In: *NeuroImage* 86 (2014), pp. 111–122.

[10] R. Martín-Clemente et al. “Information theoretic approaches for motor-imagery BCI systems: Review and experimental comparison”. In: *Entropy* 20.1 (2018), p. 7.

[11] W. Samek et al. “Robust spatial filtering with beta divergence”. In: *Advances in Neural Information Processing Systems 26*. 2013, pp. 1007–1015.

[12] H. Kang and S. Choi. “Bayesian multi-task learning for common spatial patterns”. In: *2011 International Workshop on Pattern Recognition in NeuroImaging*. IEEE, 2011, pp. 61–64.

[13] W. Wu et al. “A probabilistic framework for learning robust common spatial patterns”. In: *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society: Engineering the Future of Biomedicine, EMBC 2009*. IEEE, 2009, pp. 4658–4661.

[14] I. Onaran and N. F. Ince. “Extraction of spatially sparse common spatio-spectral filters with recursive weight elimination”. In: *6th International IEEE/EMBS*

Conference on Neural Engineering (NER). IEEE, 2013, pp. 1291–1294.

[15] F. Goksu, N. F. Ince, and A. H. Tewfik. "Sparse common spatial patterns in brain computer interface applications". In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. IEEE, 2011, pp. 533–536.

[16] C. Sannelli et al. "Ensembles of adaptive spatial filters increase BCI performance: an online evaluation". In: *Journal of Neural Engineering* 13.4 (2016), p. 046003.

[17] L. Fraiwan et al. "Automated sleep stage identification system based on time-frequency analysis of a single EEG channel and random forest classifier". In: *Computer Methods and Programs in Biomedicine* 108.1 (2012), pp. 10–19.

[18] M. Moakher. "A differential geometric aproach to the geometric mean of symmetric positive-definite matrices". In: *SIAM Journal on Matrix Analysis and Applications* 26.3 (2005), pp. 735–747.

[19] A. Barachant et al. "Riemannian geometry applied to BCI classification". In: *Latent Variable Analysis and Signal Separation*. Vol. 6365 LNCS. Springer, Berlin, Heidelberg, 2010, pp. 629–636.

[20] A. Barachant et al. "Classification of covariance matrices using a Riemannian-based kernel for BCI applications". In: *Neurocomputing* 112 (2013), pp. 172–178.

[21] V. Jayaram and A. Barachant. "MOABB: Trustworthy algorithm benchmarking for BCIs". In: *Journal of Neural Engineering* 15.6 (2018), p. 066011.

[22] F. P. Kalaganis et al. "A collaborative representation approach to detecting error-related potentials in SSVEP-BCIs". In: *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*. New York, New York, USA: ACM Press, 2017, pp. 262–270.

[23] F. P. Kalaganis et al. "A Riemannian geometry approach to reduced and discriminative covariance estimation in brain computer interfaces". In: *IEEE Transactions on Biomedical Engineering* 67.1 (2020), pp. 245–255.

[24] A. Goh and R. Vidal. "Clustering and dimensionality reduction on Riemannian manifolds". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2008, pp. 1–7.

[25] A. Qiu et al. "Manifold learning on brain functional networks in aging". In: *Medical image analysis* 20.1 (2015), pp. 52–60.

[26] X. Xie et al. "Bilinear regularized locality preserving learning on Riemannian graph for motor imagery BCI". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 26.3 (2018), pp. 698–708.

[27] S. Karygianni and P. Frossard. "Tangent-based manifold approximation with locally linear models". In: *Signal Processing* 104 (2014), pp. 232–247.

[28] M. T. Harandi, M. Salzmann, and R. Hartley. "From manifold to manifold: Geometry-aware dimensionality reduction for SPD matrices". In: *European Conference on Computer Vision*. Springer. 2014, pp. 17–32.

[29] P. L. C. Rodrigues et al. "Dimensionality Reduction for BCI classification using Riemannian geometry". In: *Proceedings of the 7th Graz Brain Computer Interface Conference 2017*. Graz, Austria, 2017.

[30] M. Harandi, M. Salzmann, and R. Hartley. "Dimensionality reduction on SPD Manifolds: The emergence of geometry-aware methods". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.1 (2018), pp. 48–62.

[31] J. Xu, M. Grosse-Wentrup, and V. Jayaram. "Interpretable Riemannian classification in brain-computer interfacing". In: *Proceedings of the 8th Graz Brain Computer Interface Conference 2019*. Graz, Austria, 2019, pp. 32–37.

[32] M. Congedo, A. Barachant, and R. Bhatia. "Riemannian geometry for EEG-based brain-computer interfaces: A primer and a review". In: *Brain-Computer Interfaces* 4.3 (2017), pp. 155–174.

[33] F. Yger, M. Berar, and F. Lotte. "Riemannian approaches in brain-computer interfaces: A review". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 25.10 (2017), pp. 1753–1762.

[34] W. M. Boothby. *An introduction to differentiable manifolds and Riemannian geometry*. Academic Press, 2003, p. 419.

[35] V. Arsigny et al. "Log-Euclidean metrics for fast and simple calculus on diffusion tensors". In: *Magnetic Resonance in Medicine* 56.2 (2006), pp. 411–421.

[36] X. Pennec, P. Fillard, and N. Ayache. "A Riemannian framework for tensor computing". In: *International Journal of Computer Vision* 66.1 (2006), pp. 41–66.

[37] S. Haufe et al. "On the interpretation of weight vectors of linear models in multivariate neuroimaging". In: *NeuroImage* 87 (2014), pp. 96–110.

[38] F. Lotte. "A tutorial on EEG signal-processing techniques for mental-state recognition in braincomputer interfaces". In: *Guide to Brain-Computer Music Interfacing*. London: Springer London, 2014, pp. 133–161.

[39] F. Lotte et al. "A review of classification algorithms for EEG-based brain-computer interfaces". In: *Journal of Neural Engineering* 4.2 (2007), R1–R13.

[40] B. Blankertz et al. "Optimizing spatial filters for robust EEG single-trial analysis". In: *IEEE Signal Processing Magazine* 25.1 (2008), pp. 41–56.

[41] S. A. Gershgorin. "Über die Abgrenzung der Eigenwerte einer Matrix". In: *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na* 6 (1931), pp. 749–754.

[42] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006, p. 738.

[43] A. Barachant et al. "Common spatial pattern revisited by Riemannian geometry". In: *2010 IEEE International Workshop on Multimedia Signal Processing, MMSP2010*. IEEE, 2010, pp. 472–476.

[44] M. Tangermann et al. "Review of the BCI competition IV". In: *Frontiers in Neuroscience* 6 (2012), p. 55.

[45] R. Leeb et al. "Brain-computer communication: Motivation, aim, and impact of exploring a virtual apartment". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 15.4 (2007), pp. 473–482.

[46] H. Cho et al. "EEG datasets for motor imagery brain–computer interface". In: *GigaScience* 6.7 (2017), gix034.

[47] A. L. Goldberger et al. "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals". In: *Circulation* 101.23 (2000), e215–e220.

[48] J. Shin et al. "Open access dataset for EEG+NIRS single-trial classification". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 25.10 (2016), pp. 1735–1745.

[49] W. Yi et al. "Evaluation of EEG oscillatory patterns and cognitive process during simple and compound limb motor imagery". In: *PLOS ONE* 9.12 (2014).

[50] B. Zhou et al. "A fully automated trial selection method for optimization of motor imagery based brain-computer interface". In: *PLOS ONE* 11.9 (2016).

[51] F. Pedregosa et al. "Scikit-learn: Machine learning in Python". In: *Journal of Machine Learning Research* 12 (2012), pp. 2825–2830. arXiv: 1201.0490.

[52] D. Coppersmith and S. Winograd. "Matrix multiplication via arithmetic progressions". In: *Journal of Symbolic Computation* 9.3 (1990), pp. 251–280.

[53] M. Grosse-Wentrup and M. Buss. "Multiclass common spatial patterns and information theoretic feature extraction". In: *IEEE Transactions on Biomedical Engineering* 55.8 (2008), pp. 1991–2000.

[54] P. Gaur et al. "A multi-class EEG-based BCI classification using multivariate empirical mode decomposition based filtering and Riemannian geometry". In: *Expert Systems with Applications* 95 (2018), pp. 201–211.