



universität
wien

DISSERTATION / DOCTORAL THESIS

Titel der Dissertation / Title of the Doctoral Thesis

„Riemannian manifold based EEG decoding algorithm design
and the application on real-time BCI system“

verfasst von / submitted by

Jiachen Xu, M.Sc.

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of
Doktor der technischen Wissenschaften (Dr. techn.)

Wien, 2023 / Vienna, 2023

Studienkennzahl lt. Studienblatt /
degree programme code as it appears on
the student record sheet:

UA 786 880

Dissertationsgebiet lt. Studienblatt /
field of study as it appears on the student
record sheet:

Informatik

Betreut von / Supervisor:

Univ.-Prof. Dipl.-Ing. Dr. -Ing. Moritz Grosse-Wentrup

Acknowledgements

Six years ago, I embarked on a solo journey to Germany, and four years ago, I moved to Vienna with our group when my PhD life set sail.

Throughout the whole journey, there are so many people I need to say thanks. But my first and foremost thanks will definitely be given to my advisor, Moritz. February 23rd of 2018, I can still clearly recall this specific day when I first met Moritz at the LMU seminar room in Großhadern. After our first chat, I was already admired, impressed, and attracted by his kindness, openness, and knowledge. Since then, I followed him from München to Tübingen and finally to Wien (the only city without "ü"), where I started my doctoral life. Frankly speaking, the life of being a PhD student inevitably occasionally comes with frustration when getting a rejection for a paper, anxiety when catching up with a deadline, and embarrassment when making sloppy mistakes. But I felt extremely lucky that happiness, confidence, and optimism have filled up almost all my entire PhD life, which is thanks to all-around support from Moritz. Moreover, I have also learned tremendously from Moritz, not only the professional knowledge like BCIs, signal processing, machine learning, manuscript reviewing, paper writing, and presentation skills, etc., but also the more colorful side of being a human, like coffee, music, cuisine, and so on. Although I have the endless appreciation that I would like to express to Moritz, the only word I can note down here is several sincere thanks. Thanks for providing me with the precious opportunity to join your group as an intern, a master's thesis, and a doctoral student. Thanks for always being so sympathetic and considerate whenever I have troubles. Thanks for your patience and support whenever having difficulties that cannot be solved alone. And also, thanks for being such a fantastic advisor!

I would also like to express my appreciation to Vinay, who closely supervised me when I was still a rookie, and I also would like to extend my gratitude to our dear group colleagues: Anita, Alex, Anja, Philipp, Tryggvi, Peter, Akshey, Gunnar, Sadiq, Christoph, Nike. I have spent a fantastic, harmonious four years with you guys, and thanks a lot for your kind help and encouragement along the way. Your friendship and camaraderie have made this journey both enjoyable and meaningful. I am also grateful to my Chinese friends in Vienna, who have provided me with countless support and enriched my life.

My last but sincere thanks will go to my dad and mom, who provided me with a free and respectful family atmosphere to grow up in; my beloved girlfriend, Yuting, who always stands beside me and accompanies me on countless days, no matter happy or sad, excited or frustrated; my senior aunt, who provided the most important support during the toughest period of my study abroad, and also to all my family members.

Thank you all for your invaluable support. It was a great four years of being a PhD in Moritz's group, a great six years of living in Europe, and a great life that I have spent with each of you.

Abstract

This dissertation endeavors to advance the field of Brain-Computer Interfaces (BCIs) by addressing concerns related to the improvement of electroencephalography (EEG) decoding algorithms and extending the applied domain of current BCIs systems. To achieve this goal, we propose a series of extensions to the state-of-the-art Riemannian manifold based EEG decoding framework. These extensions include a computationally efficient and interpretable framework based on spatial filters extracted from the tangent space, a decoding pipeline with extra nonlinearity derived from using the distance covariance matrix as an input feature, and a novel hardware solution for a closed-loop transcranial electrical stimulation system, accompanied by an original software package. Our algorithmic contributions heighten awareness regarding the interpretation of the Riemannian manifold based decoding framework and provide computationally efficient, model-agnostic solutions. Furthermore, we deepen the understanding regarding the need to incorporate nonlinearity or strictly sufficient invariance properties into the standard Riemannian framework. In addition, our technical work presents a convenient and cost-effective hardware and software solution to facilitate the deployment of the closed-loop transcranial electrical stimulation system by circumventing the usage of proprietary hardware or software, which may further advance related research. Finally, in light of our findings and discussions, we suggest several future research avenues for the community, including neural manifold, manifold based deep learning framework, and other direct extensions based on our contributions in this dissertation.

Kurzfassung

Diese Dissertation zielt darauf ab, das Feld der Brain-Computer-Interfaces (BCIs) voranzutreiben, indem Bedenken hinsichtlich der Verbesserung der Elektroenzephalographie (EEG)-Dekodierungsalgorithmen angegangen und der Anwendungsbereich aktueller BCIs-Systeme erweitert werden. Um dieses Ziel zu erreichen, schlagen wir eine Reihe von Erweiterungen für das State-of-the-Art Riemann-Manifold-basierte EEG-Dekodierungsframework vor. Diese Erweiterungen umfassen ein rechnerisch effizientes und interpretierbares Framework auf der Grundlage von räumlichen Filtern, die aus dem Tangentialraum extrahiert werden, einen Dekodierungs-Pipeline mit zusätzlicher Nichtlinearität, die aus der Verwendung der Abstandskovarianzmatrix als Eingabefeature abgeleitet wird, sowie eine neuartige Hardware-Lösung für ein geschlossenes transkraniales elektrisches Stimulationssystem, begleitet von einer Original-Software-Lösung. Unsere algorithmischen Beiträge erhöhen das Bewusstsein für die Interpretation des Riemann-Manifold-basierten Dekodierungsframeworks und bieten rechnerisch effiziente, modellagnostische Lösungen. Darüber hinaus vertiefen wir das Verständnis dafür, dass Nichtlinearität oder strikt ausreichende Invarianzeigenschaften in das Standard-Riemann-Manifold-Framework integriert werden müssen. Darüber hinaus präsentiert unsere technische Arbeit eine bequeme und kostengünstige Hardware- und Software-Lösung, um die Implementierung des geschlossenen transkraniellen elektrischen Stimulationssystems zu erleichtern, indem die Verwendung proprietärer Hardware oder Software vermieden wird, was die damit verbundene Forschung weiter vorantreiben kann. Schließlich schlagen wir angesichts unserer Ergebnisse und Diskussionen mehrere zukünftige Forschungsrichtungen für das Feld vor.

Publication List

This cumulative doctoral dissertation is a compilation and superordinate analysis of the following works, each of which has been published or is currently under review at peer-reviewed journals and conferences:

- Xu, J., Grosse-Wentrup, M., and Jayaram, V. (2019). Interpretable Riemannian classification in Brain-Computer Interfacing. *Proceedings of the 8th Graz Brain-Computer Interface Conference (GBCIC) 2019: Bridging Science and Application*. (pp: 32-37). Graz University of Technology, Austria
- Xu, J., Grosse-Wentrup, M., and Jayaram, V. (2020). Tangent space spatial filters for interpretable and efficient Riemannian classification. *Journal of Neural Engineering*, 17(2), 026043.
- Xu, J., Markham, A., Meunier, A., Raggam, P., and Grosse-Wentrup, M. (2021). Distance covariance: A nonlinear extension of Riemannian geometry for EEG-based Brain-Computer Interfacing. *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 2000-2005). IEEE.
- Xu, J., and Grosse-Wentrup, M. PyTES: A Python-Toolbox for Closed-Loop Transcranial Electrical Stimulation. *Minor revised at Journal of SoftwareX and preprint version is available at SSRN 4313924*.

Contents

Acknowledgements	i
Abstract	iii
Kurzfassung	v
Publication List	vii
1 Preamble	1
1.1 Brain-Computer Interfaces (BCIs) Background	2
1.1.1 Before 1960s: Recording and Association	2
1.1.2 1960s ~ 1990s: Simple Communication	2
1.1.3 1990s ~ Current	3
1.1.4 Classic Signal Decoding	5
1.1.5 The State-Of-The-Art of Signal Decoding	8
1.2 Riemannian Manifold based EEG Decoding Framework	10
1.2.1 Motivation: EEG Decoding from Optimization’s Perspective	10
1.2.2 Riemannian Manifold	11
1.2.3 Riemannian Geometry for EEG decoding	20
1.3 Decoding Model Interpretability	22
1.3.1 Interpretable Machine Learning Background	22
1.3.2 Interpretability for EEG decoding	23
1.3.3 Non-Invasive Brain Stimulation	24
1.4 Challenges and Questions	25
1.4.1 Research Question 1: Existence of Interpretable and Computationally Efficient Riemannian Methods	25
1.4.2 Research Question 2: Necessity of Introducing Extra Nonlinearity and Preserving Affine-Invariance	25
1.4.3 Technical Question: Finding an Elegant Solution to Achieve Remote Control of Transcranial Electrical Stimulation Signal	26
2 Publications	27
2.1 Interpretable Riemannian Classification in Brain-Computer Interfacing	27
2.1.1 Synopsis	27
2.1.2 Publication	27
2.2 Tangent Space Spatial Filters (TSSF) for Interpretable and Efficient Riemannian Classification	34
2.2.1 Synopsis	34

Contents

2.2.2	Publication	34
2.3	Distance Covariance: A Nonlinear Extension of Riemannian Geometry for EEG-based Brain-Computer Interfacing	54
2.3.1	Synopsis	54
2.3.2	Publication	54
2.4	PyTES: A Python Toolbox for Closed-loop Transcranial Electrical Stimulation	61
2.4.1	Synopsis	61
2.4.2	Publication	61
3	Discussion	67
3.1	Revisit of the Questions	67
3.1.1	Research Question 1: Existence of Interpretable and Computationally Efficient Riemannian Methods	67
3.1.2	Research Question 2: Necessity of Introducing Extra Nonlinearity and Preserving Affine-Invariance	70
3.1.3	Technical Question: Finding an Elegant Solution to Achieve Remote Control of Transcranial Electrical Stimulation Signal	71
3.2	Future Work	71
3.2.1	Neural Manifold - The Neuroscientific Evidence of the Subspace Theory of Neuronal Activity	71
3.2.2	Manifold based Deep Learning Framework	72
3.2.3	Other Extensions based on the Current Works	74
3.3	Conclusion	75
	Bibliography	77

1 Preamble

Communication and interaction are essential for daily life and significantly impact living quality, but physical impairments can prevent some people from fulfilling these essential needs. Brain-Computer Interfaces (BCIs) have been developed since the 1970s to mitigate the influence of such impairments. One of its primary missions is to bridge brain activity and external devices without physical execution directly. Examples of such BCIs include mind-controlled robotic arms and machines that answer questions by decoding brain signals, etc. Although BCIs depict a fantastic future, the reality we are facing is often inaccurate commands translated from brain signals or limited application domains of such systems.

Nevertheless, the abundance of tools provided by advances in hardware and software over the past few decades has allowed us to address the challenges encountered by the BCI system. This dissertation presents our contributions to developing the BCI system from two main aspects: improving the performance of decoding algorithms through an advanced decoding framework based on the Riemannian manifold and exploring the combination of the passive brain signal decoding system with the active brain stimulation system. The BCI systems mentioned in this dissertation specifically refer to those based on electroencephalography (EEG) signals, which are non-invasive measurements of the electrical brain activity over the scalp.

To facilitate understanding, we first present a brief overview of the development of BCIs technology from Section 1.1.1 to 1.1.3. Subsequently, we introduce the classic (Section 1.1.4) and state-of-the-art (SOTA) EEG decoding algorithms (Section 1.1.5). We then introduce the Riemannian manifold (Section 1.2) and its applications in EEG decoding (Section 1.2.3). Subsequently, we present the decoding model interpretability and its meaning in the EEG decoding scenario, which is followed by introducing one technique that can actively intervene the brain activity (Section 1.3). At the end of Chapter 1, we raise several research and technical questions by addressing the challenges faced by current EEG-BCIs systems in Section 1.4.

In Chapter 2, we present several publications to answer these questions. We then revisit these questions based on the presented papers in Section 3.1 and provide further discussion of these questions, as well as avenues for future research, in Section 3.2. Finally, we conclude this dissertation by summarizing our key contributions to developing modern EEG-BCIs systems in Section 3.3.

1.1 Brain-Computer Interfaces (BCIs) Background

This section briefly overviews the historical development of EEG-BCIs techniques over the past centuries. It is important to emphasize that although the field of BCIs and EEG research has a long history, the aim of this section is not to provide a comprehensive review of the past. Instead, we focus on exploring how the past can inspire current and future related research.

1.1.1 Before 1960s: Recording and Association

Electricity has been considered as the "mysterious invisible agent that operates our nerves and muscles" since the first half of the 19th century [Brazier, 1984], but evidence of its presence in the brain was not reported until Richard Caton's findings in 1875 [Caton, 1875]. The first photograph of the electrical current found in a dog's brain was published by Vladimir Pravdich-Neminsky in 1912. Later, in 1924, Hans Berger measured the electrical brain activity in the epidural recording of the human brain, which was later termed as electroencephalography (EEG) [Berger, 1929]. Berger's work is then followed by Gibbs *et al.*, who further discovered the association between epileptic seizures and the electrical brain activity measured by a three-channel EEG system [Gibbs et al., 1935]. Subsequently, Delgado pioneered the research of implanting electrodes into humans and animals in 1952 [Delgado, 1952]. By reviewing these "ancient" works as summarized in [Brazier, 1961], it can be noticed that before the 60s, the research about brain activity was primarily inspired by clinical usage. More specifically, researchers aimed to decipher the association between certain diseases and brain signal patterns.

1.1.2 1960s ~ 1990s: Simple Communication

Stepping into the 60s, the advent of computers profoundly propelled the development of brain-computer interfaces (BCIs). In 1958, Clark *et al.* pioneered a digital computing system and invented the average response computer [Clark et al., 1961], which enabled the extraction and visualization of average evoked responses. Subsequently, Estrin introduced the concept of an "online electroencephalographic digital computing system" [Estrin, 1965] and suggested "*by time-sharing a large computer, almost limitless repertoire of mathematical analysis can be afforded*" [Estrin, 1965]. She also managed to update the average evoked response in a real-time fashion, establishing the prototype of a BCI system. Coincidentally, in the same year, the composer Alvin Lucier developed the first application of BCIs - creating a music piece only with EEG signals. Later, in 1969, Fetz published a paper presenting the first online BCIs system, in which he demonstrated that a monkey could deflect a needle on an electrical meter using cortical control from a single neuron [Fetz, 1969]. During the 1970s, a seminal paper on BCIs entitled "Toward direct brain-computer communication" was published by Vidal [Vidal, 1973]. In this paper, Vidal presented a formal definition of Brain-Computer Interfaces as "*utilizing the brain signals in a man-computer dialogue*" and provided a comprehensive overview of all the essential components of a BCI system. Shortly thereafter, Vidal demonstrated that BCIs

1.1 Brain-Computer Interfaces (BCIs) Background

could be used for communication, as evidenced by [Vidal, 1977].

In the meantime, the research and application of event-related potentials (ERP) were gaining increasing attention. For example, P300 was first reported in the mid-60s [Chapman and Bragdon, 1964]. Later, Farwell *et al.* invented the first application - P300 speller in the late 80s [Farwell and Donchin, 1988]. Moreover, the actively regulated evoked response was first defined as sensorimotor rhythm (SMR) or μ -rhythm [Roth et al., 1967]. And the movement-related SMR, which is also known as the event (de-) synchronization (ERD/ERS), was reported by Pfurtscheller in 1977 [Pfurtscheller, 1977]. This work is soon adopted and utilized as the input signal for BCIs by Wolpaw *et al.* in 1991 [Wolpaw et al., 1991]. (A systematic summary of these evoked potentials can be found in [Kübler, 2020])

The developments over the past three decades have highlighted the significant impact of computers on the real-time analysis of electrical brain signals. This technological advancement has made it possible to filter, average, classify various evoked potentials, and utilize signal rhythm to enable individuals to use BCI systems for simple applications such as cursor movement [Pfurtscheller, 1977], speller [Farwell and Donchin, 1988], and robot control [Bozinovski et al., 1988]. Consequently, the researchers made the best use of advanced hardware systems back then, significantly contributing to BCIs technologies' development.

1.1.3 1990s ~ Current

Since the 1990s, the BCIs community has experienced remarkable growth due to the collaborative efforts of interdisciplinary researchers, as evidenced by the exponential

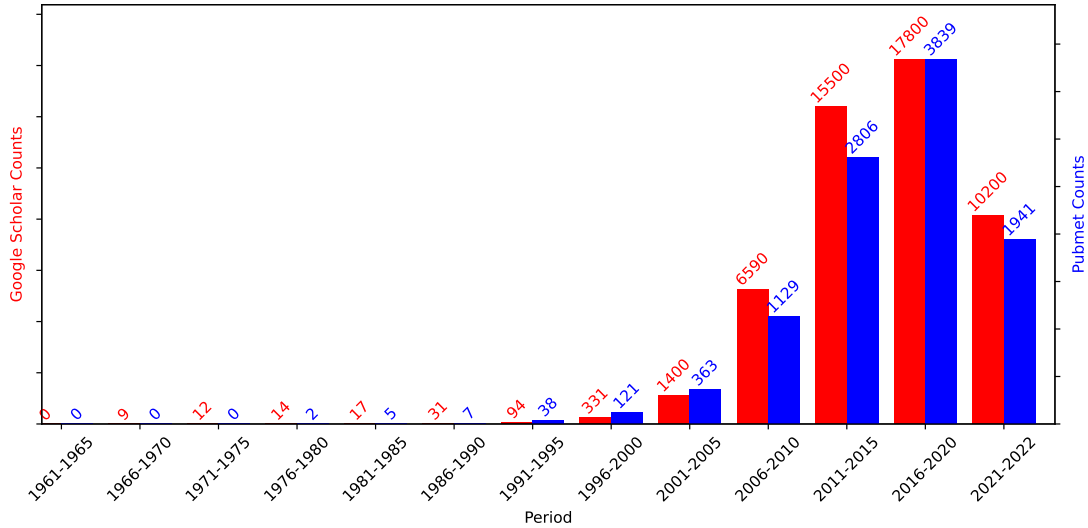


Figure 1.1: The number of matched publications with keywords "brain", "computer", "interface" and "BCI" on Google scholar (red bar) and PubMed (blue bar)

1 Preamble

increase in related publications illustrated in Figure 1.1. Given the substantial number of papers published in the past three decades, it is not feasible to enumerate every new research direction. Therefore, we introduce the recent development of BCI systems from three aspects, as illustrated in Figure 1.2: the recording technique of neural activity, the decoding algorithm of the recorded signal, and the experimental paradigm in which either device is controlled, or feedback is applied.

Recording Modalities

The technique of electrical recording of neural activity has undergone significant advancements thanks to the advancement in microelectronics. From single-channel intracortical measurements, the technique has evolved into multi-channel signal acquisition, which can be further classified into EEG, Stereotactic-EEG (sEEG), electrocorticography (ECoG), and intracortical recording based on the electrode placement depth. Subsequently, motivated by the electromagnetic coupling theory, researchers have attempted to measure brain activity through magnetic signals, i.e., magnetoencephalography (MEG), avoiding invasive electrode implantation and the signal attenuation from non-invasive electrodes caused by skull and brain tissue. This approach preserves both temporal and spatial resolution. Additionally, with a deeper understanding of the theory of neurovascular coupling, researchers have developed BCIs systems that detect the blood-oxygen-level-dependent imaging (BOLD) response or hemodynamic response. Devices such as functional near-infrared spectroscopy (fNIRS) and functional magnetic resonance imaging (fMRI) are used to measure the BOLD signal.

In summary, advanced recording techniques have improved the quality of the collected signal, which is a critical requirement for constructing a BCIs system with optimal performance. However, acquiring high-quality signals alone does not fulfill all the needs of a BCIs system. A proper processing and decoding pipeline is also necessary to translate the acquired signal into executable devices or feedback commands.

Signal Preprocessing

The measured brain signal is often considered non-stationary due to the heterogeneity and dynamics of brain activity. Some recording methods are susceptible to environmental noise and non-task-related muscular or brain activity, necessitating the preprocessing of the brain signal. Although algorithms may vary, most preprocessing methods can be categorized as time and/or frequency domain processing. For instance, epoching in the time domain can restrict the variability of the detected signal, allowing it to be assumed as stationary in the observed time window, which is often task-related. Furthermore, bandpass filtering or re-referencing techniques are also commonly used, as most neurophysiological studies have shown that brain activity lies within specific frequency bands. Consequently, we can eliminate the signal in non-related frequency bands, which are considered as noise.

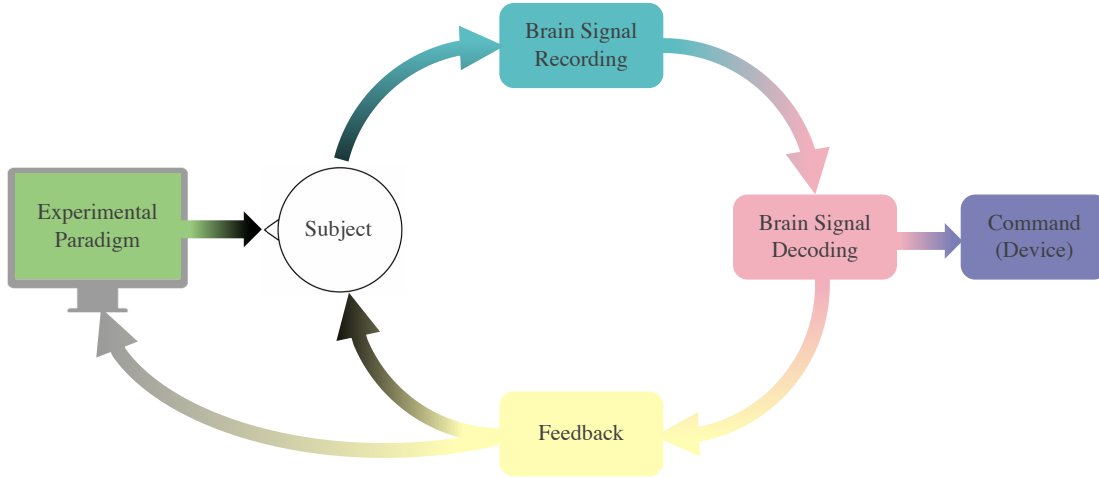


Figure 1.2: The basic components of a BCIs system.

1.1.4 Classic Signal Decoding

The detected brain activity in a BCIs system is usually elicited by passive stimuli or active intentions. In addition to investigating how different events affect brain activity, an equally crucial research question is to determine the specific task that generates a particular brain signal pattern or a systematic change of brain patterns. This process is commonly known as brain signal decoding.

The Classic Decoding Pipeline

As a pattern recognition system, the EEG signal decoding process typically involves feature extraction and classification. Moreover, as a data-centric methodology, an effective decoding pipeline should also account for the intrinsic properties of the data, which can be described from two perspectives: the datasets and the signals.

As introduced in Section 1.1.3, the EEG signal is known for being an endogenous non-stationary process susceptible to various exogenous interferences. This results in a significant variation in the quality of the recorded data at different times, from different subjects, and in different laboratories. Moreover, the size of EEG datasets is often limited due to technical obstacles such as the lengthy preparation and calibration of recording setups. Consequently, simple classifiers with low capacity have traditionally been preferred because such characteristics can minimize the risk of overfitting and increase robustness against data variation. Another benefit of simple classifiers is that faster computational speed can ensure real-time decoding and feedback, which is essential due to hardware limitations. On the other hand, to ensure sufficient discriminability of the decoding pipeline with a simple classifier, considerable effort has been dedicated to exploring feature extraction methods to alleviate the classifier's workload. Ideally, the feature extraction methods can incorporate prior neurophysiological knowledge or signal

1 Preamble

properties, such as non-stationarity, robustness to outliers, etc. Comprehensive guidance on feature design can be found in [Lotte et al., 2007].

In addition to concerns over classifier capacity, small dataset sizes - the limited number of trials per class - and high-dimensional data often present a significant challenge known as the curse of dimensionality [Jain et al., 2000]. This refers to the low ratio of the number of trials to feature dimensionality, which results in suboptimal performance and frequently affects high-density EEG signals. One straightforward solution to mitigate this issue is spatial filtering, which can reduce the dimensionality of the input features.

Spatial Filtering

Spatial filtering was initially designed to separate interferences from contaminated signals when both the desired signal and the interference occupy the same temporal frequency band [Van Veen and Buckley, 1988]. Therefore, spatial filtering processes data collected over a spatial aperture, much like temporal filtering processes signals collected over a temporal aperture [Van Veen and Buckley, 1988]. Through the intricate design of the aperture, the signal of interest can ideally be amplified while irrelevant interferences or artifacts are suppressed, regardless of their similarity in temporal frequency features.

The trait of spatial filtering coincidentally addresses a major concern in EEG signal analysis, as artifacts and interferences often share the same temporal frequency band as the task-induced EEG signal. Consequently, this concept was quickly embraced by the EEG-BCI community since the 1980s. For instance, the common average reference (CAR) algorithm aims to eliminate the common noise recorded by all electrodes. Also, the surficial Laplacian spatial filter (sLSF) sharpens the local topological features to avoid biased connectivity analysis caused by high correlation among adjacent electrodes. It accomplishes this by suppressing the influence of spatially-broad features like volume conducted from distal sources and distributed but highly coherent sources [Cohen, 2014].

Subsequently, more sophisticated spatial aperture structures have been proposed to enhance spatial selectivity. Rather than relying on data-irrelevant designs, e.g., CAR is a weightless average, and sLSF is based on the predefined math formula, optimization-based spatial filter design has become more prevalent. For example, principal component analysis (PCA) extracts a set of spatial filters, which maximize the variance of the data explained. Independent component analysis (ICA) is based on the assumption that active neuronal sources are mutually independent. Therefore, the spatial filters are usually extracted by solving two optimization problems: minimizing mutual information concerning the independence of source activity and maximizing the non-Gaussianity of each source because the probability density of the mixed signal is closer to the Gaussian distribution than any single source signal according to the central limit theorem. Additionally, for event-driven data with labels, i.e., supervised optimization, generalized eigenvalue decomposition can solve the optimization problem of maximizing the distance between signal power elicited by different events. One representative of this type of method is known as the common spatial pattern (CSP), which is one of the most successful spatial filtering methods in the EEG-BCIs field [Koles, 1991, Koles and Soong, 1998] (A concise introduction of CSP history can be found in Section 4 [Congedo et al., 2017]). Numerous

1.1 Brain-Computer Interfaces (BCIs) Background

spatial filter algorithms have since been proposed based on various optimization problems, including XDawn [Rivet et al., 2009], Source Power Comodulation (SPoC) [Dähne et al., 2014], canonical correlation analysis (CCA) [De Clercq et al., 2006], and so on.

Aside from spatial selectivity, the broad usage of spatial filtering in EEG analysis is motivated by several additional advantages, as enumerated below.

1. *Improved signal-to-noise ratio (SNR)*: Spatial filtering, particularly the linear type, can be viewed as a weighted average of signals obtained from all electrodes. By selectively amplifying or suppressing specific signal components and then aggregating all components (sum in linear spatial filtering), spatial filtering can significantly improve the signal-to-noise ratio (SNR), akin to the common mode filter.
2. *Reduced overfitting risk and dimension redundancy*: Taking a linear algebra perspective, spatial filtering can also be interpreted as domain transformations. For instance, the highly correlated high-density EEG signal can be de-correlated by applying a linear spatial filter, such as ICA, because the signal sources are mutually independent by definition. As a result, meaningful signals are projected and condensed to subspace, while the rest of the dimensions are regarded as noise space that can be discarded. This drastically reduces the dimensionality of the input EEG data, reducing the risk of overfitting with high-dimensional data.
3. *Accelerated computational speed*: Another advantage of dimensionality reduction through spatial filtering is simplifying the computational complexity. For example, if a K -dimensional spatial filter is applied to the M -dimensional EEG data, where $K \ll M$, the computational complexity for the matrix computation can be reduced from $O(M^3)$ to $O(M^2)$. This can have a significant impact when M is large, such as in the case of high-density EEG signals.
4. *Better interpretability*: Spatial filtering can also be viewed as a conventional forward problem, i.e., inferring the source activity from the measurement. The inverse or pseudo-inverse of the spatial filtering matrix is known as the spatial pattern and represents the solution to the inverse problem [Haufe et al., 2014]. By visualizing the column vectors in the spatial pattern matrix, we can identify each channel's contribution to each measurement signal and reject non-task-related brain sources based on prior neurophysiological knowledge, such as ocular movement artifacts. Additionally, topological patterns can identify large-scale brain networks to further analyze brain connectivity. If a volume conduction model is hypothesized, such as the six-layer head model in the beamformer, spatial filtering can also address the source localization problem [Grosse-Wentrup et al., 2009].
5. *Enhanced capacity of decoding pipeline*: Last but not least, spatial filtering can increase the complexity of the decoding pipeline and reduce the need for high-capacity feature and classifier designs.

1.1.5 The State-Of-The-Art of Signal Decoding

In the realm of EEG decoding pipelines, the conventional approach has been to employ dedicated spatial filter and feature designs in conjunction with a simple classifier, such as a linear or naive Bayesian classifier, as evidenced by the past BCI competitions' winning decoding pipelines [Blankertz et al., 2004, Blankertz et al., 2006b, Tangermann et al., 2012]. Recently, two emerging frameworks have gained increasing attention and may eventually replace the partial dominance of the classic decoding pipeline: deep learning (DL) and Riemannian manifold based classification frameworks. Despite their differences, these two frameworks share a significant commonality: they are both nonlinear classification frameworks, aligning with the natural properties of brain signals or EEG data. Therefore, before delving into the specifics of these frameworks, it is essential to review the evidence supporting the existence of nonlinearity in neural data.

Brain - A Nonlinear System

A linear system on a large scale may be ideal for theoretical analysis, but it is often impractical for implementation due to the stringent requirement of linearity for all sub-components. Even a single violation of linearity at one component can render the system nonlinear. With billions of neurons, the human brain is one of the most complex large-scale systems and is no exception to the previous rule. In general, the nonlinearity of brain activity can be briefly described from micro to macro perspectives in two main aspects:

1. *Neuronal signal initialization and propagation*: Several biological neuron models have been proposed to elucidate the neurophysiological mechanisms underlying neuronal action potentials. One shared feature among many of these models is the use of nonlinearity to describe neuronal dynamics [Izhikevich, 2007, Izhikevich and Edelman, 2008]. The prestigious Hodgkin-Huxley model [Hodgkin and Huxley, 1952], for instance, employs a set of nonlinear equations to characterize the electrical properties of neurons during the generation and propagation of action potential.
2. *Anisotropic attenuation of brain tissues and volume conduction model*: The conductivity of brain tissue is not uniform in all directions due to the anisotropic nature of axons and other structural components [Jones, 2008, Basser and Pierpaoli, 2011]. This leads to different levels of attenuation for electrical signals that travel parallel and perpendicular to the orientation of these structures. Consequently, accurate modeling of volume conduction in the brain requires consideration of anisotropic conductivity. The volume conduction model is hence widely used to simulate the propagation of electrical signals in the brain [Nunez and Srinivasan, 2006], and it relies on the accurate representation of the electrical properties of brain tissue. Incorporating anisotropic attenuation into the volume conduction model can significantly improve its accuracy and enhance our understanding of brain function. The downside, however, is the introduction of nonlinearity.

Regarding the two main sources of nonlinearity in the brain concluded above, the latter matters more for EEG-BCIs because the ultimate goal is to decode source activity from

the measured signal instead of the single-neuron activity. Thus, the ideal solution would be the inverse function of the nonlinear volume conduction model. This suggests that the optimal decoding model should also be of a nonlinear nature. Recent advances in nonlinear decoding pipelines have brought new hope in approaching this ultimate goal, and one of the most well-known frameworks is the deep learning decoding framework.

Deep Learning based Decoding Framework

The McCulloch-Pitts neuron model from the 1940s marked the start of deep learning [McCulloch and Pitts, 1943]. The Perceptron model [Rosenblatt, 1958], introduced later as the first trainable neuron model with updatable parameters, represented the initial phase of deep learning until it was revealed to have limitations [Minsky and Papert, 1969], such as the inability to learn XOR function, in the 1960s. Decades later, rekindled interest in deep learning emerged as researchers found violations of previously held statements in [Minsky and Papert, 1969], leading to the argument that complex behavior can be achieved with many simple computational units. Modern deep learning concepts, including the back-propagation model (BP) [Rumelhart et al., 1986], convolutional neural networks (CNN) [LeCun et al., 1995], and long-short term memory (LSTM) [Hochreiter and Schmidhuber, 1997], were developed during this wave. The universal approximation theorem later proved that deep learning can approximate any nonlinear mappings between inputs and outputs [Cybenko, 1989, Hornik et al., 1989, Hornik, 1991]. However, hardware limitations at the time hindered the further advancement of deep learning until the late 2000s. With the advancement of big data and parallel computing hardware, deep learning has flourished and is now capable of practically approximating any mappings using a deep and hierarchical network with sufficient data. This has led to revolutionary changes, as intricate feature design with prior domain knowledge is no longer necessary. Instead, the network can automatically learn features and classifiers from minimally processed or even raw input data, even though the learned features may not be interpretable.

The property of deep learning to learn features automatically is particularly advantageous for EEG signal decoding because the traditional feature design for EEG decoding requires extensive effort to craft by hand. Therefore, deep learning has been widely embraced by the EEG-BCI community since the 2010s, leading to a surge in related research [Roy et al., 2019]. This feature learning capability clearly distinguishes between early neural network usage and modern deep learning applications. In the past, neural networks were regarded as pure classifiers often paired with various feature extraction techniques. For example, the multilayer perception model is combined with autoregressive features [Anderson and Sijercic, 1996] or band power difference [Palaniappan, 2005] or finite impulse filter weights [Haselsteiner and Pfurtscheller, 2000]. Also, the local neural classifier is with power spectrum coefficients [Millan et al., 2002], and probabilistic neural network (radial basis function neural networks) with linear predictive coding spectral analysis after ICA and PCA [Hoya et al., 2003]. In contrast, modern deep learning frameworks focus more on the network structure, such as the convolution neural network [Schirrmester et al., 2017, Manor and Geva, 2015, Kwak et al., 2017] and deep belief network [Sturm et al., 2016, Yin et al., 2017], while minimizing the need for EEG fea-

1 Preamble

ture extraction with only using, e.g., raw EEG signal [Schirrmester et al., 2017], EEG time points (epoched EEG data) [Manor and Geva, 2015], or band power [Kwak et al., 2017, Sturm et al., 2016, Yin et al., 2017]. Interestingly, 74 out of 154 deep-learning-based EEG decoding papers published until 2019 used raw EEG signals as input according to [Roy et al., 2019], which indicates a trend toward shifting focus from feature extraction.

The deep learning framework has achieved significant success in offline EEG decoding thanks to the nonlinearity. However, this framework has also faced criticism, particularly regarding the unique characteristics of EEG data, especially the limited data set size as mentioned in Section 1.1.4. Moreover, the feature learning property of deep learning is a double-edged sword. On the one hand, it can significantly reduce the effort of feature extraction design by focusing on the network structure. On the other hand, non-stationary EEG signals often suffer from low SNR issues, unlike the case for images or speech signals. Therefore, using minimally processed or even raw data as input while completely ignoring the underlying neurophysiological knowledge of the data is also risky. Additionally, the interpretability of features is essential, particularly in the EEG-rehabilitation field, where it is necessary to exclude the artifacts- or interference-driven classification. Unfortunately, interpretability is one of the most well-known concerns about deep learning methods.

One may wonder if there is a framework that combines the success of nonlinear decoding and the characteristics of EEG data structure while being easily trainable with small datasets. The Riemannian manifold based classification framework fulfills all these requirements.

1.2 Riemannian Manifold based EEG Decoding Framework

1.2.1 Motivation: EEG Decoding from Optimization’s Perspective

As a subset of the pattern recognition problem, the EEG decoding model can be formulated as an optimization problem. Whether solved via conventional linear pipelines or modern nonlinear deep learning frameworks, the goal is to minimize a loss function $\mathcal{L}(\cdot)$ while satisfying constraints $g(\cdot)$.

$$\min_f \mathcal{L}(f(x), y), \text{ s.t., } x \in \mathbb{R}^n \text{ and/or } g(x), \quad (1.1)$$

where x and y stand for the data and labels, respectively. In most cases, the data is assumed to lie in Euclidean space $\mathbb{R}^n$ without any underlying geometric structure. Therefore, regular optimization algorithms such as gradient descent or multiplier methods are employed to solve the problem under constraints $g(\cdot)$. While these methods are effective for image and speech data, the validity of the previous assumption must be carefully considered in EEG decoding due to the unique characteristics of EEG signals.

As mentioned earlier, spatial filtering is widely applied to EEG signals, and the resulting spatial patterns can be used to analyze the topological distribution of brain sources. Combined with the volume conduction model introduced before, these suggest that there may underlie some geometrical structure inherent in EEG signals. For example, EEG signals are collected over the curved surface of the scalp, capturing radiated brain sources

1.2 Riemannian Manifold based EEG Decoding Framework

after anisotropic attenuation. If we consider the data to follow a particular geometrical structure, i.e., $x \in \mathcal{M}$ where $\mathcal{M}$ is a subspace of $\mathbb{R}^n$, then the optimization problem presented in Eq. (1.1) may not be sufficient as $g(x)$ for following reasons:

- No convergence to a solution when the constraints are non-convex.
- No guarantee that the iterates remain on the manifold $\mathcal{M}$.
- No utilization of the geometrical structure of $\mathcal{M}$.

Therefore, we can consider the following alternative optimization problems to avoid the drawbacks of Eq. (1.1), which is:

$$\min_f \mathcal{L}(f(x), y), \text{ s.t., } x \in \mathcal{M} \text{ and/or } g(x), \quad (1.2)$$

where $\mathcal{M}$ is a non-Euclidean subspace with curvature embedded in the Euclidean space $\mathbb{R}^n$.

1.2.2 Riemannian Manifold

When choosing a topological space for classifying input features, it is often prioritized to consider one that resembles local Euclidean space, allowing for easy application of most available classification algorithms without adaptation. Additionally, the space should have well-defined physical quantities such as length, angle, and orientation, which is crucial for interpreting the physical meaning of the features or model. One candidate space that fulfilled these considerations is the Riemannian manifold which is defined as follows:

Definition 1. Riemannian Manifold: A Riemannian manifold $(\mathcal{M}, \rho)$ is a smooth manifold $\mathcal{M}$ equipped with a Riemannian metric ρ defined as the inner product on the tangent space $\mathcal{T}_x\mathcal{M}$ for each point $x \in \mathcal{M}$, $\rho_x(\cdot, \cdot): \mathcal{T}_x\mathcal{M} \times \mathcal{T}_x\mathcal{M} \rightarrow \mathbb{R}$. [Boothby, 2003]

The above definition of Riemannian manifold contains three key components: a point x on the manifold, a Riemannian metric denoted by ρ , and a smooth manifold $\mathcal{M}$. These concepts are further explained as follows:

Common Riemannian Manifolds and the Associated Manifold Points

The Riemannian manifold is a powerful mathematical tool, but its broad definition may not be suitable for analyzing the geometric structure of certain data. To address this issue, Riemannian manifolds can be classified based on the type of points that comprise the manifold, providing a more targeted approach for related studies. Given that EEG decoding data typically comprises two-dimensional signal matrices or their variations, a useful Riemannian manifold is one constructed from $\mathbb{F}^{n \times p}, n \geq p$ matrices, which is known as the *matrix manifold* [Absil et al., 2009] and $\mathbb{F}$ can be either the real set $\mathbb{R}$ or the complex set $\mathbb{C}$. Below, we will introduce several important matrix manifolds that are relevant for EEG decoding.

Stiefel manifold: the linearly independent or orthonormal matrix manifold

The Stiefel manifold is a specialized form of the matrix manifold that characterizes matrices with independence or orthonormality. The non-compact Stiefel manifold represents the matrices with independence, while the compact Stiefel manifold is for the orthonormal matrices. The definitions for these manifolds are provided below (adapted from [Absil et al., 2009]):

Definition 2. Non-compact Stiefel Manifold: Let $\mathbb{F}_*^{n \times p} (n \geq p)$ denote the set of all $n \times p$ matrices whose columns are linearly independent. This set is an open subset of $\mathbb{F}^{n \times p}$ since its complement $\{X \in \mathbb{F}^{n \times p} : \det(X^*X) = 0\}$ is closed. Consequently, it admits a structure of an open submanifold of $\mathbb{F}^{n \times p}$. This manifold will be referred to as the *manifold* $\mathbb{F}_*^{n \times p}$, or the non-compact Stiefel manifold of full-rank $n \times p$ matrices.

Definition 3. Compact (orthogonal) Stiefel Manifold: Let $\text{St}(p, n) (n \geq p)$ denote the set of all $n \times p$ orthonormal matrices; i.e., $\text{St}(p, n) := \{X \in \mathbb{F}^{n \times p} : X^*X = I_p\}$, where I_p denotes the $p \times p$ identity matrix. The set $\text{St}(p, n)$ is called an orthogonal or compact Stiefel manifold.

The Stiefel manifold finds broad applications in various fields, such as robotics with kinematic modeling [Hüper et al., 2007, Hüper et al., 2008], quantum chemistry [Chiumiento and Melgaard, 2012], nonlinear dimension reduction [Kanamori and Takeda, 2012], and network synchronization [Markdahl et al., 2020]. In machine learning, data with column orthogonal structure is also gaining attention in applications like computer vision [Lui, 2012] and medical image analysis [Chakraborty et al., 2017a]. Recently, the optimization on the Stiefel manifold has been explored in the deep learning framework, replacing the gradient optimization on the Euclidean space [Arjovsky et al., 2016, Huang et al., 2018a, Bansal et al., 2018, Li et al., 2020], as orthonormality constraints on the parameter space appear to be advantageous.

In EEG decoding, the Stiefel manifold has gained prominence for various purposes. Yamamoto *et al.* employed dimension reduction via the Stiefel manifold to analyze the EEG subspace [Yamamoto et al., 2021]. Furthermore, using back-propagation and gradient-based methods on the Stiefel manifold in deep learning frameworks has demonstrated its potential in decoding EEG signals [Huang and Van Gool, 2017, Pan et al., 2022].

It is necessary to note that in deep learning frameworks, the Stiefel manifold serves primarily as the parameter space to ensure orthogonality rather than the input data or feature space. This is due to the difficulty of enforcing orthogonality on input data, particularly in the case of EEG signals, which can be highly correlated. Consequently, determining the appropriate manifold on which to model input data is also a crucial issue. This leads to the introduction of the second type of Riemannian manifold - the symmetric positive definite (SPD) manifold.

Symmetric positive definite (SPD) manifold

The SPD manifold is defined as the collection of symmetric positive definite matrices, represented below:

Definition 4. Symmetric positive definition (SPD) manifold: Let $\text{Sym}_+(n)$ denote the set of all $n \times n$ symmetric positive definite matrices; i.e., $\text{Sym}_+(n) := \{X \in \mathbb{R}^{n \times n} : X^T = X \text{ and } \lambda_{\min}(X) \geq 0\}$, where I_n denotes the $n \times n$ identity matrix and $\lambda_{\min}(X)$ is the smallest eigenvalue of matrix X . The set $\text{Sym}_+(n)$ is called symmetric positive definition (SPD) manifold.

Positive definite matrices possess several crucial mathematical properties, as enumerated in [Bhatia, 2009]. Our focus, however, is on a specific type of SPD matrix, the covariance matrix. The covariance matrix extends the notion of variance from a mono-dimension to a multi-dimension signal. Instead of being positive-definite, it is often positive-semi-definite because of the potentially high correlation among channels, which can result in rank deficiency and eigenvalues of zero. Real-world data may even produce covariance matrices that are negatively definite due to numerical errors, inadequate data size, non-Gaussian distributed data, etc. Therefore, regularization or dimension reduction techniques, such as Ledoit-Wolf shrinkage [Ledoit and Wolf, 2004], Oracle Approximating Shrinkage [Chen et al., 2010], and principal component analysis [Abdi and Williams, 2010], are commonly employed to ensure that the estimated covariance matrices lie on the SPD manifold. Furthermore, the covariance matrix assumes significance owing to its encapsulation of valuable information in multiple domains, as listed below:

- In computer vision, covariance matrices represent the interdependence between pixels in an image and capture crucial information about the intensity distribution.
- In speech processing, covariance matrices model the correlation between acoustic features in speech signals, and are integral components of numerous speech recognition systems.
- In natural language processing, covariance matrices model the association between words in a corpus, and are deployed to represent the co-occurrence matrix in document classification and topic modeling algorithms.
- In machine learning, covariance matrices are often used to measure variables' similarity.
- In EEG decoding, covariance matrices comprise the within-channel power (variance) along the diagonal, and off-diagonal elements, which reflect the interrelation between signals from different channels.

Due to its physical significance, the covariance matrix often serves as the input feature in various frameworks, sometimes referred to as tensor computing framework [Pennec et al., 2006]. As the covariance matrix is inherently symmetric positive definite after the regularized estimation, the SPD manifold refers to the manifold constructed from covariance matrices without further elaboration.

Other manifolds

1 Preamble

Apart from the Stiefel and SPD manifold introduced above, there exist other types of Riemannian manifolds, including the flag manifold, fixed-rank manifold, and Grassmannian manifold, which can be viewed as the quotient manifold of the Stiefel manifold [Absil et al., 2009]. The Grassmannian manifold is also employed in various domains, such as communication [Zheng and Tse, 2002], nonlinear dimension reduction [Chevallier et al., 2014], time series classification [Al-Samhi et al., 2021, Yu et al., 2020], and deep learning [Huang et al., 2018b]. As these manifolds are not the primary focus of this dissertation, they will not be further discussed here.

Building Blocks of Riemannian Manifold

Most machine learning problems, including most EEG decoding tasks, can be approached as a puzzle game of identifying similarities within data. Data sharing the same label are presumed to be highly alike, while those with dissimilar labels exhibit marked differences. The widely recognized similarity measure is the *distance*, which indicates the length of the line connecting two data points. Therefore, we will commence by reviewing the definition of a line in Euclidean space, followed by an elaboration on the "line" in the manifold.

Recap of distance and inner product in Euclidean space

In Euclidean space, the conventional approach is to select the shortest straight line connecting two data points to compute their Euclidean distance, denoted as d_{Euclid} . For instance, the d_{Euclid} between two one-dimensional points, a and b , is calculated as $d_{Euclid} = \sqrt{(a - b)^2}$. The L2 norm in vector space and the Frobenius norm $\|\cdot\|_F$ in matrix space represents the generalization of the Euclidean distance, as illustrated below.

$$d_{Euclid} = \|\mathbf{A} - \mathbf{B}\|_F = \sqrt{\sum_{i,j=0}^{N-1} (\mathbf{A}_{i,j} - \mathbf{B}_{i,j})^2}, \mathbf{A} \& \mathbf{B} \in \mathbb{R}^{N \times N}, \quad (1.3)$$

where $\mathbf{A}$ and $\mathbf{B}$ denote the two matrix points. Importantly, the Euclidean distance can be represented as the form of an inner product as shown below:

$$d_{Euclid} = \|\mathbf{A} - \mathbf{B}\|_F = \text{tr}((\mathbf{A} - \mathbf{B})^T(\mathbf{A} - \mathbf{B})) = \langle \text{vec}(\mathbf{A} - \mathbf{B}), \text{vec}(\mathbf{A} - \mathbf{B}) \rangle, \quad (1.4)$$

where $\text{vec}(\mathbf{A})$ denote the vectorization of matrix $\mathbf{A}$.

Geodesic, Riemannian distance and mean point on the manifold

However, on the manifold, the definition of distance necessitates an additional structure, namely, the metric tensor ρ , which can be regarded as a generalization of the inner product. The Riemannian metric ρ is a positive-definite metric, and the length of any smooth curve connecting two points on the manifold can be established by integral. The curve with the shortest length is referred to as the *geodesic*, as illustrated in Figure 1.3.

The two endpoints of the geodesic $\gamma(t)$ correspond to the two manifold points (covariance matrices) $\gamma(0) = \mathbf{C}_1$ and $\gamma(1) = \mathbf{C}_2$, and the midpoint of this geodesic $\gamma(\frac{1}{2}) = \mathbf{C}_m$ is known as the mean point. The Riemannian distance between $\mathbf{C}_1$ and $\mathbf{C}_2$, i.e., the length of the geodesic, is defined as follows:

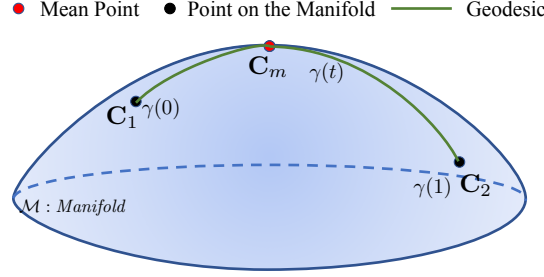


Figure 1.3: Geodesic $\gamma(t)$: The shortest smooth curve that connects two covariance matrices (black point) $\mathbf{C}_1$ and $\mathbf{C}_2$ on the Riemannian SPD manifold. The red point $\mathbf{C}_m$ indicates the midpoint of the geodesic, i.e., $\gamma(\frac{1}{2})$, which is the special case of mean point.

$$\mathcal{L}_\gamma(\mathbf{C}_1, \mathbf{C}_2) = \sqrt{\int_0^1 \left\| \frac{d\gamma(t)}{dt} \right\|^2 dt} \quad (1.5)$$

If there are more than two manifold points, the midpoint will be generalized as the mean point. One of the most well-known mean points on the manifold is Fréchet mean point [Moakher, 2005] (also known as Karcher mean), as shown in Figure 1.4 and defined in Eq. (1.6). Notably, Fréchet mean has been shown as the manifold version of scalar geometric mean [Bhatia, 2009], which hence inherits many advantages, e.g., robustness against outliers [Congedo et al., 2017], etc. More discussion about the mean point choice can refer to [Congedo et al., 2017, Congedo et al., 2019].

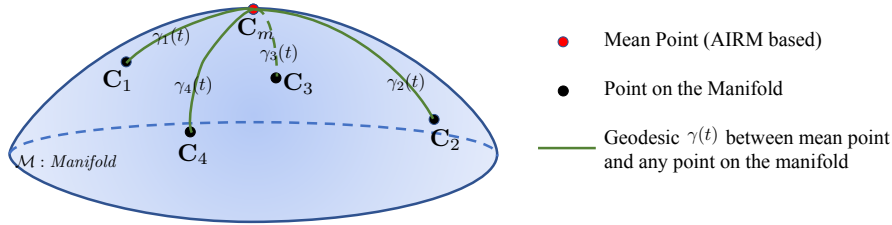


Figure 1.4: The mean point of multiple manifold points.

$$\mathbf{C}_m = \arg \min_{\mathbf{C}} \frac{1}{N} \sum_{i=1}^N \mathcal{L}_{\gamma_i}^2(\mathbf{C}, \mathbf{C}_i) \quad (1.6)$$

Tangent space and tangent vector

Tangent space can be perceived as the spanned space of the tangent curves along the geodesic at a particular point on the manifold, as depicted in Figure 1.5. Tangent space is widely utilized since it is a Hilbert space by definition, allowing for the computation of

1 Preamble

inner products. Just as the inner product is closely associated with distance measurement in Euclidean space, we can leverage tangent space to facilitate the computation of Riemannian distance by computing the inner product between the projected manifold points $\mathbf{S}_1$ and $\mathbf{S}_2$, also known as *tangent vectors*.

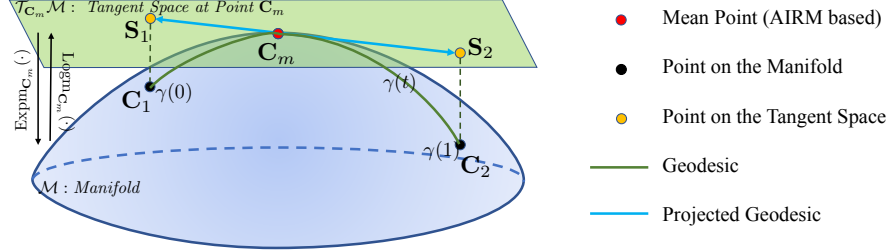


Figure 1.5: Tangent space and tangent vector constructed at the mean point

The projection from the manifold to tangent space is a non-trivial matter because it can vary under different Riemannian metrics. Put differently, given the same pair of manifold points, their Riemannian distance is not necessarily unique and actually depends on the choice of the Riemannian metric. This is analogous to the situation in Euclidean space, where the Manhattan distance [Minkowski, 1910] calculated from the L1 norm differs from the Euclidean distance calculated from the L2 norm. Thus, to explicitly quantify the Riemannian distance, we will subsequently introduce several common Riemannian metrics.

Riemannian Metric

Adopting the metaphor from [Congedo et al., 2017], a metric can be perceived as a ruler used to measure the distance between data points. An inappropriate ruler can introduce excessive complexity in the aftermath, akin to the intricate post-processing steps required after using the Euclidean metric to measure the covariance matrices computed from EEG signals. In contrast, Riemannian metrics appear to be the appropriate ruler to measure EEG covariance matrices, evidenced by the excellent performance of Riemannian methods in EEG decoding. Thus, we will briefly review several Riemannian metrics designed for SPD manifold points.

Affine-invariant Riemannian metric (AIRM)

The concept of affine-invariance dates back to Nomizu’s work [Nomizu, 1954], and the affine-invariant Riemannian metric has been independently proposed in various fields. In information geometry, it is known as the Fisher information metric and has been applied to analyze covariance matrices [Förstner and Moonen, 2003] and diffusion tensors [Fletcher and Joshi, 2004]. The first complete computing framework based on AIRM was published by Pennec *et al.* for interpolation in diffusion tensor imaging (DTI) [Pennec et al., 2006], which was later adopted by Barachant *et al.* for EEG decoding [Barachant et al., 2010], marking the beginning of Riemannian framework’s application in EEG research.

1.2 Riemannian Manifold based EEG Decoding Framework

In mathematical terms, affine transformations, including translation, rotation, scaling, and shearing, are all linear operations that can be modeled as the multiplication of the original signal matrix $\mathbf{X}$ and the transformation matrix $\mathbf{A}$ plus the translation matrix $\mathbf{T}$, i.e., $\tilde{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{T}$, $\mathbf{A} \in GL_N$ and $\mathbf{X} \in \mathbb{R}^{N \times p}$, where $\tilde{\mathbf{X}}$ is the affine-transformed signal matrix, and GL_N denotes the linear group on the SPD manifold $\text{Sym}_+(N)$. Therefore, the affine transformation for a covariance matrix $\mathbf{C} = \mathbb{E}[(\mathbf{X} - \mathbb{E}(\mathbf{X}))(\mathbf{X} - \mathbb{E}(\mathbf{X}))^T]$ can be easily derived as:

$$\tilde{\mathbf{C}} = \mathbf{A}\mathbf{C}\mathbf{A}^T, \mathbf{C} \& \tilde{\mathbf{C}} \in \text{Sym}_+(N), \quad (1.7)$$

where $\tilde{\mathbf{C}}$ denotes the affine transformed covariance matrix, i.e., $\tilde{\mathbf{C}} = \mathbb{E}[(\tilde{\mathbf{X}} - \mathbb{E}(\tilde{\mathbf{X}}))(\tilde{\mathbf{X}} - \mathbb{E}(\tilde{\mathbf{X}}))^T]$. Furthermore, this concept can be extended to the geodesic, which can be further explored in [Pennec et al., 2006]. As the geodesic is closely linked to distance, the affine-invariant distance can be defined as follows [Pennec and Ayache, 1998]:

$$d_{AIRM}(\tilde{\mathbf{C}}_1, \tilde{\mathbf{C}}_2) = d_{AIRM}(\mathbf{A}\mathbf{C}_1\mathbf{A}^T, \mathbf{A}\mathbf{C}_2\mathbf{A}^T) = d_{AIRM}(\mathbf{C}_1, \mathbf{C}_2) \quad (1.8)$$

As shown in [Pennec et al., 2006], if the transformation matrix $\mathbf{A}$ is chosen as $\mathbf{C}_1^{-\frac{1}{2}}$, the distance under AIRM can be simplified as a pseudo-norm as shown below.

$$d_{AIRM}(\mathbf{C}_1, \mathbf{C}_2) = d_{AIRM}(\mathbf{C}_1^{-\frac{1}{2}}\mathbf{C}_1(\mathbf{C}_1^{-\frac{1}{2}})^T, \mathbf{C}_1^{-\frac{1}{2}}\mathbf{C}_2(\mathbf{C}_1^{-\frac{1}{2}})^T) \quad (1.9)$$

$$= d_{AIRM}(\mathbf{I}_N, \mathbf{C}_1^{-\frac{1}{2}}\mathbf{C}_2(\mathbf{C}_1^{-\frac{1}{2}})^T) \quad (1.10)$$

$$:= \mathcal{N}(\Sigma), \quad (1.11)$$

where $\Sigma = \mathbf{C}_1^{-\frac{1}{2}}\mathbf{C}_2(\mathbf{C}_1^{-\frac{1}{2}})^T$. Furthermore, the inverse invariance of this pseudo-norm $\mathcal{N}$ has also been demonstrated in [Pennec et al., 2006] by utilizing the symmetricity of distance, i.e., $\mathcal{N}(\Sigma) = \mathcal{N}(\Sigma^{-1})$. As a result, one possible computation that satisfies these invariances of the pseudo-norm $\mathcal{N}(\cdot)$ is the sum of the squared logarithms of the eigenvalues:

$$\mathcal{N}(\Sigma)^2 = \sum_{i=1}^N (\log(\lambda_i))^2, \quad (1.12)$$

where $\lambda_i, 1 \leq i \leq N$ is the i -th eigenvalue of $\Sigma \in \mathbb{R}^{N \times N}$. This means the Riemannian distance between two manifold points $\mathbf{C}_1$ and $\mathbf{C}_2$ is

$$d_{AIRM}(\mathbf{C}_1, \mathbf{C}_2) = \sqrt{\sum_{i=1}^N (\log(\lambda_i))^2} \quad (1.13)$$

Obviously, the inverse invariance of the pseudo-norm can be easily generalized to the inverse invariance of the Riemannian distance under AIRM, i.e.,

$$d_{AIRM}(\mathbf{C}_1, \mathbf{C}_2) = d_{AIRM}(\mathbf{C}_1^{-1}, \mathbf{C}_2^{-1}) \quad (1.14)$$

1 Preamble

Subsequently, the geodesic that starts from a reference point $\mathbf{C}_{\text{ref}}$ with tangent vector $\mathbf{S}$ can be explicitly derived as:

$$\gamma(\mathbf{C}_{\text{ref}}, \mathbf{S})(t) = \mathbf{C}_{\text{ref}}^{\frac{1}{2}} \text{Exp}(t \cdot \mathbf{C}_{\text{ref}}^{-\frac{1}{2}} \mathbf{S} \mathbf{C}_{\text{ref}}^{-\frac{1}{2}}) \mathbf{C}_{\text{ref}}^{\frac{1}{2}}, \quad (1.15)$$

where $\text{Exp}(\cdot)$ stands for the matrix exponential for SPD matrix, i.e., $\text{Exp}(\mathbf{A}) = \mathbf{V} \exp(\mathbf{D}) \mathbf{V}^T$ with $\mathbf{A} = \mathbf{V} \mathbf{D} \mathbf{V}^T$ and $\mathbf{V}^T \mathbf{V} = \mathbf{I}$.

Note that this geodesic shown in Eq. (1.15) [Pennec et al., 2006] is equivalent to the more general definition at Theorem 6.1.6 in [Bhatia, 2009], i.e., $\gamma(t) = \mathbf{C}_1^{\frac{1}{2}} (\mathbf{C}_1^{-\frac{1}{2}} \mathbf{C}_2 \mathbf{C}_1^{-\frac{1}{2}})^t \mathbf{C}_1^{\frac{1}{2}}$. Their differences are illustrated in Figure 1.6, and the equivalence of these three cases can be easily proved.

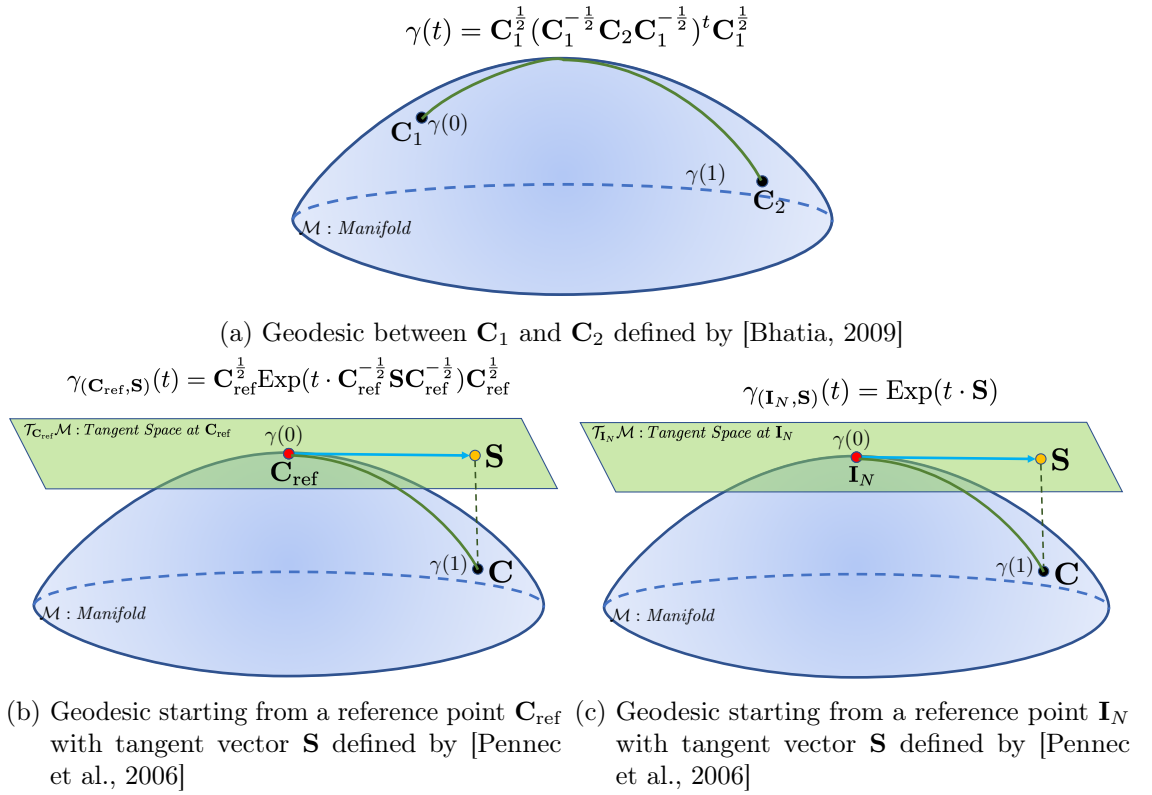


Figure 1.6: Formulation of geodesic under different cases

Based on the geodesic equation, the associated manifold point $\mathbf{C}$ with the given tangent vector $\mathbf{S}$ can be computed, which is located on the tangent space constructed at the reference point $\mathbf{C}_{\text{ref}}$. The equation for this computation is presented below.

$$\mathbf{C} = \text{Exp}_{\mathbf{C}_{\text{ref}}}(\mathbf{S}) = \mathbf{C}_{\text{ref}}^{\frac{1}{2}} \text{Exp}(\mathbf{C}_{\text{ref}}^{-\frac{1}{2}} \mathbf{S} \mathbf{C}_{\text{ref}}^{-\frac{1}{2}}) \mathbf{C}_{\text{ref}}^{\frac{1}{2}}, \quad (1.16)$$

Similarly, the inverse mapping from the manifold point to the tangent vector uses the

1.2 Riemannian Manifold based EEG Decoding Framework

inverse operation, i.e., logarithm mapping, as shown below:

$$\mathbf{S} = \text{Log}_{\mathbf{C}_{\text{ref}}}(\mathbf{C}) = \mathbf{C}_{\text{ref}}^{\frac{1}{2}} \text{Log}(\mathbf{C}_{\text{ref}}^{-\frac{1}{2}} \mathbf{C} \mathbf{C}_{\text{ref}}^{-\frac{1}{2}}) \mathbf{C}_{\text{ref}}^{\frac{1}{2}}, \quad (1.17)$$

where $\text{Log}(\cdot)$ stands for the matrix exponential for SPD matrix and

Additionally, the Riemannian metric provides an inner product definition on the tangent space, denoted as $\mathcal{T}_x \mathcal{M} \times \mathcal{T}_x \mathcal{M} \rightarrow \mathbb{R}$. This inner product, which is also the AIRM distance on the tangent space, can be computed using the logarithm mapping, as shown below:

$$d_{\text{AIRM-TS}}(\mathbf{C}_1, \mathbf{C}_2) = \text{tr}(\text{Log}_{\mathbf{C}_{\text{ref}}}(\mathbf{C}_1)^T \text{Log}_{\mathbf{C}_{\text{ref}}}(\mathbf{C}_2)) \quad (1.18)$$

$$= \text{tr}(\mathbf{S}_1^T \mathbf{S}_2) \quad (1.19)$$

$$= \text{vec}(\mathbf{S}_1)^T \text{vec}(\mathbf{S}_2) \quad (1.20)$$

In summary, the Riemannian distance with AIRM possesses three important invariant properties: affine, inverse, and geodesic (congruence) invariance, which is critical to the success of the Riemannian framework in decoding EEG signals as discussed later in Section 1.2.3. Another popular Riemannian metric on the SPD manifold is the Log-Euclidean metric, which is introduced subsequently.

Log-Euclidean metric (LogE)

The Log-Euclidean metric was proposed as a simpler and faster alternative to the AIRM metric [Arsigny et al., 2006], which has excellent theoretical properties but demands intensive computing resources and is not suitable for real-time applications when dealing with high-dimensional matrices. Despite discarding the affine invariance, the Log-Euclidean metric maintains important invariance properties, such as inversion invariance, similarity invariance, and translation invariance in the domain of logarithms. The simplest version of the Log-Euclidean metric that fulfills these invariance properties is defined as follows:

$$d_{\text{Log-Euclid}}(\mathbf{C}_1, \mathbf{C}_2) = \sqrt{\text{tr}((\text{Log}(\mathbf{C}_1) - \text{Log}(\mathbf{C}_2))^2)} \quad (1.21)$$

Similar to distance, the geodesic under the Log-Euclidean metric is also simplified, as shown in Eq. (1.22).

$$\gamma_{\text{Log-Euclid}}(t) = \text{Exp}((1-t)\text{Log}(\mathbf{C}_1) + t\text{Log}(\mathbf{C}_2)) \quad (1.22)$$

This simplicity leads to another nice property: the Fréchet mean point under the Log-Euclidean metric has a closed-form solution, as opposed to the mean under AIRM, which relies on iterative numerical methods algorithms [Barachant et al., 2010]. However, it is necessary to point out that the mean under two different metrics can be identical under certain cases, as proved in Proposition 9 of [Arsigny et al., 2005].

In summary, the Log-Euclidean metric significantly reduces computational complexity by relaxing the invariance requirements. Although it does not possess affine invariance, it still maintains essential invariance properties in the domain of logarithms. Computationally,

1 Preamble

the Log-Euclidean metric is closely related to AIRM, as both are linked with the scalar geometric mean point. The Log-Euclidean metric is a viable replacement for AIRM when facing limited computing resources or strict real-time requirements while still needing to maintain decent decoding accuracy for EEG signals.

Other Riemannian metrics

Other Riemannian metrics applicable to the SPD manifold include the Log-Cholesky and Wasserstein metrics. The Log-Cholesky metric is advantageous in computational speed and numerical stability and with the closed-form Fréchet mean, but sacrifices some invariance properties (bi-invariance preserved though) [Lin, 2019]. The Wasserstein metric, on the other hand, is based on the product between the space of means and the space of covariance matrices [Chevallier et al., 2017], holds rotation invariance [Chevallier et al., 2017, Congedo and Jain, 2019], but does not have a closed-form solution for the mean point [Congedo and Jain, 2019].

1.2.3 Riemannian Geometry for EEG decoding

A Brief History

Barachant *et al.* proposed the Riemannian manifold based EEG decoding framework [Barachant et al., 2010] by adopting the computing framework in [Pennec et al., 2006]. Meanwhile, Li *et al.* also managed to leverage the Riemannian manifold to decode EEG signals [Li et al., 2009]. The major distinction between these two parallel research lines is that the Riemannian metric used in the pipeline of [Pennec et al., 2006, Barachant et al., 2010] leverages affine-invariance to simplify the computation by avoiding the complicated geodesic definition involving Christoffel symbols and utilizing the symmetric solution of the Sylvester equation in the geodesic, which is not the case for [Li et al., 2009]. Moreover, Riemannian geometry has also been applied to analyze the optical flow of EEG signals by Lefèvre *et al.* [Lefèvre and Baillet, 2008], as EEG signals can be considered as a 3D spatiotemporal signal, similar to video signals. Despite this, due to the powerful theoretical properties and practical simplicity, the EEG decoding pipeline proposed in [Barachant et al., 2010] has gained more attention. As witnessed by multiple champions in several BCI competitions, its success has further attracted more interest from the EEG-BCI community.

Over the past decade, the Riemannian manifold based EEG decoding framework has been widely applied in various paradigms or domains, such as motor imagery [Barachant et al., 2010, Barachant et al., 2013b, Barachant et al., 2011], sleep stage categorization [Li et al., 2009, Li et al., 2012], P300 [Korczowski et al., 2015], SSVEP [Kalunga et al., 2015], artifacts detection [Barachant et al., 2013a, Saifutdinova et al., 2019], emotion recognition [Wang et al., 2021], respiratory states [Navarro-Sune et al., 2016], imagined speech [Bakhshali et al., 2020], post-traumatic stress disorder (PTSD) [Kim et al., 2020], olfactory classification [Zhang et al., 2022], etc.

Riemannian Manifold based EEG Decoding Pipeline

Despite the diversity in applied paradigms and purposes, the adopted algorithms or pipelines can be categorized into two types of classification, as illustrated below.

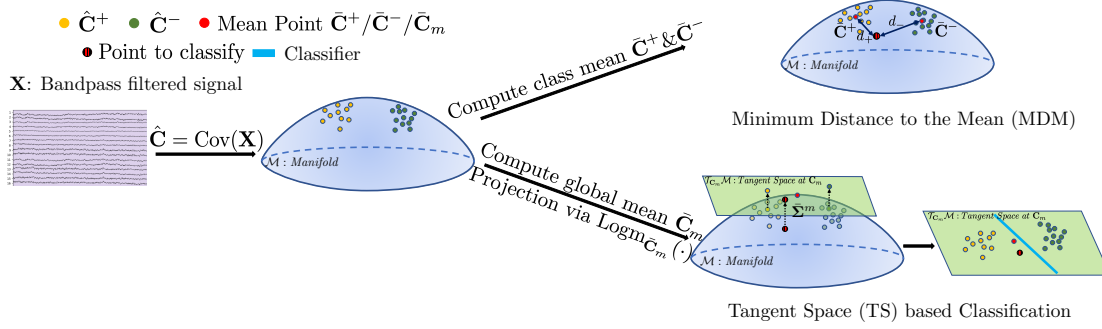


Figure 1.7: Two types of classification pipeline based on the Riemannian manifold.

The first (above) classification approach is the direct comparison of Riemannian distances between classwise mean points and new points on the manifold, commonly referred to as the minimum to the mean (MDM) or nearest centroid classifier [Congedo et al., 2017]. While advantageous for its simplicity and independence from the number of classes, MDM classifier performance is highly dependent on the choice of Riemannian metric (AIRM or Log-Euclidean), the type of mean points on the manifold (arithmetic, geometric, harmonic, or power means [Congedo et al., 2017, Congedo et al., 2019]), and input feature dimensionality [Barachant et al., 2011, Congedo et al., 2017].

In addition to the MDM classifier, the tangent space-based classification is another option, as shown in Figure 1.7 (below). This pipeline involves computing the mean point on the manifold using the chosen Riemannian metric, projecting all manifold points onto the tangent space constructed at the computed mean point using logarithm mapping, and applying standard classification algorithms defined in the Euclidean space to the tangent vectors. The tangent space is a Hilbert space, which means that most classic classification algorithms can be directly applied without adaptation. The superiority of this pipeline over the standard EEG decoding pipeline has been verified in the meta-analyses [Jayaram and Barachant, 2018].

The success of Riemannian manifold based EEG decoding pipelines can be attributed to both the exploration of the geometric structure embedded in the manifold points and the abundance of complex Euclidean classifiers. This framework achieves decent decoding accuracy and satisfactory robustness, mainly due to the theoretical properties held by the Riemannian metric. In a comprehensive review by Congedo et al. [Congedo et al., 2017], several justifications are provided to support the success of the Riemannian framework, which are briefly summarized below. For more information, please refer to [Congedo et al., 2017].

1. *Affine invariance* ensures equivalence between sensor space and source space, making the measured distance robust against changes in the mixing matrix from source

1 Preamble

space to sensor space. This is also relevant to practical scenarios such as the shift of the EEG cap.

2. *Inversion invariance* preserves the sensitivity of the relative scale difference while being insensitive to the common signals, which are inferences or artifacts.
3. Geometric distance and mean are known for their robustness against outliers, which are commonly present in EEG signals. Hence, utilizing such distance and mean can help suppress the influence of outliers.
4. *Congruence invariance* theoretically supports the generalization capabilities of Riemannian frameworks, thereby explaining their robustness in cross-session or cross-subject tasks.

1.3 Decoding Model Interpretability

The success of both deep learning and Riemannian manifold based decoding framework, particularly in the EEG-BCIs field, has significantly enhanced the translation accuracy and the robustness of EEG decoding. These advancements are primarily attributable to the augmented complexity of both the processing pipeline and classification model. However, it is imperative to acknowledge that the incessant augmentation of complexity does not necessarily proportionately enhance accuracy and robustness, and one of the potential drawbacks is the compromised interpretability of the decoding models, which can be particularly unfavorable in a clinical EEG-BCIs system. Moreover, the ability to interpret a decoding model confers the potential to integrate novel techniques into a BCIs system, thereby augmenting its efficacy or broadening its application domain, akin to the pioneering utilization of computers in the early 1970s. Therefore, in this section, we briefly present the background of interpretable machine learning and underscore the significance of interpretability in EEG-BCIs applications. Lastly, we introduce a technique that leverages the model’s interpretability to construct an efficient EEG-BCIs system.

1.3.1 Interpretable Machine Learning Background

As an emerging field, interpretable machine learning lacks a precise and widely-acknowledged definition, as noted in [Murdoch et al., 2019, Molnar, 2022]. However, following the description in [Molnar, 2022], we can describe model interpretability as the degree of understanding the causal factors behind a decision [Miller, 2019] or the consistency of model performance predictions [Kim et al., 2016]. These definitions highlight the importance of interpretable machine learning, which can promote fairness (i.e., unbiased and equitable predictions or decisions across different people or data sets) and accountability (the ability for post hoc inspection) in machine learning models [Chakraborty et al., 2017b]. By providing explanations for how the model arrived at its decision, interpretable machine learning can increase transparency, accountability, and fairness in decision-making processes, which is especially important in fields where model predictions have significant impacts on human lives, such as in loan applications or medical diagnoses.

Concrete methods for achieving interpretability can be found in Section 3.2 of [Molnar, 2022], which categorizes methods from diverse perspectives, such as intrinsic versus post hoc, model-specific versus model-agnostic, and local versus global. A similar but more condensed taxonomy with examples is presented in [Du et al., 2019].

1.3.2 Interpretability for EEG decoding

As mentioned earlier, the interpretability of machine learning models is crucial in health-care systems, especially for clinical applications, where each decision made by the model can have high stakes and be irrevocable. In the context of EEG-BCIs systems, interpreting the models is also necessary to mitigate the risk of using an artifacts-driven system and efficiently train the system using only the most important features, such as task-related spatial patterns in the CSP. The importance of interpretability can be traced back throughout the developmental history of EEG decoding technology, as discussed in Sections 1.1.1 and 1.1.2, where clinicians directly interpreted the EEG signal to associate signal characteristics with certain disease diagnoses, e.g., epilepsy [Pillai and Sperling, 2006, Tatum IV, 2021], migraine [Smyth and Winter, 1964, Sand, 1991], stroke [Jordan, 2004], etc. In more recent times, when the classic EEG decoding model was introduced, intrinsic and model-specific methods were commonly used to interpret the trained models due to their relatively simple structure, such as linear regression and support vector machines, as summarized in Section 1.1.3. One representative example is the topographic interpretation of the common spatial pattern, which can directly reflect the spatial distribution of brain sources via eigenvector and their importance via the eigenvalues [Koles, 1991, Koles and Soong, 1998].

With the recent advances in EEG decoding methods, there are now challenges in interpreting these complex models. However, thanks to intensive research in interpretable machine learning, especially for the deep learning models, there have been several works that successfully interpreted deep learning based EEG decoding frameworks, such as CNN for MI [Borra et al., 2020], emotion recognition [Mayor-Torres et al., 2021], and epileptic seizure prediction [Jemal et al., 2022]. In contrast, the interpretability of Riemannian manifold based EEG decoding frameworks has received less attention from both EEG and interpretable machine learning communities, even though the tangent space-based EEG decoding is often equipped with a simple classifier, which should be relatively easier to interpret. To the author’s knowledge, the only related work is by Wong *et al.*, who presented Riemannian interpretability under the Log-Euclidean metric for analyzing brain networks with resting-state fMRI data. Nevertheless, the interpretability methods taxonomy presented in Section 1.3.1 can serve as a guide for future research on investigating interpretable Riemannian methods, as listed below.

- *Intrinsic vs. Post Hoc*: Intrinsic methods are typically applied to models with simple structures, while post hoc methods are often used for complex models. Riemannian methods should certainly be categorized as the latter type due to their complex geometrical structure. However, finding an approximation with a simple model

1 Preamble

would be a promising way to interpret Riemannian decoding frameworks in an intrinsic way.

- *Model Specific vs. Model Agnostic*: Model-agnostic approaches, specifically model flexibility, should be the ultimate goal for interpreting Riemannian methods. This means that, regardless of the specific classifier used in the pipeline, the interpretability of the Riemannian decoding framework can always be analyzed in the same way.
- *Local vs. Global*: Unlike computer vision or natural language processing problems, personalized predictions that can be interpreted are often preferred in EEG-BCIs systems due to the non-stationary and heterogeneous nature of brain signals. Therefore, local interpretations, such as on a single-trial or single-subject level, are of greater interest.

1.3.3 Non-Invasive Brain Stimulation

The clinical utility of the interpretable EEG decoding model extends beyond the mitigation of artifacts-driven training. The interpretability provides valuable insights into the brain signal features that contribute most significantly to model performance, which can guide feedback signal provision in rehabilitation training. Traditional rehabilitation-oriented EEG-BCIs setups require various types of feedback signals (e.g., visual, auditory, or tactile) to achieve training, which can be considered indirect as they rely on non-brain signals to regulate behavior. However, since behavior is primarily controlled by brain activity, it may be possible to apply direct feedback to intervene with brain signals. Non-invasive brain stimulation (NIBS) is one of the most well-known methods for this purpose.

Unlike traditional brain stimulation methods, such as deep brain stimulation or epidural electrical stimulation, NIBS uses external sources of energy to modulate brain activity without the need for surgery or implants. The two main types of NIBS are transcranial magnetic stimulation (TMS) and transcranial electrical stimulation (TES), which include transcranial direct current stimulation (tDCS) and transcranial alternating current stimulation (tACS). The mechanism of NIBS involves the application of electrical or magnetic fields to the scalp, which can induce changes in the excitability of the neurons in the underlying brain tissue [Terao and Ugawa, 2002, Fertonani and Miniussi, 2017, Reed and Cohen Kadosh, 2018]. NIBS has a wide range of potential applications, including the treatment of depression [Dunlop et al., 2017], chronic pain [Cardenas-Rojas et al., 2020], and stroke [Di Pino et al., 2014], as well as improving memory and attention in healthy individuals [Grover et al., 2022].

The standard protocol of NIBS involves using a fixed-parameter setup for a population of individuals without considering individual differences in brain anatomy or function. However, this setup is incapable of handling the non-stationary brain signal and heterogeneity of the brain. Therefore, the development of personalized and closed-loop (adaptive) NIBS has been suggested to address this problem. Both novel setups require

continuous monitoring of brain activity and adjusting stimulation parameters in real-time based on feedback signals from the brain [Leite et al., 2017, Tervo et al., 2022]. These personalized and closed-loop NIBS approaches have the potential to improve the efficacy and reliability of NIBS for treating neurological and psychiatric disorders and enhancing cognitive function, as they account for individual variability in brain function and allow for more precise and adaptive modulation of neural activity.

1.4 Challenges and Questions

1.4.1 Research Question 1: Existence of Interpretable and Computationally Efficient Riemannian Methods

The exceptional decoding accuracy and robustness of the Riemannian manifold based decoding pipeline has gained considerable attention, particularly owing to the simplification presented in [Pennec et al., 2006], which facilitated the understanding of the Riemannian computing framework among non-mathematicians. Despite this progress, the framework remains computationally intensive, with complex operations such as matrix multiplication, matrix decomposition, and iterative computation of mean points. Such complexity poses challenges for the real-time processing of EEG signals, which is critical for effective communication with EEG-BCIs, especially when processing high-dimensional data [Yger et al., 2016]. Furthermore, the lack of interpretability of the Riemannian decoding framework due to the sophisticated mathematical operations also concerns the clinical or rehabilitation-oriented EEG-BCIs. Because without justifying the features under usage, it is impossible to rule out the influence of intrinsic artifacts on the trained EEG-BCIs system. These challenges faced by current Riemannian methods pose our first research question.

Research Question 1: Can we find an interpretable and computationally efficient way to apply the Riemannian manifold based decoding framework?

1.4.2 Research Question 2: Necessity of Introducing Extra Nonlinearity and Preserving Affine-Invariance

Both deep learning and Riemannian manifold based methods leverage nonlinearity to achieve high performance in EEG decoding. The main difference lies in the stage at which nonlinearity is introduced. Riemannian methods use the adopted Riemannian metric to incorporate nonlinearity, while deep learning models are inherently nonlinear classifiers. Based on the excellent performance of both frameworks, it seems that we do benefit from introducing nonlinearity into the decoding pipeline. Also, preserving affine-invariance will contribute to decent decoding accuracy and robustness is another unverified speculation we concluded before. Therefore, the second research question will be addressed in this dissertation is:

Research Question 2: Whether extra nonlinearity can further boost the performance of the Riemannian manifold based EEG decoding framework

and whether preserving affine-invariance in the Riemannian framework is as important as we argued before ?

1.4.3 Technical Question: Finding an Elegant Solution to Achieve Remote Control of Transcranial Electrical Stimulation Signal

Personalized and closed-loop non-invasive brain stimulation (NIBS) systems, particularly those utilizing transcranial electrical stimulation (TES), have generated significant research interest. However, the implementation of such systems still faces technical obstacles. The manual control of signals or the purchase of costly commercial software for remote TES signal control poses a challenge to related research, particularly closed-loop TES studies. Hence, the final technical question that this dissertation aims to address is:

Technical Question: Can we find a solution to conveniently control the applied transcranial electrical stimulation signal and integrate it with a BCIs system to achieve the closed-loop application?

2 Publications

2.1 Interpretable Riemannian Classification in Brain-Computer Interfacing

2.1.1 Synopsis

This study focuses on addressing the computational complexity and lack of interpretability of Riemannian methods for EEG signal decoding, which can lead to inefficient and artifacts driving classification performance. Therefore, we propose a method for extracting spatial filters from the trained classifier on the Riemannian tangent space and compare it against common spatial patterns. By evaluating the proposed decoding pipeline on multiple EEG data sets, we demonstrate that the proposed method generates filters that approach the standard Riemannian decoding pipeline's performance while also being interpretable, thanks to the associated spatial patterns.

Complete bibliographic information

Xu, J., Grosse-Wentrup, M., and Jayaram, V. (2019). Interpretable Riemannian classification in Brain-Computer Interfacing. *Proceedings of the 8th Graz Brain-Computer Interface Conference (GBCIC) 2019: Bridging Science and Application*. (pp: 32-37). Graz University of Technology, Austria

My contribution

1. Devising the project with co-authors.
2. Implementation of the approximation algorithms
3. Validating the proposed pipelines on multiple motor-magery EEG datasets
4. Paper writing (with helps in polishing and proof-reading)

2.1.2 Publication

INTERPRETABLE RIEMANNIAN CLASSIFICATION IN BRAIN-COMPUTER INTERFACING

Jiachen Xu^{1,2}, Moritz Grosse-Wentrup^{1,3,4}, Vinay Jayaram¹

¹Max Planck Institute for Intelligent Systems, Department of Empirical Inference, Tübingen, Germany

²Technical University of Munich, Department of Electrical and Computer Engineering, Munich, Germany

³Ludwig Maximilian University of Munich, Department of Statistics, Munich, Germany

⁴University of Vienna, Faculty of Computer Science, Vienna, Austria

E-mail: {jiachen.xu, moritzgw, vjayaram}@tuebingen.mpg.de

ABSTRACT: Riemannian methods are currently one of the best ways of building classifiers for EEG data in a brain-computer interface (BCI). However, they are computationally complex and suffer from a lack of interpretability. Since the full covariance matrix is used for each classification, it is not immediately possible to see what underlying signals are generating the classified changes in variance. Particularly in a rehabilitation context, where it is essential to control which brain signals are used for classification, this can be a severely limiting factor. Further, the requirement to perform a matrix logarithm can become prohibitively complex for real-time computation. In this work, we explore a method for extracting spatial filters from a solution in the Riemannian tangent space and compare it against common spatial patterns. We show via comparisons on multiple open-access datasets that it is possible to generate filters that approach the performance of the full Riemannian solution while maintaining interpretability.

INTRODUCTION

One of the major reasons that spatial filtering has enjoyed decades of popularity within the field of brain-computer interfacing is its interpretability. In contrast to most other methods, spatial filters can be directly evaluated to see whether they represent sources of brain activity or correlated noise sources, which is crucial in the fields of rehabilitation and neurofeedback. As they serve the dual purposes of dimensionality reduction and explainability, there has been a large amount of research on how to calculate optimal ones for a given task. From the original paper on common spatial patterns (CSP) [1] to modern divergence-based approaches [2], hundreds of manuscripts have been written on how to find them. In parallel to this development, in the last few years, there has been an upsurge of interest in methods based on Riemannian methods [3, 4]. By considering the sensor covariance matrix as a whole, instead of trying to isolate particular sources of interest, it is possible to overcome many of the non-stationarity issues that have plagued ma-

chine learning in EEG, resulting in tangent-space methods out-performing all others in a recent meta-analysis [5]. In contrast to the often complex optimization methods required for modern approaches to spatial filtering, and the uncertain gains they provide, Riemannian tangent space methods are amenable to standard machine learning algorithms, which significantly increases their ease of use. The downside is that, by using the entire covariance matrix, it is no longer possible to determine what exact sources are generating a difference between experimental conditions. This is of little concern in consumer BCIs, where the goal of the device is usability for healthy subjects, but in the case of neurofeedback or patient groups, this is a severely limiting factor. Proof that a brain signal, and not a correlated physiological signal such as muscular activity, is the driving force for a BCI is necessary for these contexts, and so it is not yet possible to use Riemannian methods there.

Our contribution is the following: Based on recent findings [6], we validate a method for decomposing a linear function in the Riemannian tangent space into a set of spatial filters. We show in tests on multiple datasets that these spatial filters approach the performance of the full Riemannian method and further that they are more robust than CSP to different numbers of filters used. We also show associated spatial patterns to validate that the filters capture relevant brain regions. In summary, our contributions render Riemannian classification methods applicable to BCIs for rehabilitation and neurofeedback.

METHODS

BCI paradigms commonly influence the amplitude of brain rhythms. To create better features, spatial filters have been used to isolate the component of the recorded signal corresponding to a rhythm of interest. These filters are often found by maximizing variance differences, as the variance is a mathematically convenient proxy for amplitude.

One of the most representative and common data-driven spatial filters is CSP [1, 7], which is computed by find-

ing the sources that maximize the variance ratio between two conditions. CSP has made a great contribution to the stable increase of BCI's performance and has over the years spawned many improved versions, such as filter bank CSP [8], regularized CSP [9] and divergence-based CSP [2]. In addition to increased decoding performance, another important reason behind the popularity of CSP is its interpretability. By observing the spatial patterns associated with discovered spatial filters, it is possible to verify whether they are related to neural or other sources. Unlike spatial filtering, which uses variance-based features, Riemannian methods are based on directly comparing sensor covariance matrices. In this paper, the adopted Riemannian metric is called the affine-invariant Riemannian metric (AIRM) [10], where the "affine-invariance" means that the distance between two symmetric positive definite (SPD) matrices is invariant to the distance after applying an affine transformation to both matrices [11], i.e.,

$$\delta_R(\mathbf{TC}_1, \mathbf{TC}_2) = \delta_R(\mathbf{C}_1, \mathbf{C}_2) = \|\text{Log}(\mathbf{C}_1^{-1}\mathbf{C}_2)\|_F, \quad (1)$$

where $\|\cdot\|_F$ represents the Frobenius norm, $\text{Log}(\cdot)$ means the logarithm of a matrix, $\mathbf{C}_{(\cdot)}$ are SPD matrices, and $\mathbf{T}$ is any affine transformation matrix. While this method returns a very robust estimate of distance, most machine learning algorithms require a Euclidean space to work. This can be done via projecting manifold points onto the tangent space to a given point.

The tangent space is defined as a real vector space centered around a reference point on the manifold. In practice, this point is normally selected as the mean point across all trials [10]. After fixing the reference point, the covariance matrices on the manifold are mapped onto this space using the logarithmic map (Eq. 2) and then vectorized into the tangent vectors (TVs).

$$\text{Log}_{\mathbf{C}_1}(\mathbf{C}_2) = \mathbf{C}_1^{\frac{1}{2}} \text{Log}\left(\mathbf{C}_1^{-\frac{1}{2}} \mathbf{C}_2 \mathbf{C}_1^{-\frac{1}{2}}\right) \mathbf{C}_1^{\frac{1}{2}}, \quad (2)$$

$$\text{Exp}_{\mathbf{C}_1}(\mathbf{C}_2) = \mathbf{C}_1^{\frac{1}{2}} \text{Exp}\left(\mathbf{C}_1^{-\frac{1}{2}} \mathbf{C}_2 \mathbf{C}_1^{-\frac{1}{2}}\right) \mathbf{C}_1^{\frac{1}{2}}, \quad (3)$$

where $\text{Exp}(\cdot)$ means the exponential of a matrix. Eq. 2 represents the logarithm mapping of $\mathbf{C}_2$ from the manifold to the TS based on the point $\mathbf{C}_1$ and Eq. 3 shows the inverse projection, i.e., the exponential mapping. For the affine-invariant and log-Euclidean metrics, the logarithmic map requires computing the matrix logarithm of its argument, as a way of mapping the eigenvalues of the input matrices from the strictly positive space of SPD matrices to a real-valued space. As the tangent space represents a linearization of the manifold, distances between points approximate true manifold distances, and geometry on the manifold is preserved, which allows for linear machine learning methods to be used.

A major drawback of this procedure is the difficulty in interpreting the resulting classifier because the tangent space embedding cannot be easily linked to any intuitive

physical process. We address this problem by introducing a procedure to extract spatial filters from a linear function in the tangent space, which allows for a unified methodology in the computation of spatial filters that can also take advantage of the large body of work on machine learning in linear spaces.

The procedure can be summarized in three steps: Classification, spatial filter generation, and filter selection. For classification, we follow the approach in [12] (linked to point 2 to point 5 in Algorithm 1, abbreviated as p2 to p5 afterwards) and find the optimal SVM solution in the affine-invariant tangent space (p6). Though any algorithm is possible, we chose a regularized SVM as maximum-margin classifiers can deal well with the high dimensionality of the tangent space. Once a linear classifier is computed, we project the weight vector in the tangent space back to the SPD manifold via the exponential map (Eq. 3) (p8 to p9). From here, we compute spatial filters by taking the solution to the generalized eigenvalue (GE) problem between the tangent space reference point and the exponential map of the weight vector (p10).

Algorithm 1 Extraction of Tangent Space Spatial Filters

INPUT: Band-pass filtered raw signal $\mathbf{X}_i$, $\forall i = 1, \dots, T$

OUTPUT: Spatial filter after ordering $\mathbf{F}$

- 1: **procedure** I. CLASSIFICATION
 - 2: $\mathbf{C}_i = \mathbf{X}_i \mathbf{X}_i^T$ $\forall i = 1, \dots, T$
 - 3: $\mathbf{C}_m = \arg \min_{\mathbf{C} \in \mathcal{C}(n)} \sum_{i=1}^T \delta_R^2(\mathbf{C}, \mathbf{C}_i)$ {AIRM mean}
 - 4: $\mathbf{S}_i = \text{Log}_{\mathbf{C}_m}(\mathbf{C}_i)$ {Mapping from manifold to TS}
 - 5: $\vec{S}_i = \text{vec}(\mathbf{S}_i)$ {Vectorize the upper triangular matrix}
 - 6: $\vec{W} = \text{SVM}(\vec{S}_i, y_i)$ {Weight vector on the TS}
 - 7: **procedure** II. SPATIAL FILTER GENERATION
 - 8: $\mathbf{S}_w = \text{devec}(\vec{W})$ {Convert the vector to SPD matrix}
 - 9: $\mathbf{C}_w = \text{Exp}_{\mathbf{C}_m}(\mathbf{S}_w)$ {Mapping from TS to manifold}
 - 10: $\mathbf{C}_w \mathbf{V} = \mathbf{C}_m \mathbf{V} \mathbf{T}$ {Mapping from TS to manifold}
 - 11: **procedure** III. FILTER SELECTION
 - 12: $\tilde{\mathbf{X}}_i = \mathbf{V}^T \mathbf{X}_i$ {Filter raw signal using full components}
 - 13: $\vec{f}_i = \log(\text{var}(\tilde{\mathbf{X}}_i))$ {Compute the log-variance}
 - 14: $\vec{\beta} = \text{LinearRegression}(\vec{f}_i, y_i)$ {Weight vector of fitted classifier}
 - 15: $\vec{d} = \text{sort}(|\vec{\beta}|)$ {Sort the absolute value of weight coefficient in descending order}
 - 16: $\mathbf{F} = \mathbf{V}[:, \vec{d}]$ {Get the re-ordered spatial filter}
-

The eigenvectors that solve a GE problem simultaneously diagonalize both input matrices, which can be interpreted as finding a space where the weight SPD matrix and the input SPD matrices are all roughly diagonal. This independence means that the logarithm of the matrices in this space is close to the log-variances of each of the filtered dimensions, allowing us to approximate the tangent space linear function as a linear function of the log-variances

Table 1: Summary of all MOABB datasets with left-hand versus right-hand motor imagery. The dataset marked in red color has only 3 channels and is hence excluded from this analysis.

Dataset Name	#Channels	#Subjects	#Sessions
BNCI 2014-001	22	9	2
BNCI 2014-004	3	9	5
Cho et al. 2017	64	49	1
Munich Motor Imagery	128	10	1
Physionet Motor Imagery	64	109	1
Shin et al. 2017	25	29	3
Weibo et al. 2014	60	10	1
Zhou et al. 2016	14	4	3

in the filtered space. However, the associated eigenvalues have no relationship to the linear function, requiring another criterion in order to determine which of the computed filters are most relevant to classification. In order to do this, we fit a linear function to the log-variances of our projected training data (12 to 14) and attempt to reconstruct the function values of the true tangent space function. Based on the absolute values of the coefficients from this regression problem, we rank the GE solution filters (15 to 16).

To evaluate the performance of our method in a rigorous and reproducible fashion, we adopted the datasets and framework from the open source benchmark *Mother of all BCI Benchmark (MOABB)* in our work [5]. For simplicity, we choose left- versus right-hand motor imagery, which is one of the most common paradigms in EEG-based BCIs. We restrict our analysis to the α - and β -bands (8Hz $\sim$ 32Hz). We use all available channels except those used for electrooculography (EOG). We exclude the dataset *BNCI 2014-004* from our analysis (marked in red in Table 1), because this data set has only three channels.

After bandpass filtering the raw EEG signal, the corresponding covariance matrices are estimated via the empirical covariance estimator. Afterward, the spatial filter is extracted according to the pseudocode shown in Algorithm 1, and the classifier is fit using the log-variances of the filtered signal. As for the classification algorithm, we choose Fisher’s Linear Discriminant Analysis (LDA) for simplicity. Decoding accuracy is estimated via five-fold cross-validation within each data set.

Two statistical parameters are utilized to compare our method against CSP and the full Riemannian approach: the p -values and the effect sizes. The p -value for the one-sided test is computed within each dataset, whose null hypothesis is that the median accuracy of using CSP is not larger than that of applying the spatial filter designed by us. Effect sizes show the standardized mean difference (SMD) between the accuracies of the two compared methods. All statistical tests are implemented as described in [5].

RESULTS

Meta-analysis:

As a spatial filtering technique, the natural comparison

for TS SF is against CSP. Therefore, we compared them under three typical scenarios: applying very few filters (for speed), using filters corresponding to half the number of channels (for the compromise between speed and accuracy) and using all filters (for accuracy). To ensure the results are rigorous, the comparison is done across seven datasets from MOABB [5] (as shown in Table 1).

When merely applying two spatial filters (Fig. 1a), TS SF is generally better than CSP, but the positive significance is only present in two datasets: *Shin* and *Munich*. After increasing the filter number to the half the channel dimensionality (Fig. 1b), the positive significance appears in three more datasets *Weibo*, *Physionet* and *Cho*, in addition to an increased meta-effect. Moreover, the SMD of data set *Munich* has increased to 1.5 which implies a strong effect. Lastly, when using all filters (Fig. 1c), the effect sizes are lessened in the datasets with significant performance, except for *Weibo*.

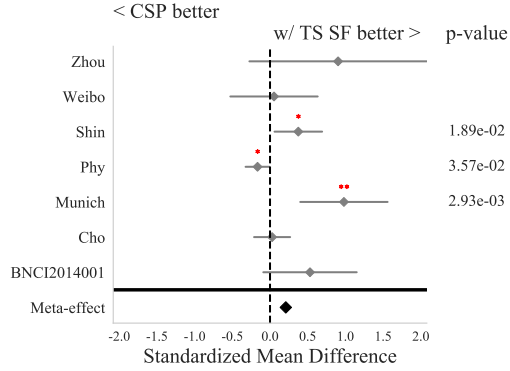
Classification accuracy w.r.t. the number of applied filters:

In addition to comparing against CSP, we are also interested in seeing how well TS SF can approximate the accuracy of the full tangent space classifier, as well as how this approximation evolves as a function of the number of filters. Therefore, we choose three representative datasets *Munich*, *Cho*, *Physionet* and plot their accuracy w.r.t. the number of applied filters in Fig. 2, showing the tangent space accuracy as well as an upper bound.

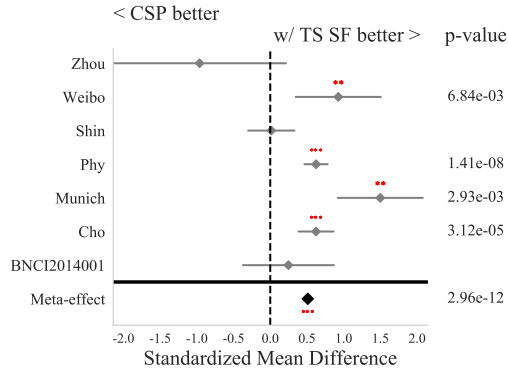
When using data set *Munich* (Fig. 2a), TS SF (green line) first outperforms CSP (red line) when the number of applied filters is less than 16. Afterward, their accuracies are tied with each other until using more than 40 filters. Subsequently, TS SF maintains its stable performance, but the accuracy of CSP starts to decrease.

The information conveyed by Fig. 2b and Fig. 2c is quite similar to each other. TS SF and CSP perform very comparably up until around 13 filters are used. After this point, using more CSP components hurts the performance, while the performance stays much more stable when using more TS SF filters. However, in Fig. 2c, the accuracy drops when using TS SF at high numbers of filters.

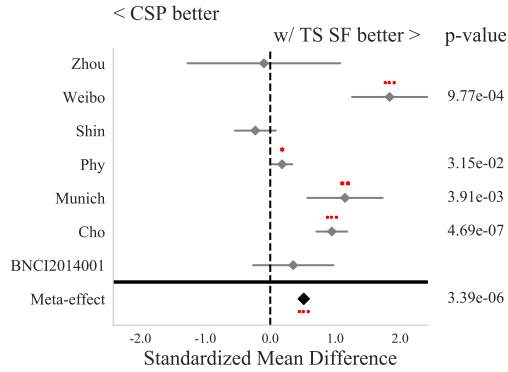
Overall, the performance of the full Riemannian method (Full TS, brown line) is always superior to both spatial filtering methods because the full approach utilizes the full information decoded in the covariance matrices.



(a) Applying the first two filters



(b) Applying the first half of all filters



(c) Applying all filters

Figure 1: Meta analysis of accuracy after applying CSP or TS SF as spatial filter. Parameters: p-value p and SMD are computed within each dataset. Red *, ** and *** represent $p < 0.05, 0.01, 0.001$ respectively. Grey diamonds signify the SMD, while grey bars show the confidence interval of the mean.

Spatial patterns:

From Fig. 1 and Fig. 2 we know that TS SF on average outperforms CSP in terms of accuracy. For a spatial filtering method, however, its interpretation is as crucial as its classification accuracy. Therefore, we compared the

corresponding spatial patterns of both TS SF and CSP in order to see what sources were chosen. We choose the data set *Munich* because it has the best overall data quality [5] and highest channel number (128), and plot the two best filters using both CSP and TS SF. In order to convert spatial filters into spatial patterns, we use the approach proposed by [13].

When comparing Fig. 3a and 3b there are some interesting results. First, the spatial patterns from CSP tend to be more patchy than from TS SF, e.g., S1, S2, S10, in particular for those subjects noted as having low data quality [14]. Second, for the subjects who have excellent signal quality, the spatial patterns from TS SF are highly consistent with from CSP, e.g., S3, S4, and S6. Third, when looking at S2, the pattern from TS SF is more interpretable comparing to the patchy pattern from CSP. When solely focusing on Fig. 3b, it is obvious to see that for S2, S3, S4, S5 and S6, they represent strong signal sources surrounding the sensorimotor cortex which is consistent with this paradigm. This is in contrast to, e.g., S7 and S10, where CSP clearly picks up ocular artifacts.

DISCUSSION

We show that it is possible to construct spatial filters that can approximate the performance of tangent space linear functions, and validate this result on a variety of open-source datasets. In our simulations, it can be seen that the TS SF out-performs CSP, in particular in the cases where the full tangent space function out-performs CSP. Further, we show in the *Munich* dataset that the spatial filters recovered by TS SF are more robust to noisy data than those from CSP. With these advances, it is possible to use Riemannian methods in neurofeedback and clinical work. To evaluate the spatial filter, the first two essential indicators are the classification accuracy and the quality of spatial patterns.

From the perspective of performance, TS SF is still imperfect. First, as shown in Fig. 2, TS SF is constantly worse than the full Riemannian methods. Hence, to move forward, it is important to diminish the approximation error as much as possible, especially when using very few filters. The choice of how to determine filter relevance, though effective, can also be improved. Regularized linear regression approaches, such as LASSO or ridge regression, may better isolate good filters, or using another target variable.

The spatial patterns associated with TS SF are robust to artifacts. As shown in Fig. 3, the reflected neural signal sources after filtering by TS SF is more reasonable and interpretable than by CSP, especially for the low-quality signal. For example, for S2 of the data set *Munich* where the percentage of trials contaminated by artifacts reaches 55% [14], CSP merely collects the artifacts whereas TS SF presents the strong sources around sensorimotor cortex which is consistent with the chosen paradigm. Nonetheless, for the subjects with high data quality, TS SF is still as good as CSP (e.g., for S3 and S4,

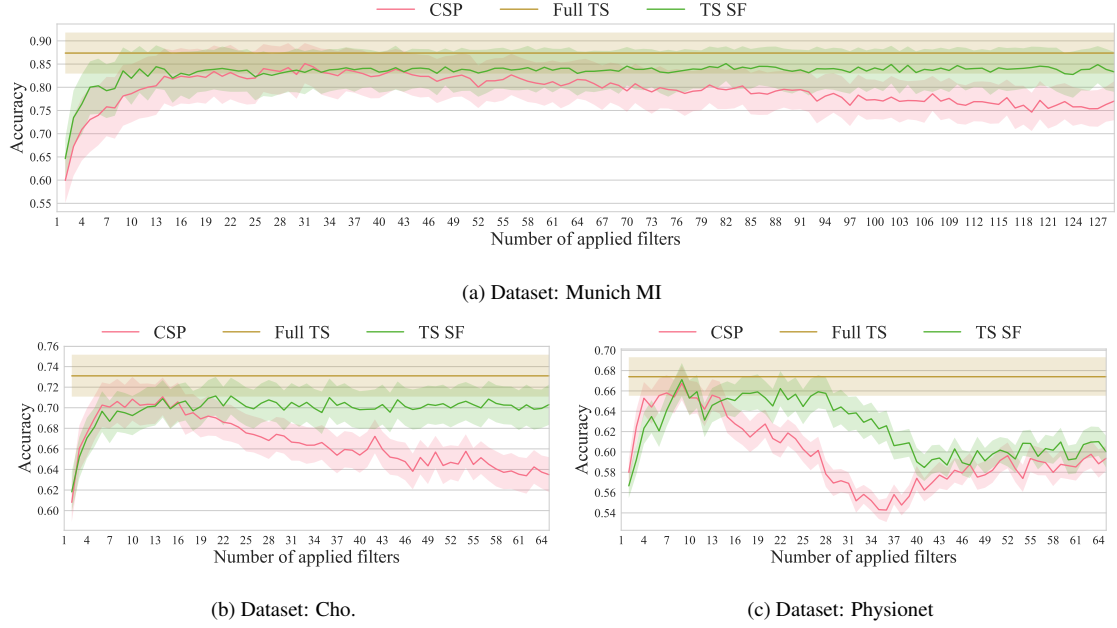


Figure 2: Accuracies comparison among three different pipelines and for each number of applied filters, the central line and the error band are computed across all subjects and sessions. **1. CSP:** Choose CSP as the spatial filter and classified by LDA. **2. Full TS:** Use the standard Riemannian classification framework with AIRM without any spatial filtering and classified by SVM. **3. TS SF:** Apply the spatial filters which are extracted from the tangent space and classified by LDA. Error band setting: confidence interval = 68%. Central line: the mean accuracy. It is not difficult to notice that TS SF has a better and more robust performance than CSP w.r.t. the increase of the number of applied filters.

where 5.0% and 6.3% of trials are affected by artifacts respectively [14]). Therefore, as far as the interpretability is concerned, TS SF is more robust than CSP.

As shown in Fig. 2, it is obvious that TS SF is always more robust than CSP considering the tendency of the accuracy w.r.t. the filter numbers. However, when focusing at Fig. 2c, we could find an evident fluctuation for TS SF. Accuracies drop at high numbers of filters in contrast to the other datasets, and the overall range of mean accuracies is far lower as well. Much of this can likely be explained by the low data quality of data set *Physionet*, which would lead to worse solutions of the initial tangent space problem that likely translate into poorer spatial filters. However, it is important to understand what conditions lead to better or worse TS SF solutions, especially when applying full filters. Also, it should be explored more that why is TS SF still more robust than CSP to filter number even when TS SF performs badly.

Usually, the more robust performance demand more complex computation, but this argument does not always hold. Although TS SF requires a tangent space function to be computed as an initial step which means the computation of the filters themselves is far more intensive than that of CSP, however, in the online application, the procedure is identical: the data is filtered via a linear projection and then the log variances of the resulting time series are used as features for a final classifier. When comparing

to the standard Riemannian classification framework, TS SF leads to an approximation error. However, the computation for online classification would be significantly reduced, especially with large channel numbers. Given that the loss in accuracy on average is relatively small, this may represent a feasible way of using Riemannian methods in high-dimensional, online settings, and further of using them to increase the efficacy of clinical neurofeedback.

CONCLUSION

In this paper, we propose a new method to generate spatial filters from the Riemannian classification framework, Tangent Space Spatial Filters (TS SF), and after comparing between CSP and TS SF, we found that TS SF has better and more robust performance than CSP with the same computational cost online. By extracting spatial filters, the Riemannian approach becomes more interpretable without much loss of accuracy.

However, we still do not have a strong mathematical argument for why this simplification works and the interpretation of the weight vector after projection to the SPD manifold. Nonetheless, the undeniable fact is that the performance of TS SF reveals to us the potential of interpretable Riemannian approaches in BCIs.

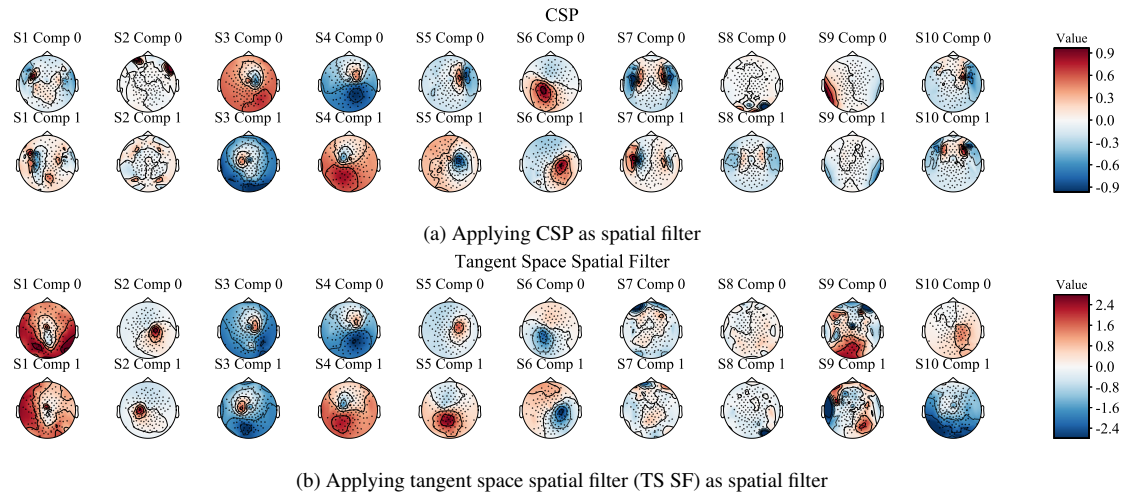


Figure 3: The spatial patterns of data set *Munich* associated with applying the first two spatial filters.

REFERENCES

- [1] Zoltan J Koles, Michael S Lazar, and Steven Z Zhou. "Spatial patterns underlying population differences in the background EEG". In: *Brain Topography* 2.4 (1990), pp. 275–284.
- [2] Wojciech Samek, Motoaki Kawanabe, and Klaus-Robert Müller. "Divergence-based framework for common spatial patterns algorithms". In: *IEEE Reviews in Biomedical Engineering* 7 (2014), pp. 50–72.
- [3] Marco Congedo, Alexandre Barachant, and Rajendra Bhatia. "Riemannian geometry for EEG-based brain-computer interfaces; a primer and a review". In: *Brain-Computer Interfaces* 4.3 (2017), pp. 155–174.
- [4] Florian Yger, Maxime Berar, and Fabien Lotte. "Riemannian approaches in brain-computer interfaces: a review". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 25.10 (2017), pp. 1753–1762.
- [5] Vinay Jayaram and Alexandre Barachant. "MOABB: Trustworthy algorithm benchmarking for BCIs". In: *Journal of Neural Engineering* (2018).
- [6] Alexandre Barachant, Jason Carmel, Kathleen Friel, and Disha Gupta. "Extraction of motor patterns from joint EEG/EMG recording: A Riemannian Geometry approach". In: *6th International Brain-Computer Interface Meeting*. 2016.
- [7] Herbert Ramoser, Johannes Muller-Gerking, and Gert 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] Kai Keng Ang, Zheng Yang Chin, Chuanchu Wang, Cuntai Guan, and Haihong Zhang. "Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b". In: *Frontiers in Neuroscience* 6 (2012), p. 39.
- [9] Fabien Lotte and Cuntai 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.
- [10] Xavier Pennec, Pierre Fillard, and Nicholas Ayache. "A Riemannian framework for tensor computing". In: *International Journal of Computer Vision* 66.1 (2006), pp. 41–66.
- [11] Xavier Pennec and Nicholas Ayache. "Uniform distribution, distance and expectation problems for geometric features processing". In: *Journal of Mathematical Imaging and Vision* 9.1 (1998), pp. 49–67.
- [12] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. "Riemannian geometry applied to BCI classification". In: *International Conference on Latent Variable Analysis and Signal Separation*. Springer. 2010, pp. 629–636.
- [13] Stefan Haufe et al. "On the interpretation of weight vectors of linear models in multivariate neuroimaging". In: *Neuroimage* 87 (2014), pp. 96–110.
- [14] Moritz Grosse-Wentrup, Christian Liefhold, Klaus Gramann, and Martin Buss. "Beamforming in noninvasive brain-computer interfaces". In: *IEEE Transactions on Biomedical Engineering* 56.4 (2009), pp. 1209–1219.

2.2 Tangent Space Spatial Filters (TSSF) for Interpretable and Efficient Riemannian Classification

2.2.1 Synopsis

This paper extends the previous study that presented approximation algorithms for Riemannian methods used in brain-computer interfaces. This study provides rigorous proofs that theoretically justify the approximation algorithm. Moreover, we also demonstrate a one-step classification framework that further reduces the computational complexity and suppresses the time cost by up to 97% in the testing stage. Based on the presented results, we further analyzed the possible reason behind the robustness of Riemannian methods from the interpretability perspective and discussed the possible indication hinted by the optimal number of the spatial filter.

Complete bibliographic information

Xu, J., Grosse-Wentrup, M., and Jayaram, V. (2020). Tangent space spatial filters for interpretable and efficient Riemannian classification. *Journal of Neural Engineering*, 17(2), 026043.

My contribution

1. Devising the project with co-authors.
2. Derivation of the theoretical results and proof of the important equivalence theorem (Details refer to the appendix)
3. Implementation of code for the whole TSSF framework on the basis of code used for 2.1
4. Validating the proposed pipelines on multiple motor-magery EEG datasets
5. Paper writing (with helps in polishing and proof-reading)

2.2.2 Publication

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 raw sensor data as $\mathbf{X} \in \mathbb{R}^{C \times N \times T}$, 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 $c \in \{1, \dots, C\}$) as $\mathbf{X}_c$. In addition, the data from the t -th trial (with $t \in \{1, \dots, T\}$) 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 $\mathbf{C} \in \mathbb{R}^{C \times C \times T}$. 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:

$$\begin{aligned} \text{Logm}(\mathbf{A}) &= \mathbf{V} \log(\mathbf{D}) \mathbf{V}^T \\ \text{Expn}(\mathbf{A}) &= \mathbf{V} \exp(\mathbf{D}) \mathbf{V}^T \\ \mathbf{A}^p &= \mathbf{V} \mathbf{D}^p \mathbf{V}^T, \quad p \in \mathbb{R} \text{ and } p \neq 0, \end{aligned} \quad (\text{II.1})$$

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:

$$\begin{aligned} \text{vec}(\mathbf{A}) &= [\alpha_{1,1} \mathbf{A}_{1,1}, \dots, \alpha_{i,j} \mathbf{A}_{i,j}, \dots, \alpha_{C,C} \mathbf{A}_{C,C}] \\ &\in \mathbb{R}^{1 \times \frac{C(C+1)}{2}}, \text{ where } 1 \leq j \leq i \leq C. \\ \alpha_{i,j} &= \begin{cases} 1 & \text{if } i = j, \sqrt{2} \text{ else} \end{cases} \end{aligned} \quad (\text{II.2})$$

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

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

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:

$$\mathbf{C}^m = \arg \min_{\mathbf{A} \in \mathbf{C}} \sum_{t=1}^T d_{\text{AIRM}}^2(\mathbf{A}, \mathbf{C}^t), \quad (\text{II.4})$$

where $\mathbf{A} \in \mathbb{R}^{C \times C}$ and $\mathbf{C} = [\mathbf{C}^1, \dots, \mathbf{C}^T] \in \mathbb{R}^{C \times C \times T}$.

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:

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

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 $\text{Exp}_{\mathbf{B}}(\mathbf{A})$, i.e., the logarithm and exponential of $\mathbf{A}$ based on another SPD matrix $\mathbf{B}$, is defined as [19, 36]:

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

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:

$$\vec{s}_t = \text{vec}(\mathbf{S}^t) \in \mathbb{R}^{\frac{C(C+1)}{2} \times 1}, \quad (\text{II.7})$$

where $\vec{s}_t$ 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 maximizing the variance ratio between common and discriminative activity of both conditions, which are as defined below:

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

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

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

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:

$$\hat{y}_t = \vec{w}^T \vec{s}_t + b \in \mathbb{R}^{1 \times 1}, \quad \forall t = 1, \dots, T, \quad (\text{III.1})$$

where $\vec{w}$ 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:

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

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

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

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

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., $(\mathbf{C}^m)_{\text{new}} = \mathbf{F}^T \mathbf{C}^m \mathbf{F}$.

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:

$$\mathbf{F}_K^* = \arg \min_{\mathbf{F}_K \in \mathbb{R}^{C \times K}} \sum_{t=1}^T (\hat{y}_t - y_t^{\text{approx}} |_{\mathbf{F}_K})^2, \quad (\text{III.4})$$

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:

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

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., $\mathbf{F}^T \mathbf{C}^t \mathbf{F}$ is a diagonally dominant matrix for all t , based on the Gershgorin circle theorem [41] we know that

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

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

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

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

$$y_t^{\text{approx}} \approx \text{Tr}(\text{Logm}(\mathbf{D}^w) \text{Logm}(\mathbf{D}^t)) = \log(\vec{d}^w)^T \log(\vec{d}^t), \quad (\text{III.8})$$

where $\vec{d}^{(\cdot)}$ 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:

$$\mathbf{C}^w \mathbf{F} = \mathbf{C}^m \mathbf{F} \mathbf{D} \Leftrightarrow \begin{cases} \mathbf{F}^T \mathbf{C}^m \mathbf{F} = \mathbf{I} \\ \mathbf{F}^T \mathbf{C}^w \mathbf{F} = \mathbf{D} \end{cases}, \quad (\text{III.9})$$

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., $\mathbf{F}_K^* = \arg \max_{\mathbf{F}_K \in \mathbb{R}^{C \times K}} \sum^T (\hat{y}_t - y_t^{\text{approx}} |_{\mathbf{F}_K})^2$, 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

$$\hat{y}_t \equiv y_t^{\text{approx}}, \text{ iff } \mathbf{F} \text{ is extracted via GED}(\mathbf{C}^w, \mathbf{C}^m) \quad (\text{III.10})$$

and with full rank.

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

$$\mathbf{F}_K^* = \arg \min_{\mathbf{F}_K \in \mathbb{R}^{C \times K}} \sum^T (y_t^{\text{approx}} |_{\mathbf{F}} - y_t^{\text{approx}} |_{\mathbf{F}_K})^2 \quad (\text{III.11})$$

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 $\text{Logm}()$ 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 $\mathbf{F}^T \mathbf{C}^t \mathbf{F}$ 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., $\log(\vec{d}^w)$. 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.,:

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

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].

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

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

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

$$\vec{w}_{\text{LDA}} = \mu^{(+)} - \mu^{(-)} \in \mathbb{R}^{\frac{C(C+1)}{2} \times 1} \quad (\text{III.14})$$

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

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

where $\overline{\mathbf{S}^{(+)}}$ 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.,

$$\overline{\mathbf{C}^{(+)}} - \overline{\mathbf{C}^{(-)}} = \text{Exp}_{\mathbf{C}^m}(\overline{\mathbf{S}^{(+)}} - \overline{\mathbf{S}^{(-)}}) \quad (\text{III.16})$$

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

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

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 $\mathbf{A}\mathbf{V} = \mathbf{B}\mathbf{V}\mathbf{\Gamma}$ ($\mathbf{\Gamma}$ 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:

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

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:

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

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 $\mathbf{X} \in \mathbb{R}^{C \times N \times T}$, loss function for linear model L

Result: TSSF and regression coefficients with K components: $\mathbf{F}_K \in \mathbb{R}^{C \times K}$, $\vec{\beta}_K \in \mathbb{R}^{K \times 1}$

begin

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

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
$\log(\text{diag}(\tilde{\mathbf{C}}_{\perp \mathbf{F}}^t))$	Log-variance	Log-var
$\text{diag}(\text{Logm}_1(\tilde{\mathbf{C}}_{\perp \mathbf{F}}^t))$	Diagonal of logarithm of covariance matrices	Diag. log-cov
$\text{vec}(\text{Logm}_{\tilde{\mathbf{C}}_{\perp \mathbf{F}}^m}(\tilde{\mathbf{C}}_{\perp \mathbf{F}}^t))$	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 $\mathbf{X} \in \mathbb{R}^{C \times N \times T}$, Second classifier Clf_2 if needed

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

begin

1. Filter the test data: $\tilde{\mathbf{X}}_{\perp \mathbf{F}_K} = \mathbf{F}_K^\top \mathbf{X} \in \mathbb{R}^{K \times N}$
2. Compute features (several options are provided, only choose one):
 - a). $\vec{e} = \log(\text{var}(\tilde{\mathbf{X}}_{\perp \mathbf{F}_K})) \in \mathbb{R}^{K \times 1}$
 - b). $\vec{e} = \text{diag}(\text{Logm}(\text{Cov}(\tilde{\mathbf{X}}_{\perp \mathbf{F}_K}))) \in \mathbb{R}^{K \times 1}$
 - c). $\vec{e} = \text{vec}(\text{Logm}(\text{Cov}(\tilde{\mathbf{X}}_{\perp \mathbf{F}_K}))) \in \mathbb{R}^{K \times 1}$
3. Return label (several options are provided, only choose one):
 - a). One-step classification (only applicable for features from 2.a) or 2.b)):
 - i). $\hat{y} = \text{sgn}(\vec{\beta}_K^\top \vec{e})$
 - b). Two-steps classification (applicable for all features):
 - i). Use a set of $\vec{e}$ 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 ~ 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.

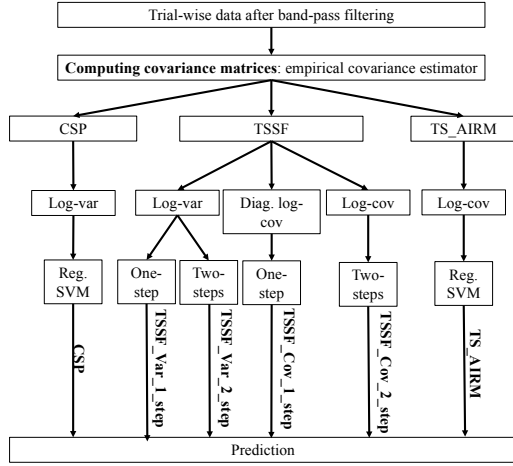


Fig. III.1: All tested pipelines in this paper. The annotated text above the line linked between classifiers and predictions is the abbreviation of the corresponding pipeline and Reg. SVM is the abbreviation of L2 regularized SVM.

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 quantitative 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

		Number of filters: 2							Number of filters: 6				
		$t=-0.69$ $p=1e+00$ (0.0%)	$t=-0.68$ $p=1e+00$ (0.0%)	$t=-0.63$ $p=1e+00$ (0.0%)	$t=-0.64$ $p=1e+00$ (0.0%)	$t=-0.51$ $p=1e+00$ (0.0%)			$t=-0.76$ $p=1e+00$ (0.0%)	$t=-0.64$ $p=1e+00$ (0.0%)	$t=-0.69$ $p=1e+00$ (0.0%)	$t=-0.60$ $p=1e+00$ (0.0%)	$t=-0.44$ $p=1e+00$ (0.0%)
CSP							CSP						
TSSF_Cov_2_step		$t=0.69$ $p=3e-13$ (85.7%)		$t=0.05$ $p=3e-01$ (0.0%)	$t=0.06$ $p=4e-01$ (14.3%)	$t=0.23$ $p=2e-02$ (28.6%)	TSSF_Cov_2_step	$t=0.76$ $p=2e-18$ (85.7%)		$t=0.19$ $p=5e-03$ (42.9%)	$t=0.05$ $p=3e-01$ (14.3%)	$t=0.36$ $p=5e-04$ (42.9%)	$t=0.20$ $p=9e-04$ (28.6%)
TSSF_Cov_1_step		$t=0.68$ $p=1e-13$ (71.4%)	$t=-0.05$ $p=7e-01$ (0.0%)		$t=0.01$ $p=5e-01$ (0.0%)	$t=-0.04$ $p=6e-01$ (0.0%)	TSSF_Cov_1_step	$t=0.64$ $p=2e-13$ (71.4%)	$t=-0.19$ $p=1e+00$ (0.0%)	$t=0.07$ $p=9e-01$ (0.0%)	$t=0.16$ $p=2e-01$ (14.3%)	$t=0.10$ $p=7e-02$ (0.0%)	
TSSF_Var_2_step		$t=0.63$ $p=3e-13$ (71.4%)	$t=-0.06$ $p=6e-01$ (14.3%)	$t=-0.01$ $p=5e-01$ (0.0%)		$t=-0.04$ $p=5e-01$ (0.0%)	TSSF_Var_2_step	$t=0.69$ $p=6e-15$ (57.1%)	$t=-0.05$ $p=8e-01$ (14.3%)	$t=0.07$ $p=8e-02$ (14.3%)	$t=0.21$ $p=9e-03$ (14.3%)	$t=0.08$ $p=9e-03$ (28.6%)	
TSSF_Var_1_step		$t=0.64$ $p=4e-13$ (71.4%)	$t=-0.05$ $p=8e-01$ (0.0%)	$t=0.04$ $p=4e-01$ (0.0%)	$t=0.04$ $p=6e-01$ (0.0%)		TSSF_Var_1_step	$t=0.60$ $p=9e-13$ (57.1%)	$t=-0.36$ $p=1e+00$ (0.0%)	$t=-0.16$ $p=8e-01$ (0.0%)	$t=-0.21$ $p=1e+00$ (0.0%)	$t=0.00$ $p=2e-01$ (0.0%)	
TS_AIRM		$t=0.51$ $p=3e-07$ (71.4%)	$t=-0.23$ $p=1e+00$ (0.0%)	$t=-0.06$ $p=9e-01$ (14.3%)	$t=-0.06$ $p=9e-01$ (14.3%)	$t=-0.08$ $p=8e-01$ (14.3%)	TS_AIRM	$t=0.44$ $p=2e-05$ (57.1%)	$t=-0.20$ $p=1e+00$ (0.0%)	$t=-0.10$ $p=9e-01$ (0.0%)	$t=-0.08$ $p=1e+00$ (14.3%)	$t=-0.00$ $p=8e-01$ (14.3%)	
	CSP	TSSF_Cov_2_step	TSSF_Cov_1_step	TSSF_Var_2_step	TSSF_Var_1_step	TS_AIRM		CSP	TSSF_Cov_2_step	TSSF_Cov_1_step	TSSF_Var_2_step	TSSF_Var_1_step	TS_AIRM

(a) Applying the first two filters

		Number of filters: 12				
		$t=-0.83$ $p=1e+00$ (0.0%)	$t=-0.68$ $p=1e+00$ (0.0%)	$t=-0.75$ $p=1e+00$ (0.0%)	$t=-0.54$ $p=1e+00$ (0.0%)	$t=-0.42$ $p=1e+00$ (0.0%)
CSP						
TSSF_Cov_2_step		$t=0.83$ $p=8e-21$ (85.7%)		$t=0.26$ $p=9e-05$ (28.6%)	$t=0.19$ $p=4e-03$ (28.6%)	$t=0.58$ $p=5e-10$ (71.4%)
TSSF_Cov_1_step		$t=0.68$ $p=1e-13$ (71.4%)	$t=-0.26$ $p=1e+00$ (0.0%)		$t=0.08$ $p=9e-01$ (0.0%)	$t=0.36$ $p=5e-04$ (42.9%)
TSSF_Var_2_step		$t=0.75$ $p=1e-17$ (57.1%)	$t=-0.19$ $p=7e-02$ (14.3%)		$t=0.37$ $p=2e-06$ (42.9%)	$t=0.13$ $p=2e-03$ (28.6%)
TSSF_Var_1_step		$t=0.54$ $p=1e-10$ (57.1%)	$t=-0.58$ $p=1e+00$ (0.0%)	$t=-0.36$ $p=1e+00$ (0.0%)	$t=-0.37$ $p=1e+00$ (0.0%)	$t=0.01$ $p=2e-01$ (0.0%)
TS_AIRM		$t=0.42$ $p=6e-05$ (57.1%)	$t=-0.25$ $p=1e+00$ (0.0%)	$t=-0.12$ $p=1e+00$ (0.0%)	$t=-0.13$ $p=1e+00$ (0.0%)	$t=0.01$ $p=8e-01$ (0.0%)
	CSP	TSSF_Cov_2_step	TSSF_Cov_1_step	TSSF_Var_2_step	TSSF_Var_1_step	TS_AIRM

(c) Applying the first twelve filters

		Number of filters: 12				
		$t=-0.83$ $p=1e+00$ (0.0%)	$t=-0.68$ $p=1e+00$ (0.0%)	$t=-0.75$ $p=1e+00$ (0.0%)	$t=-0.54$ $p=1e+00$ (0.0%)	$t=-0.42$ $p=1e+00$ (0.0%)
CSP						
TSSF_Cov_2_step		$t=0.83$ $p=8e-21$ (85.7%)		$t=0.26$ $p=9e-05$ (28.6%)	$t=0.19$ $p=4e-03$ (28.6%)	$t=0.58$ $p=5e-10$ (71.4%)
TSSF_Cov_1_step		$t=0.68$ $p=1e-13$ (71.4%)	$t=-0.26$ $p=1e+00$ (0.0%)		$t=0.08$ $p=9e-01$ (0.0%)	$t=0.36$ $p=5e-04$ (42.9%)
TSSF_Var_2_step		$t=0.75$ $p=1e-17$ (57.1%)	$t=-0.19$ $p=7e-02$ (14.3%)		$t=0.37$ $p=2e-06$ (42.9%)	$t=0.13$ $p=2e-03$ (28.6%)
TSSF_Var_1_step		$t=0.54$ $p=1e-10$ (57.1%)	$t=-0.58$ $p=1e+00$ (0.0%)	$t=-0.36$ $p=1e+00$ (0.0%)	$t=-0.37$ $p=1e+00$ (0.0%)	$t=0.01$ $p=2e-01$ (0.0%)
TS_AIRM		$t=0.42$ $p=6e-05$ (57.1%)	$t=-0.25$ $p=1e+00$ (0.0%)	$t=-0.12$ $p=1e+00$ (0.0%)	$t=-0.13$ $p=1e+00$ (0.0%)	$t=0.01$ $p=8e-01$ (0.0%)
	CSP	TSSF_Cov_2_step	TSSF_Cov_1_step	TSSF_Var_2_step	TSSF_Var_1_step	TS_AIRM

(b) Applying the first six filters

(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

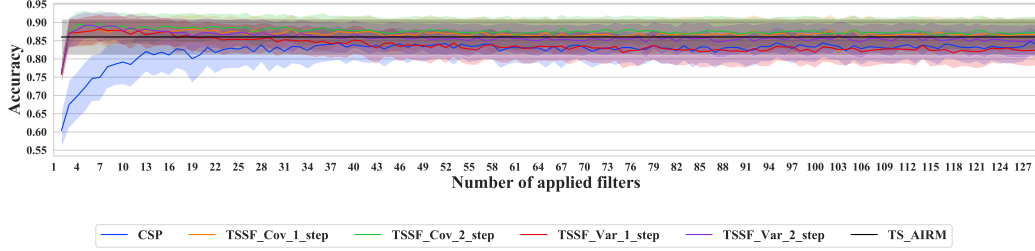


Fig. IV.2: Classification accuracy w.r.t. the number of applied filters within Munich Motor Imagery data set and the accuracies are computed across all subjects. The central line is the mean accuracy and the error band shows 68% confidence interval.

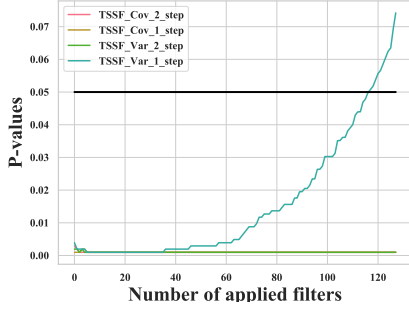


Fig. IV.3: The p-values from the statistical test between all TSSF-based pipelines and CSP w.r.t. the number of applied filters. The chosen data set is *Munich Motor Imagery* and the null-hypothesis is that the median accuracy of TSSF-based methods is not larger than CSP. Significance threshold is set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

TSSF_Var_1_step (One-step)		TS_AIRM (Two-steps)	
Procedure	Computational complexity	Procedure	
Preprocessing			
$\tilde{\mathbf{C}}_{\perp \mathbf{F}}^t = \mathbf{F}_K \mathbf{C}^t (\mathbf{F}_K)^T$, where $\mathbf{F}_K \in \mathbb{R}^{C \times K}$	$O(KC^2)$	$\text{ED}(\mathbf{C}^t)$ $\Downarrow$ $\mathbf{V}^t \mathbf{D}^t (\mathbf{V}^t)^T = \mathbf{C}^t$, where $\mathbf{V}^t \in \mathbb{R}^{C \times C}, \mathbf{D}^t \in \mathbb{R}^{C \times C}$	$O(C^3)$
		$\text{Logm}(\mathbf{D}^t)$	$O(C)$
		$\mathbf{S}^t = \mathbf{V}^t \text{Logm}(\mathbf{D}^t) (\mathbf{V}^t)^T$, where $\mathbf{S}^t \in \mathbb{R}^{C \times C}$	$O(C^3)$
Features			
$f_t = \text{diag}\left(\text{Logm}_1(\tilde{\mathbf{C}}_{\perp \mathbf{F}}^t)\right)$ or $f_t = \log\left(\text{diag}(\tilde{\mathbf{C}}_{\perp \mathbf{F}}^t)\right)$	$O(K^3)$ $O(K)$	$O(\frac{C(C+1)}{2})$	$f_t = \text{vec}(\mathbf{S}^t) \in \mathbb{R}^{\frac{C(C+1)}{2}}$
Feature matrix for all trials			
$[f_1, f_2, \dots, f_T] \in \mathbb{R}^{K \times T}$		$[f_1, f_2, \dots, f_T] \in \mathbb{R}^{\frac{C(C+1)}{2} \times T}$	
Classification			
Linear regression (testing)		Regularized SVM (testing)	
$y_t = \text{diag}(\mathbf{D})[:, K] \times f_t$, where $\text{diag}(\mathbf{D})[:, K] \in \mathbb{R}^K$	$O(K)$	$y_t = \mathbf{w}_{\text{SVM}}^T \times f_t$, where $\mathbf{w}_{\text{SVM}} \in \mathbb{R}^{\frac{C(C+1)}{2} \times 1}$	$O(\frac{C(C+1)}{2})$

Fig. V.1: Theoretical 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

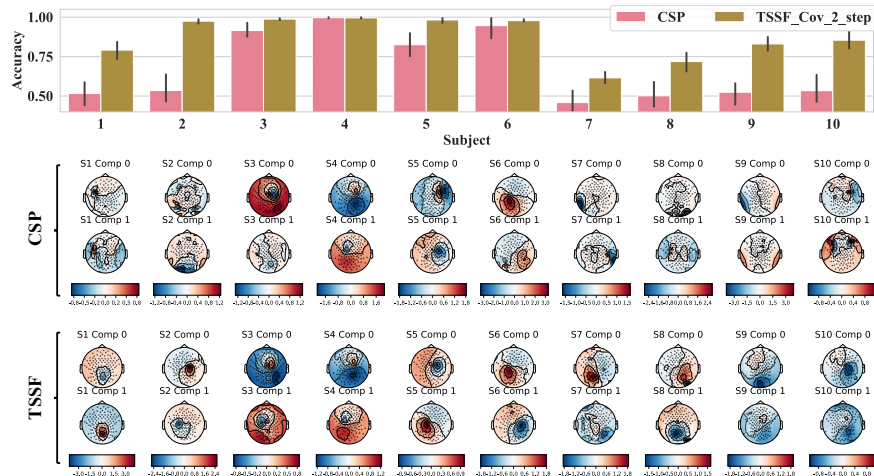


Fig. V.3: The classification accuracy and the associated spatial patterns of data set *Munich Motor Imagery* when applying the first two spatial filters for all subjects.

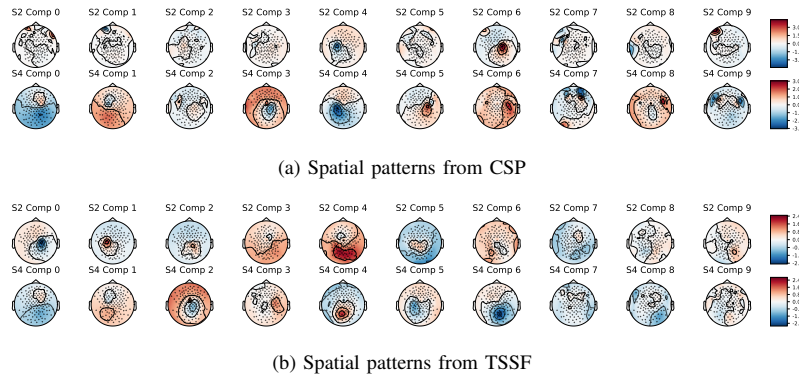


Fig. V.4: The associated spatial patterns of data set *Munich Motor Imagery* when applying the first ten spatial filters for S2 and S4. The major conclusion drawn from these two figures is that the TSSF based patterns present a better ordering comparing to the CSP which means the CSP tends to be affected 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, $\text{diag}(\text{Logm}(\mathbf{F}^T \mathbf{C}^t \mathbf{F}))$ is a good candidate when demanding high accuracy, while $\log(\text{diag}(\mathbf{F}^T \mathbf{C}^t \mathbf{F}))$ 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. Gursay. "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 approach 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.

I. EIGENVECTOR INVARIANCE AFTER LOGARITHM MAPPING

Theorem 1. *The eigenvectors of generalized eigenvalue decomposition (GED) between symmetric positive definite (SPD) matrices keep invariant after logarithm mapping as long as the projecting matrix is not reference matrix, i.e.,:*

$$\begin{aligned} \mathbf{F}, \mathbf{D} &= \text{GED}(\mathbf{C}^w, \mathbf{C}^m) \\ \mathbf{F}, \text{Logm}(\mathbf{D}) &= \text{GED}(\mathbf{S}^w, \mathbf{C}^m), \\ \text{where } \mathbf{S}^w &= \text{Logm}_{\mathbf{C}^m}(\mathbf{C}^w). \end{aligned} \quad (1)$$

Proof. The generalized eigenvalue decomposition (GED) is defined as below:

$$\text{GED}(\mathbf{C}^w, \mathbf{C}^m) \Leftrightarrow \mathbf{C}^w \mathbf{F} = \mathbf{C}^m \mathbf{F} \mathbf{D}, \quad (2)$$

where $\mathbf{F}$ represents the generalized eigenvectors and $\mathbf{D}$ is the generalized eigenvalue matrix. For the convenience, we denote the solutions of this GED problem as:

$$\mathbf{F}, \mathbf{D} = \text{GED}(\mathbf{C}^w, \mathbf{C}^m) \quad (3)$$

Since $\mathbf{C}^m$ is a SPD matrix which means it is invertible, equation (2) can be expressed as below by left multiplying $\mathbf{C}^{-m}$ for both sides:

$$\begin{aligned} \text{GED}(\mathbf{C}^w, \mathbf{C}^m) &\Leftrightarrow \mathbf{C}^{-m} \mathbf{C}^w \mathbf{F} = \mathbf{F} \mathbf{D} \\ &\Leftrightarrow \mathbf{F}, \mathbf{D} = \text{ED}(\mathbf{C}^{-m} \mathbf{C}^w) \end{aligned} \quad (4)$$

By leveraging the SPD property, we can decompose the $\mathbf{C}^{-m}$ into $\mathbf{C}^{-\frac{m}{2}} \mathbf{C}^{-\frac{m}{2}}$, which is as proved below:

$$\begin{aligned} \mathbf{C}^{-m} &= \mathbf{V}^m (\mathbf{D}^m)^{-1} (\mathbf{V}^m)^T \\ &\stackrel{(\mathbf{D}^m)^{-1} > 0}{=} \mathbf{V}^m (\mathbf{D}^m)^{-\frac{1}{2}} \mathbf{I} (\mathbf{D}^m)^{-\frac{1}{2}} (\mathbf{V}^m)^T \\ &\stackrel{(\mathbf{V}^m)^T \mathbf{V}^m = \mathbf{I}}{=} \mathbf{V}^m (\mathbf{D}^m)^{-\frac{1}{2}} (\mathbf{V}^m)^T \mathbf{V}^m (\mathbf{D}^m)^{-\frac{1}{2}} (\mathbf{V}^m)^T \\ &= \mathbf{C}^{-\frac{m}{2}} \mathbf{C}^{-\frac{m}{2}} \end{aligned} \quad (5)$$

Similarly, $\mathbf{C}^{+\frac{m}{2}} \mathbf{C}^{-\frac{m}{2}}$ can be proved as equivalent to identity matrix $\mathbf{I}$. Hence, based on these ingredients derived by SPD property, equation (4) is further written as:

$$\begin{aligned} \mathbf{F}, \mathbf{D} &= \text{ED}(\mathbf{C}^{-m} \mathbf{C}^w) \\ &\Leftrightarrow \mathbf{C}^{-m} \mathbf{C}^w \mathbf{F} = \mathbf{F} \mathbf{D} \\ &\Leftrightarrow \mathbf{C}^{-\frac{m}{2}} \mathbf{C}^{-\frac{m}{2}} \mathbf{C}^w \mathbf{C}^{-\frac{m}{2}} \mathbf{C}^{\frac{m}{2}} \mathbf{F} = \mathbf{C}^{-\frac{m}{2}} \mathbf{C}^{\frac{m}{2}} \mathbf{F} \mathbf{D} \\ &\Leftrightarrow \mathbf{C}^{-\frac{m}{2}} \mathbf{C}^w \mathbf{C}^{-\frac{m}{2}} \mathbf{C}^{\frac{m}{2}} \mathbf{F} = \mathbf{C}^{\frac{m}{2}} \mathbf{F} \mathbf{D} \\ &\Leftrightarrow \mathbf{C}^{-\frac{m}{2}} \mathbf{C}^w \mathbf{C}^{-\frac{m}{2}} \mathbf{V} = \mathbf{V} \mathbf{D}, \text{ where } \mathbf{V} = \mathbf{C}^{\frac{m}{2}} \mathbf{F} \\ &\Leftrightarrow \mathbf{V}, \mathbf{D} = \text{ED}(\mathbf{C}^{-\frac{m}{2}} \mathbf{C}^w \mathbf{C}^{-\frac{m}{2}}) \end{aligned} \quad (6)$$

Note that as $\mathbf{C}^{-m} \mathbf{C}^w$ is not symmetric, its eigenvalue decomposition is not constrained to be orthogonal. Moreover, since $\mathbf{C}^{-\frac{m}{2}} \mathbf{C}^w \mathbf{C}^{-\frac{m}{2}}$ is symmetric, the eigenvectors are orthogonal basis, which can be formulated as below:

$$\begin{aligned} \mathbf{V}^T \mathbf{V} &= \mathbf{I} \\ \mathbf{F}^T \mathbf{F} &\neq \mathbf{I} \\ \mathbf{F}^T \mathbf{C}^m \mathbf{F} &= \mathbf{I} \end{aligned} \quad (7)$$

On the other hand, $\text{ED}(\mathbf{C}^{-\frac{m}{2}} \mathbf{S}^w \mathbf{C}^{-\frac{m}{2}})$ can be solved as:

$$\begin{aligned} &\text{ED}(\mathbf{C}^{-\frac{m}{2}} \mathbf{S}^w \mathbf{C}^{-\frac{m}{2}}) \\ &\Leftrightarrow \text{ED}(\mathbf{C}^{-\frac{m}{2}} [\mathbf{C}^{\frac{m}{2}} \text{Logm}(\mathbf{C}^{-\frac{m}{2}} \mathbf{C}^w \mathbf{C}^{-\frac{m}{2}}) \mathbf{C}^{\frac{m}{2}}] \mathbf{C}^{-\frac{m}{2}}) \\ &\Leftrightarrow \text{ED}(\text{Logm}(\mathbf{C}^{-\frac{m}{2}} \mathbf{C}^w \mathbf{C}^{-\frac{m}{2}})) \\ &\Leftrightarrow \text{ED}(\text{Logm}(\mathbf{V} \mathbf{D} \mathbf{V}^T)) \\ &\Leftrightarrow \text{ED}(\mathbf{V} \text{Logm}(\mathbf{D}) \mathbf{V}^T) \\ &\Rightarrow \mathbf{V}, \text{Logm}(\mathbf{D}) = \text{ED}(\mathbf{V} \text{Logm}(\mathbf{D}) \mathbf{V}^T) \\ &\Leftrightarrow \mathbf{V}, \text{Logm}(\mathbf{D}) = \text{ED}(\mathbf{C}^{-\frac{m}{2}} \mathbf{S}^w \mathbf{C}^{-\frac{m}{2}}) \end{aligned} \quad (8)$$

Based on the results shown in equation (6) and (8), we know:

$$\begin{aligned} \mathbf{V}, \text{Logm}(\mathbf{D}) &= \text{ED}(\mathbf{C}^{-\frac{m}{2}} \mathbf{S}^w \mathbf{C}^{-\frac{m}{2}}) \\ &\Leftrightarrow \mathbf{F}, \text{Logm}(\mathbf{D}) = \text{ED}(\mathbf{C}^{-m} \mathbf{S}^w) \\ &\Leftrightarrow \mathbf{F}, \text{Logm}(\mathbf{D}) = \text{GED}(\mathbf{S}^w, \mathbf{C}^m) \end{aligned} \quad (9)$$

By combining the results of equation (3) and equation (9), we have:

$$\begin{aligned} \mathbf{F}, \mathbf{D} &= \text{GED}(\mathbf{C}^w, \mathbf{C}^m) \\ \mathbf{F}, \text{Logm}(\mathbf{D}) &= \text{GED}(\mathbf{S}^w, \mathbf{C}^m) \end{aligned} \quad (10)$$

Therefore, as a conclusion, the eigenvectors of GED problem keep invariant after logarithm mapping as long as the projecting matrix is not reference matrix. ■

II. DISTRIBUTIVE PROPERTY OF LOGARITHM OPERATION FOR SPD MATRIX

Lemma 2. *The logarithm operation for SPD matrices possesses distributive property as long as the multiplied matrix is orthogonal, i.e.,*

$$\begin{aligned} \text{Logm}(\mathbf{V}^T \mathbf{A} \mathbf{V}) &= \mathbf{V}^T \text{Logm}(\mathbf{A}) \mathbf{V} \\ &, \text{ iff } \mathbf{V}^T \mathbf{V} = \mathbf{I} \text{ and } \mathbf{A} \text{ is SPD.} \end{aligned} \quad (11)$$

Proof. Based on the SPD property of matrix $\mathbf{A}$, we can apply eigenvalue decomposition to $\mathbf{A}$ and obtain:

$$\mathbf{A} = \mathbf{V}^A \mathbf{D}^A (\mathbf{V}^A)^T, \text{ and } (\mathbf{V}^A) (\mathbf{V}^A)^T = \mathbf{I}, \quad (12)$$

where $\mathbf{V}^A$ and $\mathbf{D}^A$ represent the eigenvectors and eigenvalues of SPD matrix $\mathbf{A}$ respectively. Subsequently, combining above results with $\mathbf{V}^T \mathbf{V} = \mathbf{I}$, we have

$$\begin{aligned} \mathbf{V}^T \mathbf{V} &= \mathbf{V}^T (\mathbf{V}^A) (\mathbf{V}^A)^T \mathbf{V} = \mathbf{I} \\ &\Leftrightarrow (\mathbf{V}^T \mathbf{V}^A) (\mathbf{V}^T \mathbf{V}^A)^T = \mathbf{I} \end{aligned} \quad (13)$$

Therefore, by substituting the results from equation (12) and (13) into $\text{Logm}(\mathbf{V}^T \mathbf{A} \mathbf{V})$, we have:

$$\begin{aligned} \text{Logm}(\mathbf{V}^T \mathbf{A} \mathbf{V}) &\stackrel{\text{Eq. (12)}}{=} \text{Logm}(\mathbf{V}^T \mathbf{V}^A \mathbf{D}^A (\mathbf{V}^A)^T \mathbf{V}) \\ &= \text{Logm}((\mathbf{V}^T \mathbf{V}^A) \mathbf{D}^A (\mathbf{V}^T \mathbf{V}^A)^T) \\ &\stackrel{\text{Eq. (13)}}{=} (\mathbf{V}^T \mathbf{V}^A) \text{Logm}(\mathbf{D}^A) (\mathbf{V}^T \mathbf{V}^A)^T \\ &= \mathbf{V}^T [\mathbf{V}^A \text{Logm}(\mathbf{D}^A) (\mathbf{V}^A)^T] \mathbf{V} \\ &\stackrel{\text{Eq. (12)}}{=} \mathbf{V}^T \text{Logm}(\mathbf{V}^A \mathbf{D}^A (\mathbf{V}^A)^T) \mathbf{V} \\ &\stackrel{\text{Eq. (12)}}{=} \mathbf{V}^T \text{Logm}(\mathbf{A}) \mathbf{V} \end{aligned} \quad (14)$$

■

III. EQUIVALENCE BETWEEN TRUE AND APPROXIMATED DECISION FUNCTION

Theorem 3. *The inner product between the weight vector of linear classifier on the tangent space and the tangent vector, i.e., the true decision function on the tangent space referenced on the Fréchet mean of all manifold points, is equivalent to the approximated decision function which is computed from the spatially filtered manifold points via the full rank generalized eigenvectors between the manifold point corresponding to the weight vector of linear classifier on the tangent space and the Fréchet mean, i.e.,*

$$\hat{y}_t \equiv y_t^{\text{approx}}|_{\mathbf{F}}, \text{ iff } \mathbf{F} \text{ is extracted via } \text{GED}(\mathbf{C}^w, \mathbf{C}^m) \text{ and full rank,} \quad (15)$$

where, $\forall t = 1, \dots, T$,

$$\begin{aligned} \hat{y}_t &= \langle \text{Logm}_{\mathbf{C}^m}(\mathbf{C}^w), \text{Logm}_{\mathbf{C}^m}(\mathbf{C}^t) \rangle_{\mathbf{C}^m}, \\ y_t^{\text{approx}}|_{\mathbf{F}} &= \langle \text{Logm}_{\mathbf{F}^T \mathbf{C}^m \mathbf{F}}(\mathbf{F}^T \mathbf{C}^w \mathbf{F}), \text{Logm}_{\mathbf{F}^T \mathbf{C}^m \mathbf{F}}(\mathbf{F}^T \mathbf{C}^t \mathbf{F}) \rangle_{\mathbf{F}^T \mathbf{C}^m \mathbf{F}}, \end{aligned} \quad (16)$$

Proof. For convenience, we first list some properties of linear algebra in the tangent space that will be convenient in the proof. All properties are proved in Section I.

- 1) $\mathbf{F}, \mathbf{D} = \text{GED}(\mathbf{C}^w, \mathbf{C}^m) = \text{ED}(\mathbf{C}^{-m} \mathbf{C}^w)$, where $\mathbf{F}^T \mathbf{F} \neq \mathbf{I}$
- 2) $\mathbf{V}, \mathbf{D} = \text{ED}(\mathbf{C}^{-\frac{m}{2}} \mathbf{C}^w \mathbf{C}^{-\frac{m}{2}})$, where $\mathbf{V}^T \mathbf{V} = \mathbf{I}$
- 3) $\mathbf{V} = \mathbf{C}^{\frac{m}{2}} \mathbf{F} \Rightarrow \mathbf{F}^T \mathbf{C}^m \mathbf{F} = \mathbf{I}$

Since spatial filter $\mathbf{F}$ and the eigenvalue matrix $\mathbf{D}$ are extracted via the $\text{GED}(\mathbf{C}^w, \mathbf{C}^m)$, we have:

$$\begin{aligned} \mathbf{F}^T \mathbf{C}^m \mathbf{F} &= \mathbf{I} \\ \mathbf{F}^T \mathbf{C}^w \mathbf{F} &= \mathbf{D} \end{aligned} \quad (17)$$

On the one hand, when applying this special spatial filter $\mathbf{F}$, it is easy to prove that the approximated decision function can be reformulated as:

$$\begin{aligned} y_t^{\text{approx}}|_{\mathbf{F}} &= \langle \text{Logm}_{\mathbf{F}^T \mathbf{C}^m \mathbf{F}}(\mathbf{F}^T \mathbf{C}^w \mathbf{F}), \text{Logm}_{\mathbf{F}^T \mathbf{C}^m \mathbf{F}}(\mathbf{F}^T \mathbf{C}^t \mathbf{F}) \rangle_{\mathbf{F}^T \mathbf{C}^m \mathbf{F}}, \\ &= \text{Tr}(\text{Logm}(\mathbf{D}) \text{Logm}(\mathbf{F}^T \mathbf{C}^t \mathbf{F})) \end{aligned} \quad (18)$$

On the other hand, based on the previous proved and quoted lemmas as well as the definitions, the true decision function can be expressed as:

$$\begin{aligned} \hat{y}_t &= \langle \text{Logm}_{\mathbf{C}^m}(\mathbf{C}^w), \text{Logm}_{\mathbf{C}^m}(\mathbf{C}^t) \rangle_{\mathbf{C}^m} \\ &= \text{Tr}(\mathbf{C}^{-\frac{m}{2}} (\text{Logm}_{\mathbf{C}^m}(\mathbf{C}^w)) \mathbf{C}^{-\frac{m}{2}} \cdot \mathbf{C}^{-\frac{m}{2}} (\text{Logm}_{\mathbf{C}^m}(\mathbf{C}^t)) \mathbf{C}^{-\frac{m}{2}}) \\ &= \text{Tr}(\mathbf{C}^{-\frac{m}{2}} (\mathbf{C}^{+\frac{m}{2}} \text{Logm}(\mathbf{C}^{-\frac{m}{2}} \mathbf{C}^w \mathbf{C}^{-\frac{m}{2}}) \mathbf{C}^{+\frac{m}{2}}) \mathbf{C}^{-\frac{m}{2}} \cdot \\ &\quad \mathbf{C}^{-\frac{m}{2}} (\mathbf{C}^{+\frac{m}{2}} \text{Logm}(\mathbf{C}^{-\frac{m}{2}} \mathbf{C}^t \mathbf{C}^{-\frac{m}{2}}) \mathbf{C}^{+\frac{m}{2}}) \mathbf{C}^{-\frac{m}{2}}) \\ &= \text{Tr}(\text{Logm}(\mathbf{C}^{-\frac{m}{2}} \mathbf{C}^w \mathbf{C}^{-\frac{m}{2}}) \cdot \text{Logm}(\mathbf{C}^{-\frac{m}{2}} \mathbf{C}^t \mathbf{C}^{-\frac{m}{2}})) \end{aligned} \quad (19)$$

As indicated by point 2) of the list at the beginning of this proof (refer as List 2. in the following context), we know that

$\mathbf{V}, \mathbf{D} = \text{ED}(\mathbf{C}^{-\frac{m}{2}} \mathbf{C}^w \mathbf{C}^{-\frac{m}{2}})$, where $\mathbf{V}^T \mathbf{V} = \mathbf{I}$. Therefore, the results from Eq. (19) can be further derived as:

$$\begin{aligned} \hat{y}_t &= \text{Tr}(\text{Logm}(\mathbf{C}^{-\frac{m}{2}} \mathbf{C}^w \mathbf{C}^{-\frac{m}{2}}) \cdot \text{Logm}(\mathbf{C}^{-\frac{m}{2}} \mathbf{C}^t \mathbf{C}^{-\frac{m}{2}})) \\ &\stackrel{\text{List 2)}}{=} \text{Tr}(\text{Logm}(\mathbf{V} \mathbf{D} \mathbf{V}^T) \cdot \text{Logm}(\mathbf{C}^{-\frac{m}{2}} \mathbf{C}^t \mathbf{C}^{-\frac{m}{2}})) \\ &\stackrel{\text{Lemma 2}}{=} \text{Tr}(\mathbf{V} \text{Logm}(\mathbf{D}) \mathbf{V}^T \cdot \text{Logm}(\mathbf{C}^{-\frac{m}{2}} \mathbf{C}^t \mathbf{C}^{-\frac{m}{2}})) \\ &\stackrel{\text{Trace cyclic invariant}}{=} \text{Tr}(\text{Logm}(\mathbf{D}) \cdot \mathbf{V}^T \text{Logm}(\mathbf{C}^{-\frac{m}{2}} \mathbf{C}^t \mathbf{C}^{-\frac{m}{2}}) \mathbf{V}) \\ &\stackrel{\text{Lemma 2}}{=} \text{Tr}(\text{Logm}(\mathbf{D}) \cdot \text{Logm}(\mathbf{V}^T \mathbf{C}^{-\frac{m}{2}} \mathbf{C}^t \mathbf{C}^{-\frac{m}{2}} \mathbf{V})) \\ &\stackrel{\text{List 3)}}{=} \text{Tr}(\text{Logm}(\mathbf{D}) \cdot \text{Logm}((\mathbf{C}^{+\frac{m}{2}} \mathbf{F})^T \mathbf{C}^{-\frac{m}{2}} \mathbf{C}^t \mathbf{C}^{-\frac{m}{2}} (\mathbf{C}^{+\frac{m}{2}} \mathbf{F}))) \\ &= \text{Tr}(\text{Logm}(\mathbf{D}) \cdot \text{Logm}(\mathbf{F}^T \mathbf{C}^{+\frac{m}{2}} \mathbf{C}^{-\frac{m}{2}} \mathbf{C}^t \mathbf{C}^{-\frac{m}{2}} \mathbf{C}^{+\frac{m}{2}} \mathbf{F})) \\ &= \text{Tr}(\text{Logm}(\mathbf{D}) \cdot \text{Logm}(\mathbf{F}^T \mathbf{C}^t \mathbf{F})) \\ &= y_t^{\text{approx}} \end{aligned} \quad (20)$$

Therefore, as a summary, $\hat{y}_t \equiv y_t^{\text{approx}}$, if and only if when $\mathbf{F}$ is extracted via $\text{GED}(\mathbf{C}^w, \mathbf{C}^m)$ and $\mathbf{F}$ is with full rank. ■

I. CLASSIFICATION ACCURACY W.R.T. THE NUMBER OF APPLIED FILTERS

A. Classification accuracy w.r.t. the number of applied filters for all rest six datasets

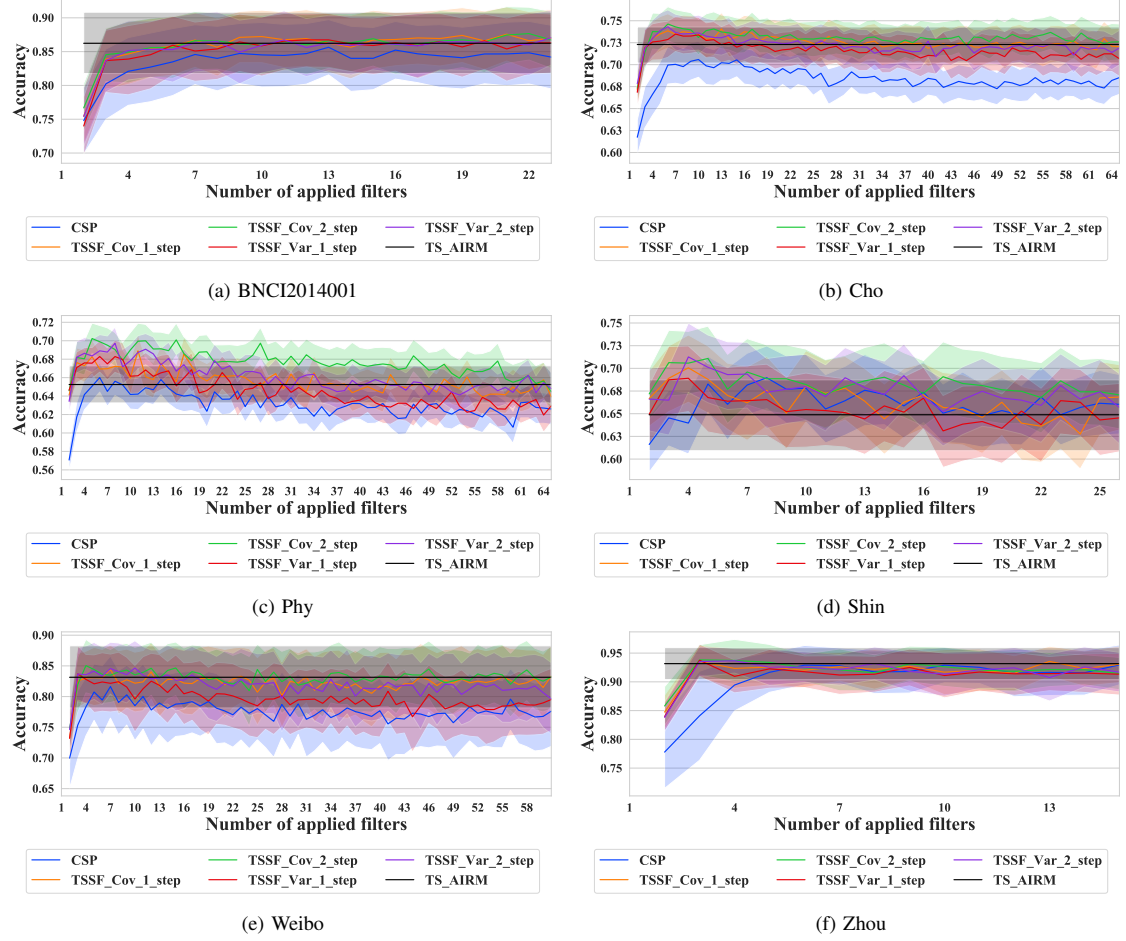


Fig. 1: Classification accuracy w.r.t. the number of applied filters within each data set and the accuracies are computed across all subjects and sessions. The central line is the mean accuracy and the error band shows confidence interval = 68%.

B. Classification accuracy w.r.t. the number of applied filters for all subjects within data set *Munich Motor Imagery*

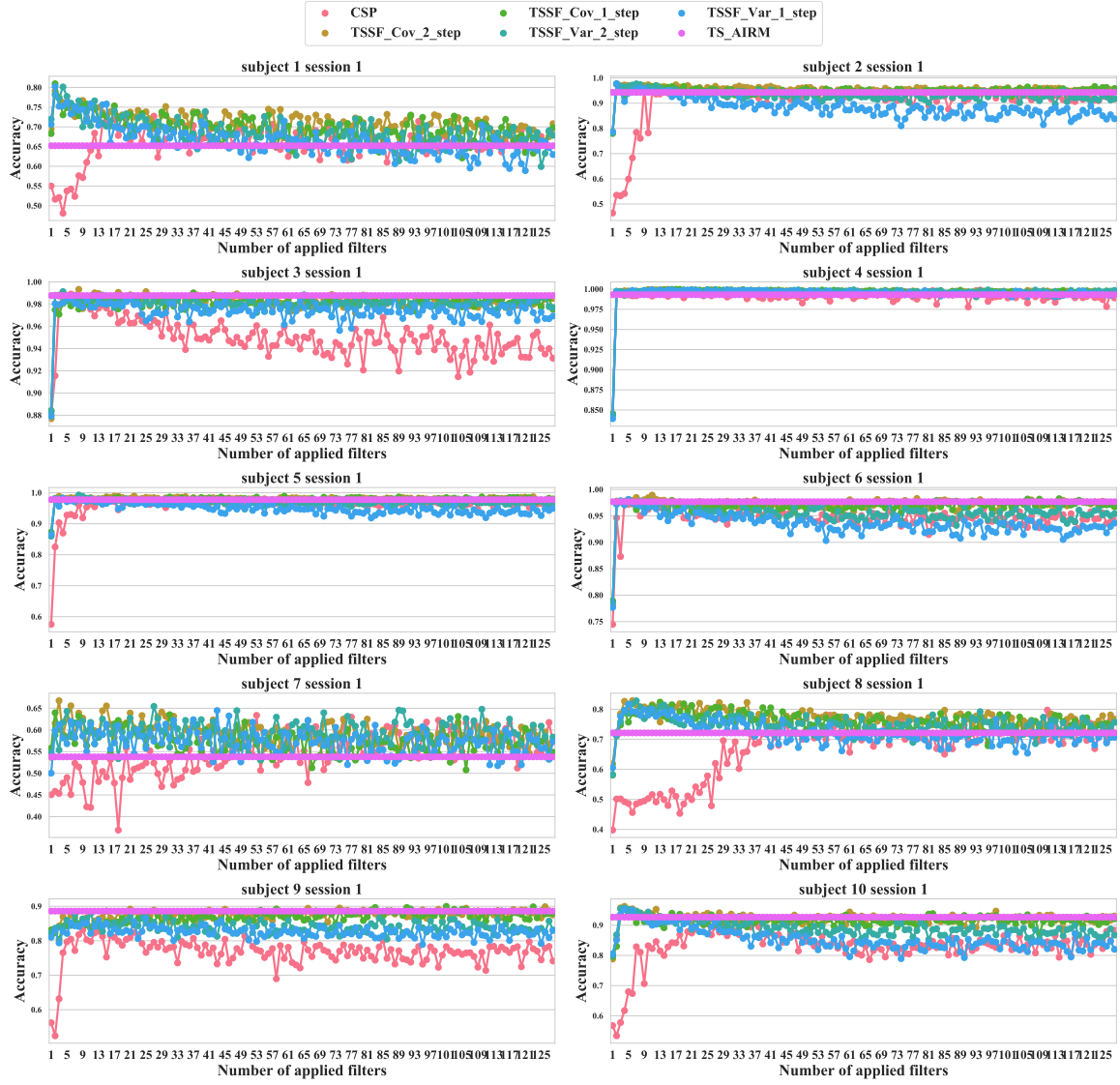


Fig. 2: Classification accuracy w.r.t. the number of applied filters for each subject of data set *Munich Motor Imagery* and the accuracies are computed using ROC-AUC metric and averaged across five folds cross validation.

2.3 Distance Covariance: A Nonlinear Extension of Riemannian Geometry for EEG-based Brain-Computer Interfacing

2.3.1 Synopsis

This study explores the possibility of improving decoding performance by extending the Riemannian framework nonlinearly by substituting the linear Pearson correlation based covariance matrices with the nonlinear distance correlation-based covariance computed from EEG signal. We also examine the theoretical properties of the distance covariance metric in this framework, particularly its invariance to affine transformations. Furthermore, we compare the proposed approach to established Riemannian methods across multiple EEG data sets. Upon the analysis, we briefly discussed the necessity and importance of nonlinearity and preserving affine-invariant properties at last.

Complete bibliographic information

Xu, J., Markham, A., Meunier, A., Raggam, P., and Grosse-Wentrup, M. (2021). Distance covariance: A nonlinear extension of Riemannian geometry for EEG-based brain-computer interfacing. *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 2000-2005). IEEE.

My contribution

1. Devising the idea and plans following the discussion with Markham A.
2. Adapting the optimized distance covariance computation code into Riemannian decoding framework. (Original code of distance covariance computation is provided by Markham A.)
3. Validating the proposed pipelines on multiple motor-magery EEG datasets
4. Paper writing (with helps in Section II.A-II.D)

2.3.2 Publication

Distance Covariance: A Nonlinear Extension of Riemannian Geometry for EEG-based Brain-Computer Interfacing

Jiachen Xu^{1,*}, Alex Markham¹, Anja Meunier¹, Philipp Raggam^{1,2} and Moritz Grosse-Wentrup^{1,3,4}

Abstract—Riemannian frameworks are the basis for some of the best-performing decoding methods in EEG-based Brain-Computer Interfacing. In this work, we consider whether a nonlinear extension of the Riemannian framework, obtained by replacing the channel-wise covariance of the EEG signal with the nonlinear distance covariance, improves decoding performance. We study the theoretical properties of the distance covariance metric in this framework, in particular invariance to affine transformations, and compare the proposed method with established Riemannian methods on three different EEG data sets. We do not find evidence that the distance covariance extension improves decoding performance in comparison to the linear Riemannian framework.

I. INTRODUCTION

Brain-Computer Interfaces (BCIs) decode cognitive states from recordings of brain imaging data, e.g., for communication with severely paralyzed patients [1], [2], to induce neural plasticity in stroke rehabilitation [3], or to passively monitor and react to their users' mental state [4], [5]. For BCIs based on non-invasive recordings of the electroencephalogram (EEG), the decoding pipeline typically consists of three components. In a first step, the EEG channels are linearly combined to attenuate irrelevant EEG sources and retain those that are most informative for a given paradigm and subject [6]–[8]. In a second step, informative features are estimated from the spatially filtered timeseries, e.g., by computing log-bandpowers in canonical EEG frequency bands. In the final decoding step, these features are then processed by a classification algorithm [9].

Among the wide variety of available BCI decoding pipelines, Riemannian methods have evolved as one of the best performing frameworks [10]–[12]. Riemannian frameworks combine the spatial filtering and feature extraction steps of a decoding pipeline by representing each trial of recorded EEG data by a channel-wise covariance matrix. Distances between covariance matrices of different trials are then measured on the Riemannian manifold of symmetric positive definite matrices (SPDs). In this way, the Riemannian frameworks measure similarity between EEG trials not only by the channel-wise power of individual EEG channels but also by the spatial dependence structure between channels. These similarity estimates then form the feature space for the final classification step, for which

standard machine learning algorithms, e.g., support vector machines (SVMs), can be employed.

It has been a long-standing discussion in the BCI community whether linear classifiers are sufficient for EEG decoding, with the general consensus that nonlinear methods may provide small benefits in certain settings [13]. Nonlinear methods, however, have mostly been explored in the final classification step, e.g., by kernelizing SVMs [9]. Spatial filtering and feature extraction algorithms, on the other hand, are predominantly based on linear methods [6], with recent deep learning methods forming a notable exception [14]. To study whether nonlinear extensions of spatial filtering methods may improve decoding performance, we replaced the linear covariance metric in the Riemannian framework with the nonlinear distance covariance metric [15]. The distance covariance is a generalization of the covariance that provides a measure of association similar to the covariance in case of bivariate normal data but is further able to represent nonlinear associations. As such, we hypothesized that an extension of Riemannian frameworks by the distance covariance would perform at least as well as standard, linear Riemannian frameworks while offering the potential to further leverage nonlinear relationships, if they should be present in the data.

After briefly introducing Riemannian frameworks and the distance covariance in Sections II-A and II-B, we present the nonlinear extension of the Riemannian framework in Section II-C. We then study its theoretical properties in Section II-D, where we show that the desirable property of invariance to affine transformations only generalizes in part to the nonlinear case. We then describe in Section II-E how we compare the linear and nonlinear frameworks on three publicly available EEG data sets, and present our results in Section III. We find that the nonlinear Riemannian framework performs on par with the linear one but does not substantially outperform it. We conclude this work by discussing the implications of our findings in Section IV.

II. METHODS

A. Riemannian Methods in BCI

For classification in BCIs, short-time epochs of the EEG signal, also called trials, are usually used to generate the feature space for machine learning algorithms. A trial is represented by the matrix $\mathbf{X}_i \in \mathbb{R}^{N \times T}$, where N denotes the number of channels and T the number of sampled time points. In this context, Riemannian geometry-based methods make use of the covariance matrix $\Sigma_i \in \mathbb{R}^{N \times N}$

¹Research Group Neuroinformatics, Faculty of Computer Science, University of Vienna, Austria

²Department of Neurology & Stroke, Hertie Institute for Clinical Brain Research, University of Tübingen, Germany

³Research Platform Data Science @ Uni Vienna

⁴Vienna Cognitive Science Hub

*Email: jiachen.xu@univie.ac.at

[10], estimated by the sample covariance matrix (SCM)

$$\hat{\Sigma}_i = \frac{1}{T-1} \mathbf{X}_i \mathbf{X}_i^\top. \quad (1)$$

Covariance matrices do not live in Euclidean space but rather in the smooth and differentiable Riemannian manifold $\mathcal{M}$ of Symmetric Positive Definite (SPD) matrices [16].

Trials can be compared with each other by calculating the distance between their SCMs, which are represented by points in the manifold [17]. The distance between two points can be calculated using the affine-invariant Riemannian metric (AIRM), which generalizes the properties of the Euclidean metric in standard vector spaces to the Riemannian manifold [18]. The distance between two covariance matrices (Σ_1, Σ_2) is thus defined as

$$d_{\text{AIRM}}(\Sigma_1, \Sigma_2) = \left\| \log \left(\lambda \left(\Sigma_1^{-\frac{1}{2}} \Sigma_2 \Sigma_1^{-\frac{1}{2}} \right) \right) \right\|_2, \quad (2)$$

where $\lambda(\cdot)$ represents the eigenvalues of its argument and $\|\cdot\|_2$ the L2 norm. This Riemannian distance can be used for classification with the Minimum Distance to Mean (MDM) procedure [10]. For each class $k \in \{1, \dots, K\}$, the Riemannian mean of its I_k intraclass SCMs $(\Sigma_i^k)_{i \in \{1, \dots, I_k\}}$ is calculated by

$$\bar{\Sigma}^k = \arg \min_{\Sigma \in \mathcal{M}} \sum_{i=1}^{I_k} d_{\text{AIRM}}(\Sigma, \Sigma_i^k). \quad (3)$$

For each new trial $\mathbf{X}_j$, the Riemannian distance between the SCM of the trial Σ_j and the Riemannian mean of each class $\bar{\Sigma}^k$ is calculated. The new trial is then assigned to the class with the minimum distance according to

$$\hat{k} = \arg \min_{k \in \{1, \dots, K\}} \left\{ d_{\text{AIRM}}(\Sigma_j, \bar{\Sigma}^k) \right\}. \quad (4)$$

Alternatively, a tangent space mapping can be employed to obtain standard feature vectors and combine Riemannian methods with traditional classification algorithms [19]. The mapping of Σ_j from the Riemannian manifold to the tangent space with Fréchet mean reference $\bar{\Sigma}$ is done via matrix logarithmic mapping:

$$\text{Logm}_{\bar{\Sigma}}(\Sigma_j) = \bar{\Sigma}^{\frac{1}{2}} \log \left(\bar{\Sigma}^{-\frac{1}{2}} \Sigma_j \bar{\Sigma}^{-\frac{1}{2}} \right) \bar{\Sigma}^{\frac{1}{2}}. \quad (5)$$

B. Distance Covariance

The *distance covariance* [15], dCov, can be thought of as a nonlinear extension of the more familiar product-moment covariance, cov, which is only a measure of linear association. Before we define dCov, first recall that the population covariance of random variables X and Y can be given in terms of expected values as

$$\text{cov}(X, Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]. \quad (6)$$

The distance covariance between random variables X and Y is defined as

$$\begin{aligned} d\text{Cov}^2(X, Y) = & \text{cov}(|X - X'|, |Y - Y'|) \\ & - 2\text{cov}(|X - X'|, |Y - Y''|), \end{aligned} \quad (7)$$

where (X', Y') is an independent and identically distributed (iid) copy of (X, Y) , and Y'' is an iid copy of Y . Note that it can be defined more generally, as in [15], however we give a restricted definition here so that it is more obviously analogous to cov and applicable to Riemannian methods. The key intuition here is that it makes use of distances, such as $|X - X'|$: this can be thought of as nonlinearly projecting X and Y into a kernel space so that defining dCov using the linear product-moment covariance (cov) in the kernel space nevertheless results in a nonlinear measure of dependence in the original space—indeed, [20] proves that dCov is equivalent to kernel-based dependence measures such as the Hilbert-Schmidt Independence Criterion (HSIC) [21].

Analogous to (1), the distance covariance matrix, $\Delta_{\mathbf{X}}$, for an N -dimensional random vector can be estimated from the observations $\mathbf{X}_i \in \mathbb{R}^{N \times T}$ of each trial. First, use the sample $\mathbf{X}_i$ to construct the flattened distances matrix F : for each row $\mathbf{x}_i^l = (x_1, \dots, x_n)$ in $\mathbf{X}_i$, the corresponding uncentered distance matrix is defined element-wise to be $U_{p,q} := |x_p - x_q|$; then U must be doubly-centered, $C_{p,q} := U_{p,q} - U_{p,\cdot} - U_{\cdot,q} + U_{\cdot,\cdot}$, where replacing an index p or q with $\cdot$ denotes taking the mean over that index; lastly, form the flattened distances matrix F so that each row F^l is the corresponding vectorized (flattened) doubly-centered matrix C . We can finally define the distance covariance matrix:

$$\hat{\Delta}_{\mathbf{X}} = \frac{1}{N^2} F F^\top. \quad (8)$$

The construction in (8) makes it clear that distance covariance matrix, like the covariance matrix, is the product of a matrix and its transpose and is thus SPD if and only if F is of full row rank. However, as we discuss in Section II-D, the distance covariance matrix does not have the same affine invariance properties of the covariance matrix.

C. Extending Riemannian Decoding with the Distance Covariance

Because the distance covariance gives rise to SPD matrices, we can simply replace the covariance matrices in equations (2) to (5) with the appropriate distance covariance matrices. In practice, we can mitigate non-SPD cases, which are induced by numerical errors, by adding regularization. We thereby obtain a nonlinear generalization of the Riemannian frameworks. The distance covariance provides a measure of association similar to the square of the covariance in case of bivariate normal data but is further able to represent nonlinear associations. In this way, the distcov-Riemannian framework is able to assess nonlinear spatial dependence structure between EEG channels.

It has been argued that invariance under affine transformations is one of the reasons for the superior decoding performance achieved by Riemannian methods [11]. In the following section, we study whether this property also applies to its nonlinear generalization.

D. Invariance Under Affine Transformation

Traditional Riemannian feature extraction based on the covariance matrix has the desirable property of being invariant to affine transformations of the underlying data. This immediately follows from properties of the covariance matrix

$$\Sigma_{A\mathbf{X}+b} = A\Sigma_{\mathbf{X}}A^\top \quad (9)$$

for $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^n$, and the affine invariant Riemannian distance [18]

$$d_{\text{AIRM}}(A\Sigma_1A^\top, A\Sigma_2A^\top) = d_{\text{AIRM}}(\Sigma_1, \Sigma_2) \quad (10)$$

for $A \in \text{GL}_n(\mathbb{R})$, the group of n -dimensional invertible matrices. In the context of BCI, this means that the features are invariant to scaling (e.g. changes in channel connectivity) and remixing of the sources (e.g. shifts in electrode placement). Unfortunately, this property cannot be simply generalized to the nonlinear case, since (9) does not generally hold for the distance covariance matrix. However, weaker invariance properties can be shown:

For $A \in \text{GL}_n(\mathbb{R})$, $b \in \mathbb{R}^n$, we have

$$\begin{aligned} (\Delta_{A\mathbf{X}+b})_{rs} &= d\text{Cov}^2(A_r\mathbf{X} + b_r, A_s\mathbf{X} + b_s) \\ &= \mathbb{E}[|A_r(\mathbf{X} - \mathbf{X}')| \cdot |A_s(\mathbf{X} - \mathbf{X}')|] \\ &\quad + \mathbb{E}[|A_r(\mathbf{X} - \mathbf{X}')|] \cdot \mathbb{E}[|A_s(\mathbf{X} - \mathbf{X}')|] \\ &\quad - 2\mathbb{E}[|A_r(\mathbf{X} - \mathbf{X}')| \cdot |A_s(\mathbf{X} - \mathbf{X}'')|], \end{aligned}$$

where $A_r = (a_{r1}, \dots, a_{rn})$ is the r -th row vector of A . Furthermore, we have

$$|A_r(\mathbf{X} - \mathbf{X}')| = \left| \sum_{j=0}^n a_{rs}(X_s - X'_s) \right| \leq \sum_{j=0}^n |a_{rs}(X_s - X'_s)|. \quad (11)$$

Equality holds if $a_{rs}(X_s - X'_s) \geq 0$ or all $a_{rs}(X_s - X'_s) \leq 0$ for all $s \in \{0, \dots, n\}$. This can only be fulfilled if for all rows A_r of A , $a_{rs} \neq 0$ for only one $s \in \{1, \dots, n\}$. In this case there exists a permutation $\pi : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ and a corresponding permutation matrix P , as well as a diagonal matrix D , such that $A = PD$. Then it follows with

$$a_{rs} = P_r D_s = \begin{cases} d_{\pi(r)\pi(s)} & \text{if } \pi(r) = s \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

and

$$|A_r(\mathbf{X} - \mathbf{X}')| = |d_{\pi(r)\pi(s)}| \cdot |X_{\pi(r)} - X'_{\pi(r)}| \quad (13)$$

that

$$\begin{aligned} (\Delta_{PD\mathbf{X}+b})_{rs} &= |d_{\pi(r)\pi(s)} d_{\pi(s)\pi(s)}| \\ &\quad \cdot \left(\mathbb{E}[|X_{\pi(r)} - X'_{\pi(r)}| \cdot |X_{\pi(s)} - X'_{\pi(s)}|] \right. \\ &\quad + \mathbb{E}[|X_{\pi(r)} - X'_{\pi(r)}|] \cdot \mathbb{E}[|X_{\pi(s)} - X'_{\pi(s)}|] \\ &\quad \left. - 2\mathbb{E}[|X_{\pi(r)} - X'_{\pi(r)}| \cdot |X_{\pi(s)} - X''_{\pi(s)}|] \right) \\ &= |d_{\pi(r)\pi(s)} d_{\pi(s)\pi(s)}| \cdot (\Delta_{PD\mathbf{X}+b})_{\pi(r)\pi(s)} \end{aligned}$$

and therefore

$$\Delta_{PD\mathbf{X}+b} = PD\Delta_{\mathbf{X}}(PD)^\top. \quad (14)$$

We have thus shown that (9) generalizes to the distance covariance matrix for permutations and scaling of the channels, and that therefore the proposed nonlinear Riemannian feature extraction is invariant to such transformations, but not to remixing of the sources.

E. Evaluation Pipeline

After justifying the theoretical legitimacy of nonlinearly extending Riemannian methods by utilizing distance covariance matrices, we validate their practical performance against sample covariance matrices with real-world EEG data. Considering the well-understood neurophysiological mechanism behind the motor imagery (MI) paradigm [22], we adopt the left- versus right-hand motor imagery task as their cortical activities are highly discriminative [23]. Subsequently, we chose three datasets with the different number of electrodes (listed in TABLE I), and the analysis is restricted to the α and β band (8-32 Hz) since they are the most relevant frequency bands to the MI task. For convenience, we base our evaluation setup on the open-source benchmark, Mother of all BCI Benchmark (MOABB) [24]. After epoching the data based on individual task intervals, we estimate the trial-wise distance- and sample covariance matrices according to procedures in Sections II-A and II-B.

TABLE I: Overview of adopted datasets

Dataset Name	#Channels	#Subjects	#Sessions	Citations
BNCI 2014-001	22	9	2	[25]
Cho	64	49	1	[26]
Munich MI	128	10	1	[7]

In the Riemannian framework, we classify the feature matrices via either measuring the geodesic distance between them directly on the manifold, which is known as minimum distance to the mean (MDM) algorithm, or by projecting them to the tangent space and then applying classifiers to the tangent vectors. We adopt both ways to investigate the benefit of bringing in nonlinearity. Further, we choose two types of Riemannian metrics, the affine invariant Riemannian metric (AIRM) and the Log-Euclidean metric (LogE), for the tangent space based classification. The Log-Euclidean metric can also be interpreted as the simplified version of AIRM because the reference point under the LogE metric is an identity matrix rather than the Fréchet mean point. By comparing both, we can explore the influence of affine-invariance.

As for the choice of the classifier on the tangent space, we choose support vector machines (SVM) with both linear and nonlinear kernels, including the polynomial and Gaussian kernel. By comparing their performances, we can study the effect of the nonlinearity introduced by the classifier. There are several hyperparameters to be set in the kernel SVM. For the regularization parameter (L2 penalty) used by all three SVM classifiers, we use a grid search to find the optimal value within the range from 0.01 to 100. For the parameter γ in both nonlinear kernels, we set it to $\frac{1}{l_{\text{ts-vec}} \cdot \text{Var}(X)}$, where $l_{\text{ts-vec}}$ is the length of the tangent vector and $\text{Var}(X)$ is

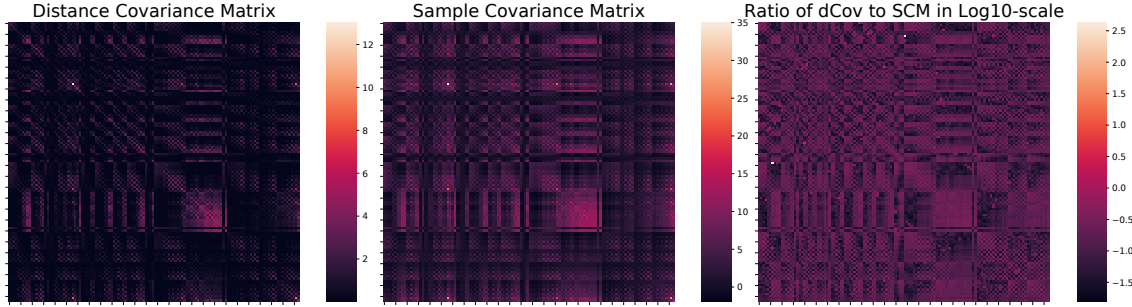


Fig. 1: Matrix structure comparison between distance covariance and sample covariance. These matrices are generated from one randomly selected trial in the Munich data set. The left one shows the structure of the distance covariance matrix computed according to Section II-B, and the middle one is the regular sample covariance matrix. The right one shows the elementwise absolute ratio of the distance covariance matrix (dCov) to sample covariance matrix (SCM) in log10 scale.

the variance of these tangent vectors. Regarding the degree of the polynomial kernel, we fix it to three to achieve a reasonable trade-off between model complexity and the risk of overfitting.

For scoring the predictions made by the classifiers (incl. MDM), we choose the ROC-AUC (receiver operating characteristic-area under the curve) metric. These scores are computed via five-fold cross-validation within each session and subject. To further detect the significant difference among these scores, we adopt the statistical test tools from MOABB [24].

III. RESULTS

Before going into the details of the decoding analysis, we first visualize the matrix structure of both distance and sample covariance matrix, as shown in the left and middle sub-figures in Fig. 1. By comparing these two figures, we can get some intuition about their similarities. These covariance matrices are computed from one randomly selected trial in the Munich data set, which has the largest number of channels. Comparing the left and middle sub-figures, notice that they share a similar blockwise structure, though there does not seem to exist a universal scaling factor between them. To compare the positive-valued distance covariance matrix with the real valued covariance matrix we show the elementwise ratio of their absolute values in the right sub-figure, for better visualization in log10-scale. As seen in this sub-figure, most of the pixels are purple, which means their log-ratios are reasonably close to 0.0. However, we can also find some yellow pixels and black blocks in the same sub-figure, which means the distance covariance values differ substantially from the sample covariance. For instance, the yellow pixel is with a value larger than 2.5, which means their ratio is larger than $10^{2.5} \approx 316$.

The next interesting question is whether these differences can bring additional advantages regarding the feature separability. Therefore, we continue with a classification accuracy comparison between distance and sample covariance matrices as input features in the Riemannian framework. One

proper Riemannian manifold-based method to investigate the raw separability of the input features is the minimum distance to the mean (MDM) classifier because it simply measures and compares the distance between the test point and the mean point of each class. The MDM classifier can be understood as the nearest neighbor algorithm using the AIRM based Riemannian distance instead of the Euclidean distance. As shown in Fig. 2, the classification accuracy of both types of covariance matrices are highly tied with each other, and no significant differences can be found in this comparison. Even if no significance is found, an interesting tendency can be noticed that the difference between the median classification accuracy of both pipelines seems to be negatively correlated with the number of channels, i.e., the size of input features.

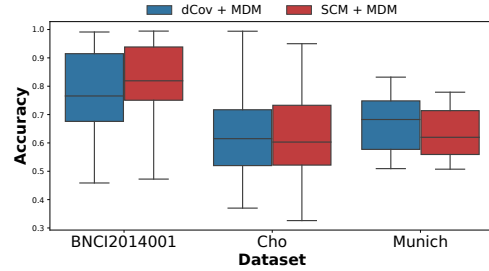


Fig. 2: Within session classification accuracy based on the minimum distance to the mean (MDM) classifier for three motor imagery (MI) datasets with the different number of channels (22, 64, 128 for BNCI2014001, Cho, and Munich, respectively). The blue color indicates the accuracy of using distance covariance matrices (dCov) as feature matrices, and the red color represents the results of using the sample covariance matrices (SCM).

This observation further questions whether the differences between the distance and sample covariance matrices can be better captured by leveraging more advanced classifiers. Therefore, we decide to classify the features on the tangent

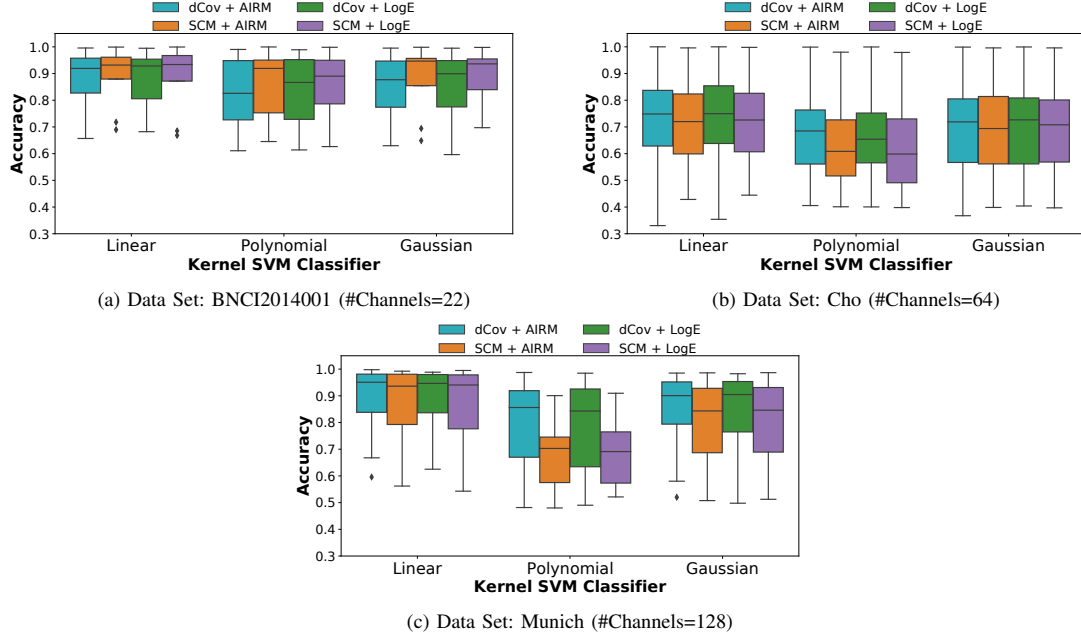


Fig. 3: Within session tangent space based classification accuracy with different kernel SVM classifiers for different datasets. X-axis represent the adopted classifier and color indicate the various combinations of Riemannian metric and feature matrices. E.g., cyan and orange show the accuracy of distance covariance matrices (dCov) and sample covariance matrices (SCM) based on affine-invariant Riemannian metric (AIRM) while yellow and purple represent using Log-Euclidean metric (LogE).

space, which can bring more flexibility to the choices of classifiers. To project both types of matrices from manifold to the tangent space, we choose two types of Riemannian metric, i.e., affine-invariant Riemannian metric (AIRM) and Log-Euclidean (LogE) metric, to explore the influence of preserving the property of affine-invariance. Further, considering the features in the high dimensional space are not necessary to be linearly separable, we choose the kernel support vector machine (SVM) as the classifiers. By using different kernels, we can investigate the influence of the nonlinearity introduced by the classifier. The results are depicted in Fig. 3, in which each sub-figure shows the classification accuracy for each data set and different color represents the accuracy of using various pipelines to classify.

We first look at the influence of different classifiers. Obviously, the classification accuracy within each data set mostly depends on its data quality, and the choice of classifier has a limited effect on the results. For instance, the linear kernel always performs the best, and the Gaussian kernel is a bit superior to the polynomial kernel. However, suppose we only focus on one specific classifier's results and compare the differences between distance and sample covariance matrices (cyan vs. orange and green vs. purple). In that case, the aforementioned negative correlation between median accuracy difference (distance covariance minus sample covariance) and the number of channels still seems to exist. In particular, in the data set Munich, which

has the largest number of channels, an evident classification accuracy difference between distance and sample covariance matrices can be noticed when using a polynomial kernel. Next, by comparing the accuracy between AIRM and LogE based pipelines (cyan vs. green and orange vs. purple), their performances are so competitive that the median accuracy is almost the same.

IV. DISCUSSION

To discuss whether the Riemannian framework can benefit from the nonlinear generalization via the distance covariance, we need to consider three aspects: feature matrix, projection, and classifier. In the comparison in Fig. 2, we see that the nonlinearity does not lead to significant differences in decoding accuracies. One possible justification is that most elements roughly have the same value, which means that most of the information encoded within two types of covariance matrices remains the same. Thus, we cannot find any significant differences in the feature separability.

Second, we notice a negative correlation between the median accuracy difference and the number of channels. One possible explanation is that the total number of the elements with extremely different value has a quadratic growth when the number of channel increase. Furthermore, this change probably further results in the small differences in feature separability. However, the currently available datasets cannot support any significant conclusions about this tendency.

When comparing the results of manifold-based classification (Fig. 2) with tangent space based classification with affine-invariant Riemannian metric (AIRM) (Fig. 3), note that Riemannian methods take advantage of nonlinear projections. Both of them utilize the same Riemannian metric (AIRM) to measure the distance between matrices, while the latter outperforms the former and contains a nonlinear projection.

Next, as shown in Fig. 3, it seems that the classification performance cannot benefit from the nonlinearity of the classifier because the linear one outperforms both nonlinear classifiers regardless of the feature matrices. This finding indicates that the features after nonlinear projection seems to be linearly separable. However, the accuracy difference when using polynomial kernel also suggests the possibility that a linear classifier cannot capture the difference between distance and sample covariance matrices, while nonlinear classifiers seem to be capable of doing it.

Finally, we would like to discuss the importance of affine-invariance from the perspective of feature separability. First, in the Riemannian metric comparison, i.e., AIRM vs. LogE, we can barely find any difference among them. Second, as proved in Section II-D, even if distance covariance matrices only satisfy a weak affine-invariance, it is hard to find any loss of feature separability in comparison to the sample covariance matrices. Therefore, based on both comparisons, it seems that the affine-invariance is not as relevant for decoding performance as previously argued.

ACKNOWLEDGMENT

This project has received funding from the German Federal Ministry of Education and Research (BMBF, grant agreement No 13GW0213A), to P.R.

REFERENCES

- [1] N. Birbaumer, N. Ghanayim, T. Hinterberger, I. Iversen, B. Kotchoubey, A. Kübler, J. Perelmouter, E. Taub, and H. Flor, "A spelling device for the paralysed," *Nature*, vol. 398, no. 6725, pp. 297–298, 1999.
- [2] T. Fomina, G. Lohmann, M. Erb, T. Ethofer, B. Schölkopf, and M. Grosse-Wentrup, "Self-regulation of brain rhythms in the precuneus: A novel BCI paradigm for patients with ALS," *Journal of Neural Engineering*, vol. 13, no. 6, 2016.
- [3] M. Grosse-Wentrup, D. Mattia, and K. Oweiss, "Using brain-computer interfaces to induce neural plasticity and restore function," *Journal of Neural Engineering*, vol. 8, no. 2, p. 025004, 2011.
- [4] T. O. Zander and C. Kothe, "Towards passive brain-computer interfaces: applying brain-computer interface technology to human-machine systems in general," *Journal of Neural Engineering*, vol. 8, no. 2, p. 025005, 2011.
- [5] T. O. Zander, L. R. Krol, N. P. Birbaumer, and K. Gramann, "Neuroadaptive technology enables implicit cursor control based on medial prefrontal cortex activity," *Proceedings of the National Academy of Sciences*, vol. 113, no. 52, pp. 14898–14903, 2016.
- [6] M. Grosse-Wentrup and M. Buss, "Multiclass common spatial patterns and information theoretic feature extraction," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 8, pp. 1991–2000, 2008.
- [7] F. Lotte and C. Guan, "Regularizing common spatial patterns to improve BCI designs: unified theory and new algorithms," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 2, pp. 355–362, 2011.
- [8] M. Grosse-Wentrup, C. Liefhold, K. Gramann, and M. Buss, "Beamforming in noninvasive brain-computer interfaces," *IEEE Transactions on Biomedical Engineering*, vol. 56, pp. 1209–1219, apr 2009.
- [9] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, "A review of classification algorithms for EEG-based brain-computer interfaces," *Journal of Neural Engineering*, vol. 4, no. 2, p. R1, 2007.
- [10] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Multiclass brain-computer interface classification by Riemannian geometry," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 4, pp. 920–928, 2011.
- [11] M. Congedo, A. Barachant, and R. Bhatia, "Riemannian geometry for EEG-based brain-computer interfaces: a primer and a review," *Brain-Computer Interfaces*, vol. 4, no. 3, pp. 155–174, 2017.
- [12] J. Xu, M. Grosse-Wentrup, and V. Jayaram, "Tangent space spatial filters for interpretable and efficient Riemannian classification," *Journal of Neural Engineering*, vol. 17, no. 2, p. 026043, 2020.
- [13] K. Müller, C. W. Anderson, and G. E. Birch, "Linear and nonlinear methods for brain-computer interfaces," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 11, no. 2, pp. 165–169, 2003.
- [14] R. T. Schirrmester, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball, "Deep learning with convolutional neural networks for EEG decoding and visualization," *Human Brain Mapping*, vol. 38, no. 11, pp. 5391–5420, 2017.
- [15] G. J. Székely, M. L. Rizzo, and N. K. Bakirov, "Measuring and testing dependence by correlation of distances," *The Annals of Statistics*, vol. 35, Dec 2007.
- [16] R. Bhatia, *Positive definite matrices*. Princeton university press, 2009.
- [17] W. Förstner and B. Moonen, "A metric for covariance matrices," in *Geodesy-the Challenge of the 3rd Millennium*, pp. 299–309, Springer, 2003.
- [18] X. Pennec, P. Fillard, and N. Ayache, "A Riemannian framework for tensor computing," *International Journal of Computer Vision*, vol. 66, no. 1, pp. 41–66, 2006.
- [19] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Common spatial pattern revisited by Riemannian geometry," in *2010 IEEE International Workshop on Multimedia Signal Processing*, pp. 472–476, IEEE, 2010.
- [20] D. Sejdinovic, B. Sriperumbudur, A. Gretton, and K. Fukumizu, "Equivalence of distance-based and RKHS-based statistics in hypothesis testing," *The Annals of Statistics*, pp. 2263–2291, 2013.
- [21] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with Hilbert-Schmidt norms," in *International Conference on Algorithmic Learning Theory*, pp. 63–77, Springer, 2005.
- [22] G. Pfurtscheller and F. L. Da Silva, "Event-related EEG/MEG synchronization and desynchronization: basic principles," *Clinical Neurophysiology*, vol. 110, no. 11, pp. 1842–1857, 1999.
- [23] G. Pfurtscheller and C. Neuper, "Motor imagery activates primary sensorimotor area in humans," *Neuroscience Letters*, vol. 239, no. 2–3, pp. 65–68, 1997.
- [24] V. Jayaram and A. Barachant, "MOABB: trustworthy algorithm benchmarking for BCIs," *Journal of Neural Engineering*, vol. 15, no. 6, p. 066011, 2018.
- [25] M. Tangermann, K.-R. Müller, A. Aertsen, N. Birbaumer, C. Braun, C. Brunner, R. Leeb, C. Mehring, K. J. Müller, G. Mueller-Putz, et al., "Review of the BCI competition IV," *Frontiers in Neuroscience*, vol. 6, p. 55, 2012.
- [26] H. Cho, M. Ahn, S. Ahn, M. Kwon, and S. C. Jun, "EEG datasets for motor imagery brain-computer interface," *GigaScience*, vol. 6, no. 7, p. gix034, 2017.

2.4 PyTES: A Python Toolbox for Closed-loop Transcranial Electrical Stimulation

2.4.1 Synopsis

To overcome the technical difficulties associated with closed-loop TES studies, we first review the current solutions before introducing a pioneering approach based on PyTES. PyTES is an original Python toolbox that is easy to install and utilize. The user-friendly interface, low cost, and open-source nature of PyTES facilitate the implementation of the closed-loop TES system, even potentially enabling its use for at-home administration. Moreover, we discussed and demonstrated the different strategies for ensuring the operability of the toolbox across multiple operating systems. Lastly, we have briefly touched upon the safety concerns associated with any potential future developments or utilization of this toolbox, particularly with regard to clinical applications.

Complete bibliographic information

Xu, J., and Grosse-Wentrup, M. PyTES: A Python-Toolbox for Closed-Loop Transcranial Electrical Stimulation. *Minor revised at Journal of SoftwareX and preprint version is available at SSRN 4313924.*

My contribution

1. Drafting the technical solutions to circumvent the manual control of stimulation signal
2. Construction of the hardware system with waveform generator
3. Implementation of the PyTES toolbox with Python and SCPI commands
4. Paper writing

2.4.2 Publication

PyTES: A Python toolbox for closed-loop transcranial electrical stimulation

Jiachen Xu^{a,*}, Moritz Grosse-Wentrup^{a,b,c}

^aResearch Group Neuroinformatics, Faculty of Computer Science, University of Vienna, Vienna, Austria

^bResearch Platform Data Science at Uni Vienna, Vienna, Austria

^cVienna Cognitive Science Hub, Vienna, Austria

Abstract

Closed-loop transcranial electrical stimulation (TES) is attracting increasing interest, but limited hardware and software solutions impede wide adoption. In particular, existing solutions are restricted to specific hardware, operating systems, and commercial software packages. We present a Python-based toolbox - PyTES - for closed-loop TES. PyTES is open-source, easy to integrate with existing Python packages for real-time brain decoding, and is able to interface with a wide range of low-cost hardware solutions. In this paper, we review existing solutions for closed-loop TES, introduce the design principles of PyTES, and demonstrate how to realize a closed-loop TES protocol with PyTES.

Keywords: Closed-loop system, transcranial electrical stimulation (TES), personalized system, python toolbox

1. Introduction

Transcranial electric stimulation (TES) is a low-cost, non-invasive, and convenient to use neuromodulatory technique while facing mixed results [1, 2, 3], and limited efficacy [4, 5, 6]. One potential reason that can hamper the overall performance of TES is the conventional fixed-parameter stimulation setup, which cannot cope with the heterogeneity [7, 8] and the dynamics of the brain [9]. Therefore, the adaptive tuning of the stimulation parameters attracts increasing attention, in particular, with behavioral- or neural feedback in a closed-loop system [10, 11].

By definition, a closed-loop system can self-regulate its output to the desired state based on the feedback signal. This type of system possesses stable adaptiveness that is desirable for the TES study. Hence, this has yielded many promising new avenues of closed-loop TES research [10, 12, 13]. Most related studies, however, focus on the neurophysiological mechanism or algorithmic performance, while a general guideline for implementing a closed-loop TES system has not yet been demonstrated. Moreover, most available implementations are closed-source or inconvenient to use or limited to certain operating systems, e.g., Windows, or specific programming languages, e.g., Matlab. These restrictions introduce additional obstacles that hinder the related closed-loop TES studies. Therefore, this paper presents a novel Python-based solution to tackle the difficulties mentioned above based on an original and open-source Python toolbox - PyTES (Code is available at ¹). To establish a closed-loop system, this toolbox can be seamlessly integrated with the hardware, e.g., stimulator or digital-analog-converter, and other Python-based frameworks, e.g., PsychoPy

[14], or OpenVibe [15]. More details about the required hardware for a closed-loop system can be found in Section 2.2. In addition, PyTES also supports other types of TES like high-density TES (multichannel simultaneous stimulation) [16] and envelope transcranial alternating current stimulation (tACS) (irregular/arbitrary signal stimulation) [17], etc.

The remainder of this paper is organized as follows: We first summarize the general guideline for constructing a closed-loop TES system and review existing hardware solutions in Section 2.1. Next, a novel Python-based implementation is introduced in Section 2.2 and more details about the PyTES toolbox are presented in Section 2.3.

2. Methods

2.1. General guideline and current hardware implementations

One significant problem for a closed-loop TES system is the automatic update of the stimulation signal, which is usually manually done in a conventional setup. This means that the stimulation parameters newly computed from the input and feedback signal should automatically update the analog stimulation signal. Therefore, to implement the hardware system, we need at least one software (program) to compute the parameters and one stimulator which can take the digital parameters or signals as input and output the analog stimulation signal. Furthermore, for some stimulators that do not support the input of digital signals or parameters, it is necessary to use a digital-analog converter (DAC) to convert the digital output of the software to the analog input of the stimulator. We hence reviewed the current hardware implementations of the closed-loop TES system and summarized them in Table 1 according to the general guideline stated before.

From Table 1, we can first notice the two dominant preferences about stimulators: the DC-Stimulator Plus (DCSP, Neu-

*Corresponding author

Email address: jiachen.xu@univie.ac.at (Jiachen Xu)

¹<https://github.com/TateXu/pytes>

Reference	Year	Stimulation Type	Stimulator	DAC	Software/Language
[18]	2013	TES	DCSP	NI-DAQ PCI	RTXI (C++)
[19]	2013	TES	DCSP	NI-DAQ PCI-6221	RTXI (C++)
[20]	2016	TES	DCSP	NI-DAQ PCI	RTXI (C++)
[21]	2016	TES	DCSP	N/A	N/A
[22]	2018	TES	DCSP	Cerebus Neural Signal Processor	N/A
[23]	2018	TES	DCSP	Arduino Due board	Matlab
[24]	2019	TES	DCSP	NI-DAQ USB-6343	Matlab
[25]	2020	TES	DCSP	NI-DAQ USB-6343	C++
[26]	2021	TES	DCSP	NI-DAQ USB-6229	Matlab
[27]	2013	TES		StarStim	CoreGUI
[28]	2018	TES		StarStim	CoreGUI
[29]	2018	TES		StarStim	CoreGUI
[30]	2018	TES		StarStim	Matlab
[31]	2021	TES		StarStim	CLM
[32]	2014	TES	Self-designed	N/A	N/A
[33]	2018	TES	Self-designed	N/A	N/A
[34]	2017	TES	Soterix	Powerlab 26T (to trigger stimulator)	
[35]	2019	TES	ANT Neuro	Arduino Due board	Matlab
[36]	2019	TES	NeuroMod16	N/A	N/A

Table 1: Summary of closed-loop system implementation with following abbreviations: RealTime eXperiment Interface - RTXI [37], Closed-Loop Manager - CLM, Not available/applicable - N/A.

roconn, Germany) (9/19) [18, 19, 20, 21, 22, 23, 24, 25, 26] and Starstim (Neuroelectronics, USA) (5/19) [27, 28, 29, 30, 31], which has an integrated DAC. Another five studies either self-design the stimulators [32, 33] or use other stimulators, e.g., Soterix [34], ANT Neuro [35], NeuroMod16 [36]. As for DAC, most DCSP based studies (6/9) use the national instrument data acquisition (NI-DAQ) cards [18, 19, 20, 24, 25, 26]. In addition, C++ [18, 19, 20, 25] and Matlab [23, 24, 26] are the popular language of software or program to compute the updated stimulation parameters.

2.2. An novel open-sourced Python-based implementation

The listed implementations provide diverse ways to control the stimulation parameters, while their usages are restricted in several aspects. First, the DAC-embedded stimulator, i.e., StarStim, requires an additional pro license for activating the remote control function [38] and only supports commercially licensed Matlab. Second, most DCSP-based solutions adopt the NI DAQ card as DAC, which requires tricky software installations, e.g., LabView and compulsory driver. Further, different DAQ cards demand diverse drivers and sometimes are only available on specific operating systems.

Considering these limitations, we decide to design a hardware implementation that should ideally be open-source with minimum installation requirements and compatible across multiple operating systems. Thus, we introduce a Python-based toolbox - PyTES, because Python supports low-level data communication for configuring the hardware and high-level analysis of the feedback signals with the advanced algorithms and displaying the experimental paradigms. Subsequently, to minimize the demand for installing drivers and dedicated software and reduce the cost, we adopt an arbitrary waveform generator

(AWG) as DAC. AWG can stably output analog signals even with arbitrary shapes and can be easily configured by computer commands, e.g., standard commands for programmable instruments (SCPI). Finally, we adopt DCSP as the stimulator because StarStim only supports the Matlab language, which is beyond our open-source scope. One exemplary setup of the Python-based hardware implementation is shown in Figure 1, in which we choose the PyTES (software), an SCPI command based AWG (Rigol DG1062Z, DAC), and the DCSP (stimulator).

To adapt our implementation to a closed-loop system, PyTES can be seamlessly integrated with other open-source Python-based frameworks to control the stimulation parameters, e.g., PsychoPy for presenting experimental paradigms [14] or OpenVibe for processing real-time feedback signal [15]. More details for the integration can be found on the GitHub page. Our implementation also supports a “plug-and-play” feature for the Linux system, i.e., installation-free for drivers and Python packages. In addition to the ease of installation, the presented implementation possesses several more notable features that benefited from PyTES, which are introduced in Section 2.3.

2.3. PyTES - Python based toolbox

One fundamental feature to facilitate deploying a closed-loop TES system is the remote control of the stimulation signal. Moreover, some other special TES techniques require further features. For instance, the high-density tACS requires multi-channel simultaneous stimulation and the unconventional TES like the amplitude-modulated tACS (AM-tACS) [16], and envelope-tACS [17] demands an arbitrary shape of the stimulation signal. Furthermore, current implementations are limited to a specific

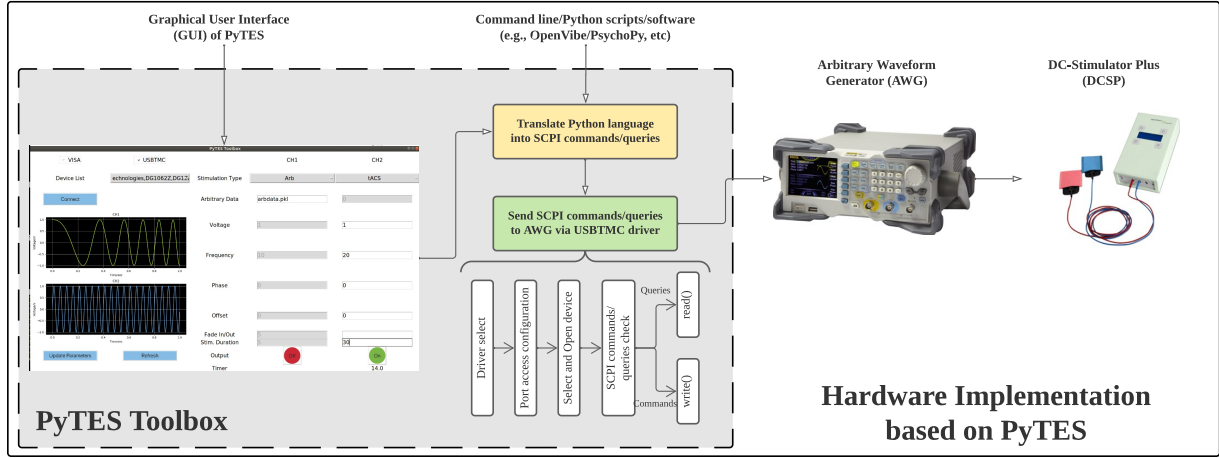


Figure 1: Example of PyTES toolbox based hardware implementation for closed-loop TES and architecture of PyTES toolbox design. From left to right are: Graphical User Interface (GUI) and architecture of PyTES Toolbox (Software), Arbitrary Waveform Generator (Rigol DG1062Z, DAC) and DCSP (Stimulator)

operating system, mainly Windows, and the compatibility with other systems like Linux is less investigated.

To address these requirements, we design our PyTES toolbox with a two-level architecture, as shown in Figure 1. The lower level communicates and configures the hardware with SCPI commands, i.e., a common hardware language. Also, this layer deals with compatibility issues across multiple operating systems by using diverse drivers. On the Linux system, we leverage the built-in driver for Universal Serial Bus Test Measurement and Control (USBTMC) [39] to realize the “plug-and-play” feature. We also adopt the Virtual Instrument Software Architecture (VISA) protocol, which can interface multiple operating systems and devices. However, this VISA protocol requires additional installations, which are elaborated on the GitHub page. Reaping the benefit of the lower level, the higher layer only needs to translate the Python commands into SCPI commands and can be easily called by other Python programs or frameworks without worrying about the encapsulated driver. For example, the intensity of the stimulation signal can be easily adjusted via:

```
from pytes.signal_generator import SignalGenerator
control = SignalGenerator()
control.amp(value=1, chn=1, stim_mode='tACS')
```

For the closed-loop application, this Python command can be inserted into any experimental or online Python frameworks, e.g., PsychoPy or OpenVibe, to automate the real-time update of the stimulation signal. We also designed a graphical user interface (GUI) dedicated to the TES applications to further facilitate the usage. This GUI also covers several other functions, e.g., multichannel simultaneous stimulation, arbitrary stimulation, fade in/out, etc. Importantly, these features can be seamlessly integrated into the closed-loop TES system.

3. Conclusion

To address the technical challenges of the closed-loop TES studies, we first review the current solutions and then introduce a novel implementation based on PyTES. PyTES is an original Python toolbox with effortless installation and usage. The convenience, low cost, and the open-source features of PyTES clear the major obstacles to deploying the closed-loop TES system and even enable the possibility of at-home administration. However, one fundamental while the undiscussed topic is the safety concern. Even though we validate it for the presented setup, as shown in Figure 1, it is still highly suggested to re-evaluate the safety and robustness of their setups as it is unrealistic for us to test all possibilities. Hence, the authors will not take liability for any risk or problem induced by using PyTES in their own setups. As a Python toolbox, the PyTES-based implementations enjoy the abundance and advance of algorithm-related resources. We believe these advantages will further facilitate and promote closed-loop TES studies.

4. Declaration of Conflicting Interest

All authors declare that they have no known competing financial interests or personal relationships that could have appeared to inappropriately influence the work reported in this paper.

References

- [1] D. Brignani, M. Ruzzoli, P. Mauri, C. Miniussi, Is transcranial alternating current stimulation effective in modulating brain oscillations?, *PLoS one* 8 (2) (2013) e56589.
- [2] G. L. Sahlem, B. W. Badran, J. J. Halford, N. R. Williams, J. E. Korte, K. Leslie, M. Strachan, J. L. Breedlove, J. Runion, D. L. Bachman, et al., Oscillating square wave transcranial direct current stimulation (tDCS) delivered during slow wave sleep does not improve declarative memory more than sham: a randomized sham controlled crossover study, *Brain stimulation* 8 (3) (2015) 528–534.

- [3] A. Liu, M. Vöröslakos, G. Kronberg, S. Henin, M. R. Krause, Y. Huang, A. Opitz, A. Mehta, C. C. Pack, B. Krekelberg, et al., Immediate neurophysiological effects of transcranial electrical stimulation, *Nature communications* 9 (1) (2018) 1–12.
- [4] J. Mutz, D. R. Edgcombe, A. R. Brunoni, C. H. Fu, Efficacy and acceptability of non-invasive brain stimulation for the treatment of adult unipolar and bipolar depression: a systematic review and meta-analysis of randomised sham-controlled trials, *Neuroscience & Biobehavioral Reviews* 92 (2018) 291–303.
- [5] J. Mutz, V. Vipulanathan, B. Carter, R. Hurlmann, C. H. Fu, A. H. Young, Comparative efficacy and acceptability of non-surgical brain stimulation for the acute treatment of major depressive episodes in adults: systematic review and network meta-analysis, *The BMJ* 364.
- [6] A. H. Moffa, D. Martin, A. Alonzo, D. Bennabi, D. M. Blumberger, I. M. Benseñor, Z. Daskalakis, F. Fregni, E. Haffen, S. H. Lisanby, et al., Efficacy and acceptability of transcranial direct current stimulation (tDCS) for major depressive disorder: an individual patient data meta-analysis, *Progress in Neuro-Psychopharmacology and Biological Psychiatry* 99 (2020) 109836.
- [7] K. V. Haak, C. F. Beckmann, Understanding brain organisation in the face of functional heterogeneity and functional multiplicity, *NeuroImage* 220 (2020) 117061.
- [8] J. A. Prasad, A. H. Balwani, E. C. Johnson, J. D. Miano, V. Sampathkumar, V. De Andrade, K. Fezzaa, M. Du, R. Vescovi, C. Jacobsen, et al., A three-dimensional thalamocortical dataset for characterizing brain heterogeneity, *Scientific Data* 7 (1) (2020) 1–7.
- [9] P. Mitra, *Observed brain dynamics*, Oxford University Press, 2007.
- [10] A. Karabanov, A. Thielscher, H. R. Siebner, Transcranial brain stimulation: closing the loop between brain and stimulation, *Current opinion in neurology* 29 (4) (2016) 397.
- [11] T. O. Bergmann, Brain state-dependent brain stimulation, *Frontiers in psychology* 9 (2018) 2108.
- [12] G. Thut, T. O. Bergmann, F. Fröhlich, S. R. Soekadar, J.-S. Brittain, A. Valero-Cabré, A. T. Sack, C. Miniussi, A. Antal, H. R. Siebner, et al., Guiding transcranial brain stimulation by EEG/MEG to interact with ongoing brain activity and associated functions: a position paper, *Clinical Neurophysiology* 128 (5) (2017) 843–857.
- [13] F. Fröhlich, L. Townsend, Closed-loop transcranial alternating current stimulation: towards personalized non-invasive brain stimulation for the treatment of psychiatric illnesses, *Current Behavioral Neuroscience Reports* 8 (2) (2021) 51–57.
- [14] J. W. Peirce, PsychoPy—psychophysics software in Python, *Journal of neuroscience methods* 162 (1–2) (2007) 8–13.
- [15] Y. Renard, F. Lotte, G. Gibert, M. Congedo, E. Maby, V. Delannoy, O. Bertrand, A. Lécuyer, Openvibe: An open-source software platform to design, test, and use brain–computer interfaces in real and virtual environments, *Presence* 19 (1) (2010) 35–53.
- [16] E. Negahbani, F. H. Kasten, C. S. Herrmann, F. Fröhlich, Targeting alpha-band oscillations in a cortical model with amplitude-modulated high-frequency transcranial electric stimulation, *Neuroimage* 173 (2018) 3–12.
- [17] A. Wilsch, T. Neuling, J. Obleser, C. S. Herrmann, Transcranial alternating current stimulation with speech envelopes modulates speech comprehension, *Neuroimage* 172 (2018) 766–774.
- [18] J.-S. Brittain, P. Probert-Smith, T. Z. Aziz, P. Brown, Tremor suppression by rhythmic transcranial current stimulation, *Current Biology* 23 (5) (2013) 436–440.
- [19] M. R. Boyle, F. Fröhlich, Eeg feedback-controlled transcranial alternating current stimulation, in: 2013 6th International IEEE/EMBS Conference on Neural Engineering (NER), IEEE, 2013, pp. 140–143.
- [20] C. Lustenberger, M. R. Boyle, S. Alagapan, J. M. Mellin, B. V. Vaughn, F. Fröhlich, Feedback-controlled transcranial alternating current stimulation reveals a functional role of sleep spindles in motor memory consolidation, *Current Biology* 26 (16) (2016) 2127–2136.
- [21] V. Raco, R. Bauer, S. Tharsan, A. Gharabaghi, Combining TMS and tACS for closed-loop phase-dependent modulation of corticospinal excitability: a feasibility study, *Frontiers in cellular neuroscience* 10 (2016) 143.
- [22] A. S. Widge, M. Boggess, A. P. Rockhill, A. Mullen, S. Sheopory, R. Loonis, D. K. Freeman, E. K. Miller, Altering alpha-frequency brain oscillations with rapid analog feedback-driven neurostimulation, *PloS one* 13 (12) (2018) e0207781.
- [23] F. Mansouri, P. Fettes, L. Schulze, P. Giacobbe, J. Zariffa, J. Downar, A real-time phase-locking system for non-invasive brain stimulation, *Frontiers in Neuroscience* (2018) 877.
- [24] R. Lorenz, L. E. Simmons, R. P. Monti, J. L. Arthur, S. Limal, I. Laakso, R. Leech, I. R. Violante, Efficiently searching through large tACS parameter spaces using closed-loop Bayesian optimization, *Brain stimulation* 12 (6) (2019) 1484–1489.
- [25] G. Zarubin, C. Gundlach, V. Nikulin, A. Villringer, M. Bogdan, Transient amplitude modulation of alpha-band oscillations by short-time intermittent closed-loop tACS, *Frontiers in human neuroscience* (2020) 366.
- [26] H. I. Stecher, A. Notbohm, F. H. Kasten, C. S. Herrmann, A comparison of closed loop vs. fixed frequency tACS on modulating brain oscillations and visual detection, *Frontiers in Human Neuroscience* 15 (2021) 272.
- [27] P. Schestatsky, L. Morales-Quezada, F. Fregni, Simultaneous EEG monitoring during transcranial direct current stimulation, *JoVE (Journal of Visualized Experiments)* (76) (2013) e50426.
- [28] N. Ketz, A. P. Jones, N. B. Bryant, V. P. Clark, P. K. Pilly, Closed-loop slow-wave tACS improves sleep-dependent long-term memory generalization by modulating endogenous oscillations, *Journal of Neuroscience* 38 (33) (2018) 7314–7326.
- [29] A. P. Jones, J. Choe, N. B. Bryant, C. S. Robinson, N. A. Ketz, S. W. Skorheim, A. Combs, M. L. Lamphere, B. Robert, H. A. Gill, et al., Dose-dependent effects of closed-loop tACS delivered during slow-wave oscillations on memory consolidation, *Frontiers in neuroscience* (2018) 867.
- [30] C. S. Robinson, N. B. Bryant, J. W. Maxwell, A. P. Jones, B. Robert, M. Lamphere, A. Combs, H. M. Al Azzawi, B. C. Gibson, J. L. Sanguinetti, et al., The benefits of closed-loop transcranial alternating current stimulation on subjective sleep quality, *Brain Sciences* 8 (12) (2018) 204.
- [31] G. Martens, D. Ibáñez-Soria, A. Barra, A. Soria-Frisch, A. Piarulli, O. Gosseries, R. Salvador, A. Rojas, M. A. Nitsche, E. Kroupi, et al., A novel closed-loop EEG-tDCS approach to promote responsiveness of patients in minimally conscious state: A study protocol, *Behavioural Brain Research* 409 (2021) 113311.
- [32] T. Roh, K. Song, H. Cho, D. Shin, H.-J. Yoo, A wearable neuro-feedback system with EEG-based mental status monitoring and transcranial electrical stimulation, *IEEE transactions on biomedical circuits and systems* 8 (6) (2014) 755–764.
- [33] Y. Miao, V. J. Koomson, A CMOS-based bidirectional brain machine interface system with integrated fdNIRS and tDCS for closed-loop brain stimulation, *IEEE Transactions on Biomedical Circuits and Systems* 12 (3) (2018) 554–563.
- [34] J. Leite, L. Morales-Quezada, S. Carvalho, A. Thibaut, D. Doruk, C.-F. Chen, S. C. Schachter, A. Rotenberg, F. Fregni, Surface EEG-transcranial direct current stimulation (tDCS) closed-loop system, *International journal of neural systems* 27 (06) (2017) 1750026.
- [35] F. Mansouri, A. Shanbour, F. Mazza, P. Fettes, J. Zariffa, J. Downar, Effect of theta transcranial alternating current stimulation and phase-locked transcranial pulsed current stimulation on learning and cognitive control, *Frontiers in Neuroscience* (2019) 1181.
- [36] N. Cellini, R. E. Shimizu, P. M. Connolly, D. M. Armstrong, L. T. Hernandez, A. G. Polakiewicz, R. Estrada, M. Aguilar-Simon, M. P. Weisend, S. C. Mednick, et al., Short duration repetitive transcranial electrical stimulation during sleep enhances declarative memory of facts, *Frontiers in human neuroscience* 13 (2019) 123.
- [37] R. J. Lin, J. Bettencourt, J. A. White, D. J. Christini, R. J. Butera, Real-time experiment interface for biological control applications, in: 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology, IEEE, 2010, pp. 4160–4163.
- [38] MatNIC2, <https://www.neuroelectronics.com/solution/software-integrations/matnic2>.
- [39] Source code of USBTMC driver in Linux, <https://github.com/torvalds/linux/blob/master/drivers/usb/class/usbtmc.c>.

3 Discussion

In Section 3.1, we will first revisit the questions raised in Section 1.4. Specifically, we will reflect on the two main research questions, discuss follow-up questions (including hypotheses and research goals), and evaluate how well they have been addressed throughout the study. We will also discuss the implications of the findings and highlight any limitations or gaps in the research that can be addressed in future work.

Next, in Section 3.2, we will introduce some avenues for future research building on the previous discussion. This may include proposing new research questions or hypotheses, identifying areas for further investigation or refinement, or suggesting alternative approaches or methodologies that can be employed to build on the current findings.

Finally, in Section 3.3, we present how we address the research and technical questions and then summarize the dissertation’s key contributions and the implications for the EEG-BCIs field. Subsequently, we discuss any practical or theoretical implications that may arise from the findings. Overall, this discussion section aims to provide a comprehensive and reflective assessment of the research conducted in this dissertation and identify opportunities for future research directions.

3.1 Revisit of the Questions

3.1.1 Research Question 1: Existence of Interpretable and Computationally Efficient Riemannian Methods

To mitigate the computational complexity and the lack of interpretability arising from matrix operations, spatial filtering for dimension reduction appears to be a promising solution. One benefit of using spatial filtering is the reduction of computational complexity by reducing the dimensionality of the input data. Additionally, spatial filtering allows interpretability via visualizing associated spatial patterns, which can describe the spatial distribution and contribution of various brain sources [Haufe et al., 2014]. To extract such a spatial filter, the design and optimization of a target loss function are required, as shown below:

$$\mathbf{W}^* = \arg \min_f \mathcal{L}(\mathbf{W}^T \mathbf{X}, \mathbf{X}), \text{ s.t., } \mathbf{X} \in \mathbb{R} \text{ or } g(x), \quad (3.1)$$

where $\mathcal{L}(\cdot)$, $\mathbf{W}$, and $\mathbf{X}$ denote an abstract loss function, spatial filter, and input data matrix, respectively.

When the geometric structure of $\mathbf{X}$ is known, the constraint can be generalized from the Euclidean space $\mathbb{R}$ to the manifold $\mathcal{M}$, which transforms the problem into a nonlinear dimension reduction problem, also known as manifold learning [Tenenbaum et al., 2000].

In the Riemannian manifold field, various works on Riemannian optimizations have been conducted [Absil et al., 2009, Horev et al., 2016, Harandi et al., 2017], and the dimension reduction method has also been applied to EEG data [Rodrigues et al., 2017]. However, manifold optimization is often considered as complicated, which hence violates our motivation to design an efficient Riemannian method.

A framework for extracting spatial filters based on generalized eigenvalue decomposition is proposed to avoid optimization on the manifold, taking advantage of the equivalence between true and approximated decision functions on the tangent space. By utilizing this equivalence and an approximation of the input feature, the manipulation of manifold points can be avoided, simplifying the time complexity from $O(N^3)$ to $O(N^2)$ and $O(N^2)$ or $O(N)$, respectively. This framework increases the usability of Riemannian methods in EEG decoding tasks, particularly for deployment in real-time BCIs systems, while also providing interpretability to reduce the risk of using artifact sources.

One caveat, however, is that the circumvention of the matrix operation is confined to the testing or inference stage, indicating that the expensive training cost of the current Riemannian methods is still inevitable. Therefore, this poses a further question: how to reduce the training cost, particularly for the non-stationary EEG data?

Training cost in BCIs systems mostly comes from the computational cost of the algorithm and the size of the training data set. Approximating or simplifying algorithms, such as the proposed TSSF framework, can be a viable solution. The Log-Euclidean metric is another well-known approach that simplifies the computation of the Fréchet mean because it can be seen as using an identity matrix as the reference point [Barachant et al., 2013b]. However, reducing the computational cost of Riemannian optimization requires further support, such as selecting Riemannian metric, retraction mapping, algorithms to compute geometric (or Fréchet or Karcher) mean, and so on. It is worth emphasizing that using different metrics or mapping functions may result in losing invariance properties, which can be essential for EEG decoding. Therefore, the necessity of preserving these invariances properties should be carefully reviewed. Below, we provide more details on selected Riemannian optimization methods.

- **Choice of the metrics:** Beyond the distance-inspired Riemannian metrics discussed in Section 1.2.2, it is also worth considering divergence-based metrics such as the family of α -divergences (including Rényi divergence, a generalization of the Kullback-Leibler divergence [Van Erven and Harremoës, 2014]), S-divergence [Sra, 2012], and its generalizations as a parameter family [Chebbi and Moakher, 2012], among others. Notably, S-divergence shares some similarities with Riemannian distance, as discussed in Section 4 of [Sra, 2016].
- **Retraction mapping:** A retraction can be defined as a mapping from $\mathcal{T}_{\mathcal{M}}$ to $\mathcal{M}$, which maps tangent vectors to the manifold, as detailed in Section 4.1 of [Absil et al., 2009]. Apart from the exponential mapping presented in Eq. 1.16, other retractions include $\mathbf{C} = \mathbf{C}_{\text{ref}} + \mathbf{S}$ and $\mathbf{C} = \mathbf{C}_{\text{ref}} + \mathbf{S} + \frac{1}{2}\mathbf{C}_{\text{ref}}\mathbf{S}^{-1}\mathbf{C}_{\text{ref}}$, as explained in Section 4.1.3 of [Jeuris et al., 2012]. A comprehensive review of the available optimization methods, such as line search-based algorithms like steepest descent

and local model-based algorithms like the Riemannian Trust Region [Absil et al., 2007], is presented in [Absil et al., 2009].

- **Geometric mean computation:** Another approach to optimizing the Riemannian decoding pipeline for EEG signals is to focus on more efficient methods for computing the mean point of the SPD manifold. Several algorithms have been proposed, including Riemannian steepest descent [Rentmeesters and Absil, 2011], Riemannian conjugate gradient [Jeuris et al., 2012], Richardson-like iteration [Bini and Iannazzo, 2013], Riemannian Broyden–Fletcher–Goldfarb–Shanno (RBFSGS) methods [Qi et al., 2010] and its extension based on limited memory [Yuan et al., 2016], a power coefficient adaptation [Ho et al., 2013] of the current dominant way to compute the Fréchet mean as used in [Barachant et al., 2010], etc.

Reducing the required training data for a BCIs system is a long-standing problem in the EEG-BCIs community due to the non-stationarity of the EEG data and the time-consuming nature of the training process. While "zero-training" or "subject-independent training" methods were studied in the past [Blankertz et al., 2006a, Krauledat et al., 2008, Fazli et al., 2009], recent attention has turned towards transfer learning [Pan and Yang, 2010], which involves leveraging previously recorded data to enhance the performance of the current task [Jayaram et al., 2016]. Transfer learning has two benefits for EEG signal decoding: first, it allows for a reduction in the training time and data set size, and second, it can address the small data set issue by pooling previously recorded data into a single large data set. Transfer learning has been the subject of intensive research in the past decade, with a recent competition organized in NeurIPS and the results published in [Wei et al., 2022]. A comprehensive review of transfer learning in EEG decoding can be found in [Wu et al., 2020, Wan et al., 2021].

Transfer learning techniques have gained popularity in Riemannian manifold-based EEG decoding due to their shared motivation. Transfer learning can be generally categorized into domain adaptive and weight adaptation [Alamgir et al., 2010, Jayaram et al., 2016]. Domain adaptation involves projecting from one Euclidean space to another Euclidean (sub-)space, which aligns with the idea of dimension reduction on the manifold as presented in Section 1.4. The equivalent operation of projecting vectors on the manifold is known as parallel translation or parallel transport [Absil et al., 2009], where the tangent vector on a Riemannian manifold can be transported along the geodesic to lie on the new tangent space constructed at new points [Cho and Lee, 2017]. This operation only holds when the manifold possesses an affine connection [Absil et al., 2009] and justifies the congruence invariance under affine-invariant Riemannian metric [Congedo et al., 2017]. Notable works in this direction include: transforming covariance matrices of each session/subject and centering them with respect to a reference matrix to enable the inter-session/subject comparisons [Zanini et al., 2017], the domain adaptation among tangent spaces via parallel transport [Maman et al., 2019, Yair et al., 2019], the Riemannian Procrustes analysis (RPA) which can be seen as the extension of covariant shift on the manifold and includes the rescaling and rotation during the alignments [Rodrigues et al., 2018], transformation for the data set with different dimensionalities [Rodrigues et al., 2020] and the extension

of [Rodrigues et al., 2018] which confines all operations on the tangent space [Bleuzé et al., 2022].

Overall, transfer learning is an emerging and promising topic in the EEG-BCI community, providing a way to reduce the long calibration time of BCIs systems. In addition to classification algorithms, another important research direction is the development of feature extraction methods that address undesirable signal properties such as non-stationarity and nonlinearity.

3.1.2 Research Question 2: Necessity of Introducing Extra Nonlinearity and Preserving Affine-Invariance

The neuronal signal is believed to be generated and propagated nonlinearly, as introduced in Section 1.1.5. Currently, popular decoding frameworks, such as deep learning and Riemannian manifold-based decoding, also incorporate nonlinearity to some extent. To better understand the influence and necessity of including nonlinearity in the decoding pipeline, we injected nonlinearity into matrices estimation and comprehensively compared the decoding performance among pipelines with different levels of nonlinearity in [Xu et al., 2021]. The extra nonlinearity injected in the matrix estimation using distance correlation [Székely et al., 2007] did not bring significant benefits, but the pipeline showed more robustness, especially when the EEG data dimensionality was large. When using the nonlinear classifier on the tangent space, the nonlinear matrix-based pipeline also performed slightly better than the traditional linearly estimated covariance matrices, especially when the dimensionality of the EEG data increased. These findings support the hypothesis that nonlinearly estimated covariance matrices might provide additional information, particularly in high-dimensional datasets. This hypothesis is related to the high correlation between neighboring electrodes, as evidenced by the stronger blockwise structure observed in the sample covariance matrix compared to the distance covariance matrix (as shown in Fig. 1 of [Xu et al., 2021]). However, this argument needs validation on more datasets with diverse dimensionalities.

In addition to nonlinearity, the necessity of incorporating affine invariance into the Riemannian decoding pipeline is another question that needs further investigation. While invariances are believed to contribute to the robustness of decoding performance, they come with a computational cost. Thus, relaxing the requirement of strict invariances may reduce computational costs and provide more flexibility in the pipeline. In [Xu et al., 2021], two aspects were considered to tune affine-invariance - matrix estimation and the Riemannian metric. Neither of these aspects significantly degraded decoding performance, suggesting the possibility of loosening affine-invariance in the decoding pipeline. However, it is important to note that operations like parallel transport require the manifold with affine connection [Absil et al., 2009], and discarding affine invariance hence can be risky for such operation.

The potential of matrix estimation as a new aspect to extend the Riemannian decoding pipeline was highlighted by the results of Xu *et al.* [Xu et al., 2021]. While the findings did not reveal significant benefits, they shed light on the importance of incorporating feature design in the pipeline via matrix estimation. Unlike traditional EEG decoding

pipelines that rely on carefully designed features, the Riemannian framework eliminates this step, which can also limit the flexibility of the decoding pipeline. Recent studies have explored using different types of covariance matrices in the Riemannian framework, such as analytic signal covariance matrices [Xu et al., 2019] and autocovariance matrices [Carrara et al., 2022, Carrara and Papadopoulos, 2023]. However, their adaptation to the Riemannian framework requires careful consideration of invariance properties and whether they are symmetric positive definite. Further research is needed to explore the potential of matrix estimation as a new avenue for feature design in the Riemannian decoding pipeline.

3.1.3 Technical Question: Finding an Elegant Solution to Achieve Remote Control of Transcranial Electrical Stimulation Signal

Transcranial electrical stimulation (TES) has shown efficacy in treating certain diseases. However, due to the heterogeneity and non-stationarity of brain signals, the stimulation signal requires manual adjustment occasionally by medical professionals, which can be burdensome for both staff and patients. To address these technical difficulties, we have designed an open-source Python toolbox to facilitate the generation and real-time control of electrical stimulation signals. Our toolbox can be easily integrated into existing BCI experimental platforms such as OpenVibe and Psychopy. It can also serve as feedback in a BCI system, potentially promoting the clinical application of BCIs.

3.2 Future Work

Through presented papers presented in Section 2 and the further discussion in Section 3.1, some exciting research directions are inspired and hinted as shown below:

3.2.1 Neural Manifold - The Neuroscientific Evidence of the Subspace Theory of Neuronal Activity

Subspace theory has inspired various algorithmic applications in the EEG-BCI field, including spatial filtering, dimension reduction, and transfer learning, particularly the domain adaptation type. These algorithms have been successful and are supported by numerous papers. However, it remains to be investigated whether the neuronal activity follows the subspace theory. As presented in previous sections, we have speculated many mathematical justifications about why the Riemannian manifold based decoding framework works so well from the theoretical perspective. If we can find some neuroscientific evidence, our justification for using Riemannian methods will surely be strengthened. Fortunately, there is some exciting progress in the neuroscience community.

In the 1990s, Riehle *et al.* [Riehle and Requin, 1989] discovered a correlation between single-neuron activity and the direction of motor movement. Later, Santhanam *et al.* [Santhanam et al., 2009] applied factor analysis to confine the analysis of neuronal data on a three-dimensional manifold, enabling the description of high-dimensional neuronal

activity by projecting the data onto a low-dimensional space embedded in the high-dimensional array space. This work was followed by many others, such as studies on the global neuronal dynamics of zebrafish [Ahrens et al., 2012], preparation without movement [Kaufman et al., 2014], and dimension reduction for neural recordings [Cunningham and Yu, 2014]. Satler *et al.* subsequently showed that selectively perturbing the neuronal activity mapping inside or outside the low-dimension manifold can significantly affect the performance in motor task learning [Sadler et al., 2014]. A comprehensive review of the related works can be found in [Gallego et al., 2017], where the neural manifold hypothesis is defined. Interestingly, population dynamics are not only observed in motor-related tasks but also in other tasks or functions such as decision-making [Mante et al., 2013], speech [Bouchard and Chang, 2014], and spatial navigation systems [Nieh et al., 2021]. Furthermore, recent work has shown that neural manifold theory can also promote the understanding of biological and artificial neural networks [Chung and Abbott, 2021].

Notably, it seems that Riemannian methods in EEG signal decoding and neural manifold hypothesis share at least one similarity because both approaches aim to infer the neuronal population dynamics (or brain source activity), which is directly linked to behavior. The only difference is that the neural manifold hypothesis provides a bottom-to-up (neuron-to-cortex) perspective, while the Riemannian decoding framework is from the up-to-bottom aspect (scalp-to-cortex). Nevertheless, if we possess prior knowledge about the low-dimensional space through neuroscientific evidence, it can potentially guide the modification of the current Riemannian decoding algorithm by restricting the choice of the Riemannian metric, optimal dimension of the low-dimension manifold, etc. Furthermore, exploring whether Riemannian metrics can be used to analyze micro-electrode array data and modeling neural dynamics will also be exciting research directions.

3.2.2 Manifold based Deep Learning Framework

The deep learning and Riemannian manifold based decoding framework are two typical nonlinear approaches that have demonstrated advantages in decoding EEG data. The former focuses on exploring the nonlinear geometrical structure of the data, while the latter uses a powerful nonlinear classifier to approximate any function supported by the universal approximation theorem. This theorem has been proved from different perspectives, e.g., from the aspect of activation function [Cybenko, 1989] or the network structure [Hornik et al., 1989, Hornik, 1991]. One fundamental assumption shared by these proofs is that the whole network and the data are all defined in the Euclidean space $\mathbb{R}^n$. Recent advancements have generalized these proofs to the Cartan-Hardward manifold [Kratsios and Bilokopytov, 2020], which is a Riemannian manifold for real SPD matrices (see Chapter XII in [Lang, 2012] or [Lim, 2012] for more details). This further means the neural network built upon the SPD manifold can also approximate any mappings. By reviewing the individual success of Riemannian methods and deep learning framework, it naturally yields one follow-up question: whether these two frameworks can be integrated such that we can enjoy the nonlinear feature extraction and nonlinear classifier simultaneously.

Interestingly, the neural network constructed on a manifold was already pioneered by

Huang *et al.* [Huang and Van Gool, 2017] even before the proof of universal approximation theorem on the manifold [Kratsios and Bilokopytov, 2020]. Huang’s network is essentially an extension of the multilayer perceptron, with the adaptation of all Euclidean operations to their manifold equivalents. For example, the linear mapping $\mathbf{y} = \mathbf{w}^T \mathbf{x}$ is replaced with the bilinear mapping $\mathbf{Y} = \mathbf{W}\mathbf{X}\mathbf{W}^T$, and the rectified linear unit (ReLU) is modified from $\mathbf{y} = \max(\mathbf{0}, \mathbf{x})$ to $\mathbf{Y} = \mathbf{U} \max(\mathbf{0}, \mathbf{\Sigma}) \mathbf{U}^T$, $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T$. However, adapting the operations from Euclidean space to the Riemannian manifold poses difficulties in generalizing the neural network because a theoretical justification or proof is required for most operations. Fortunately, some fundamental operations’ generalizations have been justified by different researchers:

- **Linear mapping $\rightarrow$ Bilinear mapping:** A simple adaptation as shown in [Huang and Van Gool, 2017]
- **Rectifying the output $\rightarrow$ Rectifying the eigenvalue of SPD matrices:** A simple adaptation as shown in [Huang and Van Gool, 2017]
- **Back Propagation $\rightarrow$ Riemannian Matrix Backpropagation:** Ionescu et al. [Ionescu et al., 2015a] and its extended version [Ionescu et al., 2015b]. Note that An earlier equivalence is known as the Daleckii-Kreĭn formula [Nielsen and Bhatia, 2013] and the equivalence was proved in [Engin et al., 2018].
- **Batch Normalization (BN) $\rightarrow$ Riemannian Batch Normalization (RBN):** Adapted and proved in [Brooks et al., 2019a]
- **Convolution operation $\rightarrow$ Weighted Fréchet mean estimation:** Adapted and proved in [Chakraborty et al., 2020]
- **Ungated Statistical recurrent unit (SRU) $\rightarrow$ SPD-SRU:** Adapted and proved in [Chakraborty et al., 2018]

Based on the theoretical foundations mentioned earlier, several works have been proposed for analyzing EEG data. For example, Liu *et al.* replaced the LogEig layer in SPDNet [Huang and Van Gool, 2017] with a bag-of-matrix-summarization layer and established their connections [Liu et al., 2019]. Further, Suh *et al.* decomposed the features outputted from SPDNet into several subspaces and learned the common spatial pattern (CSP) from each subspace to boost the performance and ability to interpret the learned model [Suh and Kim, 2021], which coincides with the idea behind filter bank CSP. Moreover, Ju *et al.* directly utilized the stacked feature computing from the filter bank and fed the corresponding covariance matrices into the SPDNet, which is named as Tensor-CSPNet [Ju and Guan, 2022]. Some further works that adopt the idea of using the multi-frequency bands on SPDNet include [Wilson et al., 2022] and [Kobler et al., 2022]. And the latter also proposed a learnable frequency band mapping net (TSMNET) as a domain adaptation framework [Kobler et al., 2022]. Additionally, Wang *et al.* utilized prototype learning framework [Yang et al., 2018] to learn the geometric mean and another two loss functions

after the LogEig layer of SPDNet as the domain adaptation loss function, i.e., adversarial loss and centroid alignment loss as domain adaptation loss [Wang et al., 2021].

All in all, using neural networks based on the SPD manifold for EEG decoding is an emerging research field with many possibilities. In addition to the potential for transfer learning methods as demonstrated in recent works [Wang et al., 2021, Kobler et al., 2022], SPDNet is also compatible with complex-valued input, such as the Hermitian matrices used for analyzing complex radar signals, as shown in [Brooks et al., 2019a] and implemented in [Brooks et al., 2019b]. Another interesting direction for research would be to adapt the current CSP-inspired SPDNet approach into a TSSF version, given that TSSF is a generalization of CSP and is closely linked with the Riemannian manifold. However, this adaptation may pose a risk of overfitting as the extraction of TSSF requires a trained classifier, so additional mathematical work would be needed.

3.2.3 Other Extensions based on the Current Works

In addition to the above two future works that might demand considerable effort, there are also some more straightforward extensions based on the current works.

CSP-Like Extensions for Tangent Space Spatial Filter (TSSF)

As TSSF shares similarities with CSP, it can benefit from many extended algorithms designed for CSP. For instance, the information-theory based multiclass classification can be explored on the TSSF [Grosse-Wentrup and Buss, 2008]. Additionally, regularized CSP [Lotte and Guan, 2010], a classic and effective transfer learning method, can be adapted to TSSF to improve performance with cross-session or cross-subject data. Other CSP variants, such as the invariant CSP (iCSP) [Blankertz et al., 2007] and filter bank CSP (FBCSP) [Ang et al., 2008], can also be explored with TSSF.

Number of Optimal Filters and the Number of Neural Nodes in Neural Manifold

In [Xu et al., 2020], one interesting phenomenon has been briefly reported is the optimal number of spatial filters required to achieve convergence in decoding accuracy seems to be independent of the dimension of EEG signal matrix, as shown in the Appendix of [Xu et al., 2020]. For instance, by observing the results over seven motor-imagery datasets with diverse numbers of electrodes, the optimal number of filters seems to be around six to ten. Coincidentally, in the [Sadtler et al., 2014], Sadtler *et al.* verified the dynamics of the neuronal activity during the motor task learning lies in the 10-dimension manifold embedded in the 85-91 dimension microelectrode array space. The coincidence between the optimal filter number from the Riemannian methods and the number of neural nodes in Sadtler’s studies might hint at an exciting research direction, which is to explore whether such speculation also holds in other experimental paradigms and whether they infer the same source activity.

Quantitatively Analysis of the Influence of Nonlinearity and Invariance Properties

In Section 3.1.2, we highlighted the significance of introducing nonlinearity and invariant properties into the Riemannian decoding pipeline. A subsequent question is whether we can quantitatively measure the impacts of different invariance properties and nonlinearity by adjusting various Riemannian metrics, matrix estimation techniques, or classifiers. Furthermore, these analyses should be conducted on more datasets with varying numbers of channels to ensure generality. By conducting such a meta-analysis, this research direction can potentially provide comprehensive guidance on customizing the choices in the Riemannian decoding pipeline in order to meet the requirement of diverse application scenarios.

Adaptation of PyTES to other Types of Signal Generation

The development of closed-loop systems for non-invasive brain stimulation, particularly in transcranial magnetic stimulation (TMS), has also gained significant attention recently [Kraus et al., 2016, Ding et al., 2022, Tervo et al., 2022]. Extending the current toolbox to control other types of signals, such as magnetic signals, would be an interesting direction to explore, especially given the compatibility of PyTES with various experimental platforms. However, it is crucial to note that using customized software for clinical purposes may raise ethical and legal safety concerns. Therefore, addressing these issues should be a top priority before further development.

3.3 Conclusion

This dissertation aimed to address two key concerns in the development of EEG-BCI technology: accurate EEG signal decoding within the BCI system and the potential extension of BCI applications to other domains. To address these concerns, we thoroughly reviewed background information on EEG-BCI technology, including its historical development, domains of application, and classic and modern EEG signal decoding methods. We then introduced the Riemannian manifold based EEG decoding framework, which is widely regarded as one of the most accurate and robust frameworks in the EEG field. Subsequently, we formulated several concrete research and technical questions to guide our investigation, including: (i). the existence of interpretable and computationally efficient Riemannian methods, (ii). the necessity of introducing extra nonlinearity and preserving affine-invariance, (iii) finding an elegant solution to achieve remote control of transcranial electrical stimulation signal.

To answer these questions: (i) we have presented a novel decoding framework that utilizes spatial filters extracted from a trained classifier on the tangent space by solving generalized eigenvalue problems. Our approach offers an efficient and interpretable alternative to traditional Riemannian decoding frameworks. We have also provided mathematical proof of the equivalence between our proposed framework and the standard Riemannian approach, which ensures its theoretical performance. To validate the practical usability of our method, we evaluated our framework on multiple datasets and found that

3 Discussion

it significantly reduces training time by up to 97% in the testing (inference) stage without sacrificing accuracy. Furthermore, the Riemannian manifold based decoding framework can be visually interpreted by observing the associated spatial patterns of the proposed spatial filter. (ii) we have introduced a novel approach for incorporating nonlinearity and tuning affine-invariance in the Riemannian manifold based EEG decoding pipeline by using the distance covariance matrix (a nonlinear extension of the sample covariance matrix). To support our proposal, we first proved several properties of the distance covariance matrix, including its invariance properties and symmetric positive definiteness. We then evaluated our pipelines using diverse Riemannian metrics and input covariance matrices to adjust the level of nonlinearity and affine-invariance. We analyzed the results across multiple datasets with varying numbers of channels and found that additional nonlinearity did not significantly improve decoding accuracy, but it seemed to impact robustness, particularly in high-dimensional cases. Furthermore, our findings suggest that affine-invariance may not be as crucial as previously suggested. (iii) we have shown that utilizing a waveform generator based hardware solution can conveniently control the stimulation signal. To enhance its accessibility on the computer, we have developed a free and open-source Python toolbox - PyTES, which enables the remote control of the stimulation signal. We have also confirmed the high compatibility of the proposed toolbox with common EEG experimental platforms.

We have hence made key contributions to the EEG-BCI community by (i) providing a computationally efficient and interpretable decoding framework based on the Riemannian manifold, which may promote the online and rehabilitation-oriented applications of Riemannian methods in the EEG-BCIs system, (ii) offering a novel way to explore the influence of nonlinearity and invariance properties in the Riemannian manifold based decoding framework, (iii) providing a free and open-sourced solution to approach the convenient remote control of electrical stimulation signal and the construction of closed-loop TES systems.

In the discussion of this dissertation, several avenues for future research were also hinted at, which mainly include: (i) developing a calibration-free and robust BCIs system using transfer learning algorithms based on the Riemannian manifold, (ii) conducting a quantitative analysis of the impact of nonlinearity and invariance properties on the performance of Riemannian methods to guide the selection of Riemannian metric, mean points, and input covariance matrices, (iii) exploring the existence of a low-dimensional manifold in neuronal activity, and (iv) exploring the application of manifold based neural networks for EEG decoding. Ultimately, we hope this dissertation can serve as a catalyst for advancing the design of accurate EEG decoding algorithms and expanding the range of domains to which BCI systems can be applied. Our vision is to enable individuals to communicate and interact with the world effortlessly, regardless of physical limitations.

Bibliography

- [Abdi and Williams, 2010] Abdi, H. and Williams, L. J. (2010). Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459.
- [Absil et al., 2007] Absil, P.-A., Baker, C. G., and Gallivan, K. A. (2007). Trust-region methods on Riemannian manifolds. *Foundations of Computational Mathematics*, 7:303–330.
- [Absil et al., 2009] Absil, P.-A., Mahony, R., and Sepulchre, R. (2009). Optimization algorithms on matrix manifolds. In *Optimization Algorithms on Matrix Manifolds*. Princeton University Press.
- [Ahrens et al., 2012] Ahrens, M. B., Li, J. M., Orger, M. B., Robson, D. N., Schier, A. F., Engert, F., and Portugues, R. (2012). Brain-wide neuronal dynamics during motor adaptation in zebrafish. *Nature*, 485(7399):471–477.
- [Al-Samhi et al., 2021] Al-Samhi, W., Al-Soswa, M., and Al-Dhabi, Y. (2021). Time series data classification on Grassmann manifold. In *Journal of Physics: Conference Series*, volume 1848, page 012037. IOP Publishing.
- [Alamgir et al., 2010] Alamgir, M., Grosse-Wentrup, M., and Altun, Y. (2010). Multitask learning for Brain-Computer Interfaces. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 17–24. JMLR Workshop and Conference Proceedings.
- [Anderson and Sijercic, 1996] Anderson, C. W. and Sijercic, Z. (1996). Classification of EEG signals from four subjects during five mental tasks. In *Proceedings of the Conference on Engineering Applications in Neural Networks (EANN’96)*, pages 407–414. Turkey.
- [Ang et al., 2008] Ang, K. K., Chin, Z. Y., Zhang, H., and Guan, C. (2008). Filter bank common spatial pattern (FBCSP) in Brain-Computer Interface. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 2390–2397. IEEE.
- [Arjovsky et al., 2016] Arjovsky, M., Shah, A., and Bengio, Y. (2016). Unitary evolution recurrent neural networks. In *ICML’16: Proceedings of the 33rd International Conference on Machine Learning*, pages 1120–1128. PMLR.
- [Arsigny et al., 2005] Arsigny, V., Fillard, P., Pennec, X., and Ayache, N. (2005). *Fast and Simple Computations on Tensors with Log-Euclidean Metrics*. PhD thesis, INRIA.

Bibliography

- [Arsigny et al., 2006] Arsigny, V., Fillard, P., Pennec, X., and Ayache, N. (2006). Log-Euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 56(2):411–421.
- [Bakhshali et al., 2020] Bakhshali, M. A., Khademi, M., Ebrahimi-Moghadam, A., and Moghimi, S. (2020). EEG signal classification of imagined speech based on Riemannian distance of correntropy spectral density. *Biomedical Signal Processing and Control*, 59:101899.
- [Bansal et al., 2018] Bansal, N., Chen, X., and Wang, Z. (2018). Can we gain more from orthogonality regularizations in training deep networks? In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.
- [Barachant et al., 2013a] Barachant, A., Andreev, A., and Congedo, M. (2013a). The Riemannian Potato: An automatic and adaptive artifact detection method for online experiments using Riemannian geometry. In *TOBI Workshop IV*, pages 19–20.
- [Barachant et al., 2010] Barachant, A., Bonnet, S., Congedo, M., and Jutten, C. (2010). Riemannian geometry applied to BCI classification. In *Latent Variable Analysis and Signal Separation*, volume 10, pages 629–636. Springer Berlin Heidelberg.
- [Barachant et al., 2011] Barachant, A., Bonnet, S., Congedo, M., and Jutten, C. (2011). Multiclass Brain-Computer Interface classification by Riemannian geometry. *IEEE Transactions on Biomedical Engineering*, 59(4):920–928.
- [Barachant et al., 2013b] Barachant, A., Bonnet, S., Congedo, M., and Jutten, C. (2013b). Classification of covariance matrices using a Riemannian-based kernel for BCI applications. *Neurocomputing*, 112:172–178.
- [Basser and Pierpaoli, 2011] Basser, P. J. and Pierpaoli, C. (2011). Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor MRI. *Journal of Magnetic Resonance*, 213(2):560–570.
- [Berger, 1929] Berger, H. (1929). Über das Elektroenkephalogramm des Menschen. *Archiv für Psychiatrie und Nervenkrankheiten*, 87(1):527–570.
- [Bhatia, 2009] Bhatia, R. (2009). Positive definite matrices. In *Positive Definite Matrices*. Princeton University Press.
- [Bini and Iannazzo, 2013] Bini, D. A. and Iannazzo, B. (2013). Computing the Karcher mean of symmetric positive definite matrices. *Linear Algebra and its Applications*, 438(4):1700–1710.
- [Blankertz et al., 2006a] Blankertz, B., Dornhege, G., Krauledat, M., Müller, K.-R., Kunzmann, V., Losch, F., and Curio, G. (2006a). The Berlin Brain-Computer Interface:

- EEG-based communication without subject training. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2):147–152.
- [Blankertz et al., 2007] Blankertz, B., Kawanabe, M., Tomioka, R., Hohlefeld, F., Müller, K.-R., and Nikulin, V. (2007). Invariant Common Spatial Patterns: Alleviating Nonstationarities in Brain-Computer Interfacing. In Platt, J., Koller, D., Singer, Y., and Roweis, S., editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc.
- [Blankertz et al., 2004] Blankertz, B., Müller, K.-R., Curio, G., Vaughan, T., Schalk, G., Wolpaw, J., Schlogl, A., Neuper, C., Pfurtscheller, G., Hinterberger, T., Schroder, M., and Birbaumer, N. (2004). The BCI competition 2003: Progress and perspectives in detection and discrimination of EEG single trials. *IEEE Transactions on Biomedical Engineering*, 51(6):1044–1051.
- [Blankertz et al., 2006b] Blankertz, B., Müller, K.-R., Krusienski, D. J., Schalk, G., Wolpaw, J. R., Schlogl, A., Pfurtscheller, G., Millan, J. d. R., Schroder, M., and Birbaumer, N. (2006b). The BCI competition III: Validating alternative approaches to actual BCI problems. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2):153–159.
- [Bleuzé et al., 2022] Bleuzé, A., Mattout, J., and Congedo, M. (2022). Tangent space alignment: Transfer learning for Brain-Computer Interface. *Frontiers in Human Neuroscience*, pages 16–1049985.
- [Boothby, 2003] Boothby, W. M. (2003). *An introduction to differentiable manifolds and Riemannian geometry*, volume 120. Gulf Professional Publishing.
- [Borra et al., 2020] Borra, D., Fantozzi, S., and Magosso, E. (2020). Interpretable and lightweight convolutional neural network for EEG decoding: Application to movement execution and imagination. *Neural Networks*, 129:55–74.
- [Bouchard and Chang, 2014] Bouchard, K. E. and Chang, E. F. (2014). Control of spoken vowel acoustics and the influence of phonetic context in human speech sensorimotor cortex. *Journal of Neuroscience*, 34(38):12662–12677.
- [Bozinovski et al., 1988] Bozinovski, S., Sestakov, M., and Bozinovska, L. (1988). Using EEG alpha rhythm to control a mobile robot. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1515–1516. IEEE.
- [Brazier, 1961] Brazier, M. A. B. (1961). A history of the electrical activity of the brain: The first half-century. *Archives of Internal Medicine*, 110(6):932–934.
- [Brazier, 1984] Brazier, M. A. B. (1984). Pioneers in the discovery of evoked potentials. *Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section*, 59(1):2–8.

Bibliography

- [Brooks et al., 2019a] Brooks, D., Schwander, O., Barbaresco, F., Schneider, J.-Y., and Cord, M. (2019a). Riemannian batch normalization for SPD neural networks. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- [Brooks et al., 2019b] Brooks, D., Schwander, O., Barbaresco, F., Schneider, J.-Y., and Cord, M. (2019b). Second-order networks in PyTorch. In *Geometric Science of Information: 4th International Conference, GSI 2019, Toulouse, France, August 27–29, 2019, Proceedings 4*, pages 751–758. Springer.
- [Cardenas-Rojas et al., 2020] Cardenas-Rojas, A., Pacheco-Barrios, K., Giannoni-Luza, S., Rivera-Torrejon, O., and Fregni, F. (2020). Noninvasive brain stimulation combined with exercise in chronic pain: A systematic review and meta-analysis. *Expert Review of Neurotherapeutics*, 20(4):401–412.
- [Carrara and Papadopoulou, 2023] Carrara, I. and Papadopoulou, T. (2023). Classification of BCI-EEG based on augmented covariance matrix. *arXiv preprint arXiv:2302.04508*.
- [Carrara et al., 2022] Carrara, I. I., Senellart, S. A., and Papadopoulou, T. (2022). On the use of the "Augmented" autocovariance matrix for the classification of BCI-EEG. In *Proceedings of Cortico Days*.
- [Caton, 1875] Caton, R. (1875). Electrical currents of the brain. *The Chicago Journal of Nervous and Mental Disease*, 2(4):610.
- [Chakraborty et al., 2017a] Chakraborty, R., Banerjee, M., and Vemuri, B. C. (2017a). Statistics on the space of trajectories for longitudinal data analysis. In *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pages 999–1002. IEEE.
- [Chakraborty et al., 2020] Chakraborty, R., Bouza, J., Manton, J., and Vemuri, B. C. (2020). ManifoldNet: A deep neural network for manifold-valued data with applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- [Chakraborty et al., 2018] Chakraborty, R., Yang, C.-H., Zhen, X., Banerjee, M., Archer, D., Vaillancourt, D., Singh, V., and Vemuri, B. (2018). A statistical recurrent model on the manifold of symmetric positive definite matrices. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.
- [Chakraborty et al., 2017b] Chakraborty, S., Tomsett, R., Raghavendra, R., Harborne, D., Alzantot, M., Cerutti, F., Srivastava, M., Preece, A., Julier, S., Rao, R. M., et al. (2017b). Interpretability of deep learning models: A survey of results. In *2017 IEEE Smartworld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (Smartworld/SCALCOM/UIC/ATC/CBDcom/IOP/SCI)*, pages 1–6. IEEE.

- [Chapman and Bragdon, 1964] Chapman, R. M. and Bragdon, H. R. (1964). Evoked responses to numerical and non-numerical visual stimuli while problem solving. *Nature*, 203(4950):1155–1157.
- [Chebbi and Moakher, 2012] Chebbi, Z. and Moakher, M. (2012). Means of Hermitian positive-definite matrices based on the log-determinant α -divergence function. *Linear Algebra and its Applications*, 436(7):1872–1889.
- [Chen et al., 2010] Chen, Y., Wiesel, A., Eldar, Y. C., and Hero, A. O. (2010). Shrinkage algorithms for MMSE covariance estimation. *IEEE Transactions on Signal Processing*, 58(10):5016–5029.
- [Chevallier et al., 2017] Chevallier, E., Kalunga, E., and Angulo, J. (2017). Kernel density estimation on spaces of Gaussian distributions and symmetric positive definite matrices. *SIAM Journal on Imaging Sciences*, 10(1):191–215.
- [Chevallier et al., 2014] Chevallier, S., Barthélemy, Q., and Atif, J. (2014). Subspace metrics for multivariate dictionaries and application to EEG. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7178–7182. IEEE.
- [Chiumiento and Melgaard, 2012] Chiumiento, E. and Melgaard, M. (2012). Stiefel and Grassmann manifolds in quantum chemistry. *Journal of Geometry and Physics*, 62(8):1866–1881.
- [Cho and Lee, 2017] Cho, M. and Lee, J. (2017). Riemannian approach to batch normalization. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- [Chung and Abbott, 2021] Chung, S. Y. and Abbott, L. F. (2021). Neural population geometry: An approach for understanding biological and artificial neural networks. *Current Opinion in Neurobiology*, 70:137–144.
- [Clark et al., 1961] Clark, W. A., Brown, R. M., Goildstein, M. H., Molnar, C. E., O’Brien, D. F., and Zieman, H. E. (1961). The average response computer (ARC): A digital device for computing averages and amplitude and time histograms of electrophysiological response. *IRE Transactions on Bio-Medical Electronics*, 8(1):46–51.
- [Cohen, 2014] Cohen, M. X. (2014). *Analyzing neural time series data: Theory and practice*. MIT press.
- [Congedo et al., 2017] Congedo, M., Barachant, A., and Bhatia, R. (2017). Riemannian geometry for EEG-based brain-computer interfaces: A primer and a review. *Brain-Computer Interfaces*, 4(3):155–174.

Bibliography

- [Congedo and Jain, 2019] Congedo, M. and Jain, S. (2019). A Julia package for manipulating Brain-Computer Interface data in the manifold of positive definite matrices. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 211–216. IEEE.
- [Congedo et al., 2019] Congedo, M., Rodrigues, P. L. C., and Jutten, C. (2019). The Riemannian minimum distance to means field classifier. In *Proceedings of the 8th Graz Brain Computer Interface Conference 2019, Bridging Science and Application*, pages 6–10. Graz University of Technology, Austria.
- [Cunningham and Yu, 2014] Cunningham, J. P. and Yu, B. M. (2014). Dimensionality reduction for large-scale neural recordings. *Nature Neuroscience*, 17(11):1500–1509.
- [Cybenko, 1989] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314.
- [Dähne et al., 2014] Dähne, S., Meinecke, F. C., Haufe, S., Höhne, J., Tangermann, M., Müller, K.-R., and Nikulin, V. V. (2014). SPoC: a novel framework for relating the amplitude of neuronal oscillations to behaviorally relevant parameters. *NeuroImage*, 86:111–122.
- [De Clercq et al., 2006] De Clercq, W., Vergult, A., Vanrumste, B., Van Paesschen, W., and Van Huffel, S. (2006). Canonical correlation analysis applied to remove muscle artifacts from the electroencephalogram. *IEEE Transactions on Biomedical Engineering*, 53(12):2583–2587.
- [Delgado, 1952] Delgado, J. M. (1952). Permanent implantation of multilead electrodes in the brain. *The Yale Journal of Biology and Medicine*, 24(5):351.
- [Di Pino et al., 2014] Di Pino, G., Pellegrino, G., Assenza, G., Capone, F., Ferreri, F., Formica, D., Ranieri, F., Tombini, M., Ziemann, U., Rothwell, J. C., et al. (2014). Modulation of brain plasticity in stroke: A novel model for neurorehabilitation. *Nature Reviews Neurology*, 10(10):597–608.
- [Ding et al., 2022] Ding, Z., Wang, Y., Li, J., and Li, X. (2022). Closed-loop TMS-EEG reactivity with occipital alpha-phase synchronized. *Journal of Neural Engineering*, 19(5):056027.
- [Du et al., 2019] Du, M., Liu, N., and Hu, X. (2019). Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77.
- [Dunlop et al., 2017] Dunlop, K., Hanlon, C. A., and Downar, J. (2017). Noninvasive brain stimulation treatments for addiction and major depression. *Annals of the New York Academy of Sciences*, 1394(1):31–54.
- [Engin et al., 2018] Engin, M., Wang, L., Zhou, L., and Liu, X. (2018). DeepKSPD: Learning Kernel-matrix-based SPD representation for fine-grained image recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 612–627.

- [Estrin, 1965] Estrin, T. (1965). On-line electroencephalographic digital computing system. *Electroencephalography and Clinical Neurophysiology*, 19(5):524–526.
- [Farwell and Donchin, 1988] Farwell, L. A. and Donchin, E. (1988). Talking off the top of your head: Toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology*, 70(6):510–523.
- [Fazli et al., 2009] Fazli, S., Grozea, C., Danoczy, M., Blankertz, B., Popescu, F., and Müller, K.-R. (2009). Subject independent EEG-based BCI decoding. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C., and Culotta, A., editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc.
- [Fertonani and Miniussi, 2017] Fertonani, A. and Miniussi, C. (2017). Transcranial electrical stimulation: What we know and do not know about mechanisms. *The Neuroscientist*, 23(2):109–123.
- [Fetz, 1969] Fetz, E. E. (1969). Operant conditioning of cortical unit activity. *Science*, 163(3870):955–958.
- [Fletcher and Joshi, 2004] Fletcher, P. T. and Joshi, S. (2004). Principal geodesic analysis on symmetric spaces: Statistics of diffusion tensors. In *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis: ECCV 2004 Workshops CVAMIA and MMBIA, Prague, Czech Republic, May 15, 2004, Revised Selected Papers*, pages 87–98. Springer.
- [Förstner and Moonen, 2003] Förstner, W. and Moonen, B. (2003). A metric for covariance matrices. *Geodesy-the Challenge of the 3rd Millennium*, pages 299–309.
- [Gallego et al., 2017] Gallego, J. A., Perich, M. G., Miller, L. E., and Solla, S. A. (2017). Neural manifolds for the control of movement. *Neuron*, 94(5):978–984.
- [Gibbs et al., 1935] Gibbs, F. A., Davis, H., and Lennox, W. G. (1935). The electroencephalogram in epilepsy and in conditions of impaired consciousness. *Archives of Neurology & Psychiatry*, 34(6):1133–1148.
- [Grosse-Wentrup and Buss, 2008] Grosse-Wentrup, M. and Buss, M. (2008). Multiclass common spatial patterns and information theoretic feature extraction. *IEEE Transactions on Biomedical Engineering*, 55(8):1991–2000.
- [Grosse-Wentrup et al., 2009] Grosse-Wentrup, M., Liefhold, C., Gramann, K., and Buss, M. (2009). Beamforming in noninvasive Brain-Computer Interfaces. *IEEE Transactions on Biomedical Engineering*, 56(4):1209–1219.
- [Grover et al., 2022] Grover, S., Wen, W., Viswanathan, V., Gill, C. T., and Reinhart, R. M. (2022). Long-lasting, dissociable improvements in working memory and long-term memory in older adults with repetitive neuromodulation. *Nature Neuroscience*, 25(9):1237–1246.

Bibliography

- [Harandi et al., 2017] Harandi, M., Salzmann, M., and Hartley, R. (2017). Dimensionality reduction on SPD manifolds: The emergence of geometry-aware methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(1):48–62.
- [Haselsteiner and Pfurtscheller, 2000] Haselsteiner, E. and Pfurtscheller, G. (2000). Using time-dependent neural networks for EEG classification. *IEEE Transactions on Rehabilitation Engineering*, 8(4):457–463.
- [Haufe et al., 2014] Haufe, S., Meinecke, F., Görgen, K., Dähne, S., Haynes, J.-D., Blankertz, B., and Bießmann, F. (2014). On the interpretation of weight vectors of linear models in multivariate neuroimaging. *Neuroimage*, 87:96–110.
- [Ho et al., 2013] Ho, J., Cheng, G., Salehian, H., and Vemuri, B. (2013). Recursive Karcher expectation estimators and geometric law of large numbers. In *Artificial Intelligence and Statistics*, pages 325–332. PMLR.
- [Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- [Hodgkin and Huxley, 1952] Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4):500.
- [Horev et al., 2016] Horev, I., Yger, F., and Sugiyama, M. (2016). Geometry-aware principal component analysis for symmetric positive definite matrices. In *Asian Conference on Machine Learning*, pages 1–16. PMLR.
- [Hornik, 1991] Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257.
- [Hornik et al., 1989] Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.
- [Hoya et al., 2003] Hoya, T., Hori, G., Bakardjian, H., Nishimura, T., Suzuki, T., Miyawaki, Y., Funase, A., and Cao, J. (2003). Classification of single trial EEG signals by a combined principal+ independent component analysis and probabilistic neural network approach. In *Proc. ICA2003*, volume 197. Citeseer.
- [Huang et al., 2018a] Huang, L., Liu, X., Lang, B., Yu, A., Wang, Y., and Li, B. (2018a). Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- [Huang and Van Gool, 2017] Huang, Z. and Van Gool, L. (2017). A Riemannian network for SPD matrix learning. In *Thirty-first AAAI Conference on Artificial Intelligence*.
- [Huang et al., 2018b] Huang, Z., Wu, J., and Van Gool, L. (2018b). Building deep networks on Grassmann manifolds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

- [Hüper et al., 2007] Hüper, K., Kleinstaub, M., and Leite, F. S. (2007). On the geometry of rolling maps and applications to robotics. *ROBOMAT 07*, page 117.
- [Hüper et al., 2008] Hüper, K., Kleinstaub, M., and Silva Leite, F. (2008). Rolling Stiefel manifolds. *International Journal of Systems Science*, 39(9):881–887.
- [Ionescu et al., 2015a] Ionescu, C., Vantzos, O., and Sminchisescu, C. (2015a). Matrix backpropagation for deep networks with structured layers. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2965–2973.
- [Ionescu et al., 2015b] Ionescu, C., Vantzos, O., and Sminchisescu, C. (2015b). Training deep networks with structured layers by matrix backpropagation. *arXiv preprint arXiv:1509.07838*.
- [Izhikevich, 2007] Izhikevich, E. M. (2007). *Dynamical systems in neuroscience*. MIT press.
- [Izhikevich and Edelman, 2008] Izhikevich, E. M. and Edelman, G. M. (2008). Large-scale model of mammalian thalamocortical systems. *Proceedings of the National Academy of Sciences*, 105(9):3593–3598.
- [Jain et al., 2000] Jain, A. K., Duin, R. P. W., and Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37.
- [Jayaram et al., 2016] Jayaram, V., Alamgir, M., Altun, Y., Scholkopf, B., and Grosse-Wentrup, M. (2016). Transfer learning in Brain-Computer Interfaces. *IEEE Computational Intelligence Magazine*, 11(1):20–31.
- [Jayaram and Barachant, 2018] Jayaram, V. and Barachant, A. (2018). MOABB: Trustworthy algorithm benchmarking for BCIs. *Journal of Neural Engineering*, 15(6):066011.
- [Jemal et al., 2022] Jemal, I., Mezghani, N., Abou-Abbas, L., and Mitiche, A. (2022). An interpretable deep learning classifier for epileptic seizure prediction using EEG data. *IEEE Access*, 10:60141–60150.
- [Jeuris et al., 2012] Jeuris, B., Vandebril, R., and Vandereycken, B. (2012). A survey and comparison of contemporary algorithms for computing the matrix geometric mean. *Electronic Transactions on Numerical Analysis*, 39(ARTICLE):379–402.
- [Jones, 2008] Jones, D. K. (2008). Studying connections in the living human brain with diffusion MRI. *Cortex*, 44(8):936–952.
- [Jordan, 2004] Jordan, K. G. (2004). Emergency EEG and continuous EEG monitoring in acute ischemic stroke. *Journal of Clinical Neurophysiology*, 21(5):341–352.
- [Ju and Guan, 2022] Ju, C. and Guan, C. (2022). Tensor-CSPNet: A novel geometric deep learning framework for motor imagery classification. *IEEE Transactions on Neural Networks and Learning Systems*.

Bibliography

- [Kalunga et al., 2015] Kalunga, E., Chevallier, S., and Barthélemy, Q. (2015). Data augmentation in Riemannian space for Brain-Computer Interfaces. In *STAMMLINS*.
- [Kanamori and Takeda, 2012] Kanamori, T. and Takeda, A. (2012). Non-convex optimization on Stiefel manifold and applications to machine learning. In *Neural Information Processing: 19th International Conference, ICONIP 2012, Doha, Qatar, November 12-15, 2012, Proceedings, Part I 19*, pages 109–116. Springer.
- [Kaufman et al., 2014] Kaufman, M. T., Churchland, M. M., Ryu, S. I., and Shenoy, K. V. (2014). Cortical activity in the null space: Permitting preparation without movement. *Nature Neuroscience*, 17(3):440–448.
- [Kim et al., 2016] Kim, B., Khanna, R., and Koyejo, O. O. (2016). Examples are not enough, learn to criticize! Criticism for interpretability. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.
- [Kim et al., 2020] Kim, Y.-W., Kim, S., Shim, M., Jin, M. J., Jeon, H., Lee, S.-H., and Im, C.-H. (2020). Riemannian classifier enhances the accuracy of machine-learning-based diagnosis of PTSD using resting EEG. *Progress in Neuro-Psychopharmacology and Biological Psychiatry*, 102:109960.
- [Kobler et al., 2022] Kobler, R. J., Hirayama, J.-i., Zhao, Q., and Kawanabe, M. (2022). SPD domain-specific batch normalization to crack interpretable unsupervised domain adaptation in EEG. *arXiv preprint arXiv:2206.01323*.
- [Koles, 1991] Koles, Z. J. (1991). The quantitative extraction and topographic mapping of the abnormal components in the clinical EEG. *Electroencephalography and Clinical Neurophysiology*, 79(6):440–447.
- [Koles and Soong, 1998] Koles, Z. J. and Soong, A. C. (1998). EEG source localization: Implementing the spatio-temporal decomposition approach. *Electroencephalography and Clinical Neurophysiology*, 107(5):343–352.
- [Korczowski et al., 2015] Korczowski, L., Congedo, M., and Jutten, C. (2015). Single-trial classification of multi-user P300-based Brain-Computer Interface using Riemannian geometry. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1769–1772. IEEE.
- [Kratsios and Bilokopytov, 2020] Kratsios, A. and Bilokopytov, I. (2020). Non-Euclidean universal approximation. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 10635–10646. Curran Associates, Inc.
- [Krauledat et al., 2008] Krauledat, M., Tangermann, M., Blankertz, B., and Müller, K.-R. (2008). Towards zero training for Brain-Computer Interfacing. *PLoS One*, 3(8):e2967.

- [Kraus et al., 2016] Kraus, D., Naros, G., Bauer, R., Khademi, F., Leão, M. T., Ziemann, U., and Gharabaghi, A. (2016). Brain state-dependent transcranial magnetic closed-loop stimulation controlled by sensorimotor desynchronization induces robust increase of corticospinal excitability. *Brain Stimulation*, 9(3):415–424.
- [Kübler, 2020] Kübler, A. (2020). The history of BCI: From a vision for the future to real support for personhood in people with locked-in syndrome. *Neuroethics*, 13(2):163–180.
- [Kwak et al., 2017] Kwak, N.-S., Müller, K.-R., and Lee, S.-W. (2017). A convolutional neural network for steady state visual evoked potential classification under ambulatory environment. *PloS One*, 12(2):e0172578.
- [Lang, 2012] Lang, S. (2012). *Fundamentals of differential geometry*, volume 191. Springer Science & Business Media.
- [LeCun et al., 1995] LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995.
- [Ledoit and Wolf, 2004] Ledoit, O. and Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411.
- [Lefèvre and Baillet, 2008] Lefèvre, J. and Baillet, S. (2008). Optical flow and advection on 2-Riemannian manifolds: A common framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):1081–1092.
- [Leite et al., 2017] Leite, J., Morales-Quezada, L., Carvalho, S., Thibaut, A., Doruk, D., Chen, C.-F., Schachter, S. C., Rotenberg, A., and Fregni, F. (2017). Surface EEG-transcranial direct current stimulation (tDCS) closed-loop system. *International Journal of Neural Systems*, 27(06):1750026.
- [Li et al., 2020] Li, J., Fuxin, L., and Todorovic, S. (2020). Efficient Riemannian optimization on the Stiefel manifold via the Cayley transform. In *International Conference on Learning Representations*.
- [Li et al., 2012] Li, Y., Wong, K. M., and de Bruin, H. (2012). Electroencephalogram signals classification for sleep-state decision - a Riemannian geometry approach. *IET Signal Processing*, 6(4):288–299.
- [Li et al., 2009] Li, Y., Wong, K. M., and deBruin, H. (2009). EEG signal classification based on a Riemannian distance measure. In *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*, pages 268–273. IEEE.
- [Lim, 2012] Lim, Y. (2012). Factorizations and geometric means of positive definite matrices. *Linear Algebra and its Applications*, 437(9):2159–2172.
- [Lin, 2019] Lin, Z. (2019). Riemannian geometry of symmetric positive definite matrices via Cholesky decomposition. *SIAM Journal on Matrix Analysis and Applications*, 40(4):1353–1370.

Bibliography

- [Liu et al., 2019] Liu, H., Li, J., Wu, Y., and Ji, R. (2019). Learning neural bag-of-matrix-summarization with Riemannian network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8746–8753.
- [Lotte et al., 2007] Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., and Arnaldi, B. (2007). A review of classification algorithms for EEG-based Brain-Computer Interfaces. *Journal of Neural Engineering*, 4(2):R1.
- [Lotte and Guan, 2010] Lotte, F. and Guan, C. (2010). Regularizing common spatial patterns to improve BCI designs: Unified theory and new algorithms. *IEEE Transactions on Biomedical Engineering*, 58(2):355–362.
- [Lui, 2012] Lui, Y. M. (2012). Advances in matrix manifolds for computer vision. *Image and Vision Computing*, 30(6-7):380–388.
- [Maman et al., 2019] Maman, G., Yair, O., Eytan, D., and Talmon, R. (2019). Domain adaptation using Riemannian geometry of SPD matrices. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4464–4468. IEEE.
- [Manor and Geva, 2015] Manor, R. and Geva, A. B. (2015). Convolutional neural network for multi-category rapid serial visual presentation BCI. *Frontiers in Computational Neuroscience*, 9:146.
- [Mante et al., 2013] Mante, V., Sussillo, D., Shenoy, K. V., and Newsome, W. T. (2013). Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503(7474):78–84.
- [Markdahl et al., 2020] Markdahl, J., Thunberg, J., and Goncalves, J. (2020). High-dimensional Kuramoto models on Stiefel manifolds synchronize complex networks almost globally. *Automatica*, 113:108736.
- [Mayor-Torres et al., 2021] Mayor-Torres, J. M., Medina-DeVilliers, S., Clarkson, T., Lerner, M. D., and Riccardi, G. (2021). Evaluation of interpretability for deep learning algorithms in EEG emotion recognition: A case study in Autism. *arXiv preprint arXiv:2111.13208*.
- [McCulloch and Pitts, 1943] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5:115–133.
- [Millan et al., 2002] Millan, J. d. R., Mouriño, J., Franzé, M., Cincotti, F., Varsta, M., Heikkinen, J., and Babiloni, F. (2002). A local neural classifier for the recognition of EEG patterns associated to mental tasks. *IEEE Transactions on Neural Networks*, 13(3):678–686.
- [Miller, 2019] Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38.

- [Minkowski, 1910] Minkowski, H. (1910). *Geometrie der Zahlen*. B. G. Teubner Verlag.
- [Minsky and Papert, 1969] Minsky, M. and Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry*. MIT Press.
- [Moakher, 2005] Moakher, M. (2005). A differential geometric approach to the geometric mean of symmetric positive-definite matrices. *SIAM Journal on Matrix Analysis and Applications*, 26(3):735–747.
- [Molnar, 2022] Molnar, C. (2022). *Interpretable Machine Learning*. 2 edition.
- [Murdoch et al., 2019] Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., and Yu, B. (2019). Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080.
- [Navarro-Sune et al., 2016] Navarro-Sune, X., Hudson, A., Fallani, F. D. V., Martinerie, J., Witon, A., Pouget, P., Raux, M., Similowski, T., and Chavez, M. (2016). Riemannian geometry applied to detection of respiratory states from EEG signals: The basis for a Brain-Ventilator Interface. *IEEE Transactions on Biomedical Engineering*, 64(5):1138–1148.
- [Nieh et al., 2021] Nieh, E. H., Schottdorf, M., Freeman, N. W., Low, R. J., Lewallen, S., Koay, S. A., Pinto, L., Gauthier, J. L., Brody, C. D., and Tank, D. W. (2021). Geometry of abstract learned knowledge in the hippocampus. *Nature*, 595(7865):80–84.
- [Nielsen and Bhatia, 2013] Nielsen, F. and Bhatia, R. (2013). *Matrix information geometry*. Springer.
- [Nomizu, 1954] Nomizu, K. (1954). Invariant affine connections on homogeneous spaces. *American Journal of Mathematics*, 76(1):33–65.
- [Nunez and Srinivasan, 2006] Nunez, P. L. and Srinivasan, R. (2006). *Electric fields of the brain: the neurophysics of EEG*. Oxford University Press, USA.
- [Palaniappan, 2005] Palaniappan, R. (2005). Brain Computer Interface design using band powers extracted during mental tasks. In *Conference Proceedings. 2nd International IEEE EMBS Conference on Neural Engineering, 2005.*, pages 321–324. IEEE.
- [Pan and Yang, 2010] Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.
- [Pan et al., 2022] Pan, Y.-T., Chou, J.-L., and Wei, C.-S. (2022). MAtt: A Manifold Attention Network for EEG Decoding. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 31116–31129. Curran Associates, Inc.
- [Pennec and Ayache, 1998] Pennec, X. and Ayache, N. (1998). Uniform distribution, distance and expectation problems for geometric features processing. *Journal of Mathematical Imaging and Vision*, 9:49–67.

Bibliography

- [Pennec et al., 2006] Pennec, X., Fillard, P., and Ayache, N. (2006). A Riemannian framework for tensor computing. *International Journal of Computer Vision*, 66:41–66.
- [Pfurtscheller, 1977] Pfurtscheller, G. (1977). Graphical display and statistical evaluation of event-related desynchronization (ERD). *Electroencephalography and Clinical Neurophysiology*, 43(5):757–760.
- [Pillai and Sperling, 2006] Pillai, J. and Sperling, M. R. (2006). Interictal EEG and the diagnosis of epilepsy. *Epilepsia*, 47:14–22.
- [Qi et al., 2010] Qi, C., Gallivan, K. A., and Absil, P.-A. (2010). Riemannian BFGS algorithm with applications. In *Recent Advances in Optimization and its Applications in Engineering: The 14th Belgian-French-German Conference on Optimization*, pages 183–192. Springer.
- [Reed and Cohen Kadosh, 2018] Reed, T. and Cohen Kadosh, R. (2018). Transcranial electrical stimulation (tES) mechanisms and its effects on cortical excitability and connectivity. *Journal of Inherited Metabolic Disease*, 41:1123–1130.
- [Rentmeesters and Absil, 2011] Rentmeesters, Q. and Absil, P.-A. (2011). Algorithm comparison for Karcher mean computation of rotation matrices and diffusion tensors. In *2011 19th European Signal Processing Conference*, pages 2229–2233. IEEE.
- [Riehle and Requin, 1989] Riehle, A. and Requin, J. (1989). Monkey primary motor and premotor cortex: Single-cell activity related to prior information about direction and extent of an intended movement. *Journal of Neurophysiology*, 61(3):534–549.
- [Rivet et al., 2009] Rivet, B., Souloumiac, A., Attina, V., and Gibert, G. (2009). xDAWN algorithm to enhance evoked potentials: Application to Brain-Computer Interfaces. *IEEE Transactions on Biomedical Engineering*, 56(8):2035–2043.
- [Rodrigues et al., 2017] Rodrigues, P. L. C., Bouchard, F., Congedo, M., and Jutten, C. (2017). Dimensionality Reduction for BCI classification using Riemannian geometry. In *Proceedings of the 7th Graz Brain-Computer Interface Conference 2017*, pages 80–85. Graz University of Technology, Austria.
- [Rodrigues et al., 2020] Rodrigues, P. L. C., Congedo, M., and Jutten, C. (2020). Dimensionality transcending: A method for merging BCI datasets with different dimensionalities. *IEEE Transactions on Biomedical Engineering*, 68(2):673–684.
- [Rodrigues et al., 2018] Rodrigues, P. L. C., Jutten, C., and Congedo, M. (2018). Riemannian procrustes analysis: Transfer learning for Brain-Computer Interfaces. *IEEE Transactions on Biomedical Engineering*, 66(8):2390–2401.
- [Rosenblatt, 1958] Rosenblatt, F. (1958). The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386.

- [Roth et al., 1967] Roth, S. R., Serman, M. B., and Clemente, C. D. (1967). Comparison of EEG correlates of reinforcement, internal inhibition and sleep. *Electroencephalography and Clinical Neurophysiology*, 23(6):509–520.
- [Roy et al., 2019] Roy, Y., Banville, H., Albuquerque, I., Gramfort, A., Falk, T. H., and Faubert, J. (2019). Deep learning-based electroencephalography analysis: A systematic review. *Journal of Neural Engineering*, 16(5):051001.
- [Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.
- [Sadler et al., 2014] Sadler, P. T., Quick, K. M., Golub, M. D., Chase, S. M., Ryu, S. I., Tyler-Kabara, E. C., Yu, B. M., and Batista, A. P. (2014). Neural constraints on learning. *Nature*, 512(7515):423–426.
- [Saifutdinova et al., 2019] Saifutdinova, E., Congedo, M., Dudysova, D., Lhotska, L., Koprivova, J., and Gerla, V. (2019). An unsupervised multichannel artifact detection method for sleep EEG based on Riemannian geometry. *Sensors*, 19(3):602.
- [Sand, 1991] Sand, T. (1991). EEG in migraine: A review of the literature. *Functional neurology*, 6(1):7–22.
- [Santhanam et al., 2009] Santhanam, G., Yu, B. M., Gilja, V., Ryu, S. I., Afshar, A., Sahani, M., and Shenoy, K. V. (2009). Factor-analysis methods for higher-performance neural prostheses. *Journal of Neurophysiology*, 102(2):1315–1330.
- [Schirrmester et al., 2017] Schirrmester, R. T., Springenberg, J. T., Fiederer, L. D. J., Glasstetter, M., Eggenberger, K., Tangermann, M., Hutter, F., Burgard, W., and Ball, T. (2017). Deep learning with convolutional neural networks for EEG decoding and visualization. *Human Brain Mapping*, 38(11):5391–5420.
- [Smyth and Winter, 1964] Smyth, V. and Winter, A. (1964). The EEG in migraine. *Electroencephalography and clinical neurophysiology*, 16(1-2):194–202.
- [Sra, 2012] Sra, S. (2012). A new metric on the manifold of kernel matrices with application to matrix geometric means. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.
- [Sra, 2016] Sra, S. (2016). Positive definite matrices and the S-divergence. *Proceedings of the American Mathematical Society*, 144(7):2787–2797.
- [Sturm et al., 2016] Sturm, I., Lapuschkin, S., Samek, W., and Müller, K.-R. (2016). Interpretable deep neural networks for single-trial EEG classification. *Journal of Neuroscience Methods*, 274:141–145.
- [Suh and Kim, 2021] Suh, Y.-J. and Kim, B. H. (2021). Riemannian embedding banks for common spatial patterns with EEG-based SPD neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 854–862.

Bibliography

- [Székely et al., 2007] Székely, G. J., Rizzo, M. L., and Bakirov, N. K. (2007). Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769 – 2794.
- [Tangermann et al., 2012] Tangermann, M., Müller, K.-R., Aertsen, A., Birbaumer, N., Braun, C., Brunner, C., Leeb, R., Mehring, C., Miller, K., Mueller-Putz, G., Nolte, G., Pfurtscheller, G., Preissl, H., Schalk, G., Schlögl, A., Vidaurre, C., Waldert, S., and Blankertz, B. (2012). Review of the BCI Competition IV. *Frontiers in Neuroscience*, 6.
- [Tatum IV, 2021] Tatum IV, W. O. (2021). *Handbook of EEG interpretation*. Springer Publishing Company.
- [Tenenbaum et al., 2000] Tenenbaum, J. B., Silva, V. d., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.
- [Terao and Ugawa, 2002] Terao, Y. and Ugawa, Y. (2002). Basic mechanisms of TMS. *Journal of Clinical Neurophysiology*, 19(4):322–343.
- [Tervo et al., 2022] Tervo, A. E., Nieminen, J. O., Lioumis, P., Metsomaa, J., Souza, V. H., Sinisalo, H., Stenroos, M., Sarvas, J., and Ilmoniemi, R. J. (2022). Closed-loop optimization of transcranial magnetic stimulation with electroencephalography feedback. *Brain Stimulation*, 15(2):523–531.
- [Van Erven and Harremos, 2014] Van Erven, T. and Harremos, P. (2014). Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820.
- [Van Veen and Buckley, 1988] Van Veen, B. D. and Buckley, K. M. (1988). Beamforming: A versatile approach to spatial filtering. *IEEE ASSP magazine*, 5(2):4–24.
- [Vidal, 1973] Vidal, J. J. (1973). Toward direct brain-computer communication. *Annual Review of Biophysics and Bioengineering*, 2(1):157–180.
- [Vidal, 1977] Vidal, J. J. (1977). Real-time detection of brain events in EEG. *Proceedings of the IEEE*, 65(5):633–641.
- [Wan et al., 2021] Wan, Z., Yang, R., Huang, M., Zeng, N., and Liu, X. (2021). A review on transfer learning in EEG signal analysis. *Neurocomputing*, 421:1–14.
- [Wang et al., 2021] Wang, Y., Qiu, S., Ma, X., and He, H. (2021). A prototype-based SPD matrix network for domain adaptation EEG emotion recognition. *Pattern Recognition*, 110:107626.
- [Wei et al., 2022] Wei, X., Faisal, A. A., Grosse-Wentrup, M., Gramfort, A., Chevallier, S., Jayaram, V., Jeunet, C., Bakas, S., Ludwig, S., Barmpas, K., et al. (2022). 2021 BEETL Competition: Advancing Transfer Learning for Subject Independence Heterogenous EEG Data Sets. In *NeurIPS 2021 Competitions and Demonstrations Track*, pages 205–219. PMLR.

- [Wilson et al., 2022] Wilson, D., Gemein, L. A. W., Schirrmeister, R. T., and Ball, T. (2022). Deep Riemannian Networks for EEG Decoding. *arXiv preprint arXiv:2212.10426*.
- [Wolpaw et al., 1991] Wolpaw, J. R., McFarland, D. J., Neat, G. W., and Forneris, C. A. (1991). An EEG-based brain-computer interface for cursor control. *Electroencephalography and Clinical Neurophysiology*, 78(3):252–259.
- [Wu et al., 2020] Wu, D., Xu, Y., and Lu, B.-L. (2020). Transfer learning for EEG-based Brain-Computer Interfaces: A review of progress made since 2016. *IEEE Transactions on Cognitive and Developmental Systems*, 14(1):4–19.
- [Xu et al., 2020] Xu, J., Grosse-Wentrup, M., and Jayaram, V. (2020). Tangent space spatial filters for interpretable and efficient Riemannian classification. *Journal of Neural Engineering*, 17(2):026043.
- [Xu et al., 2019] Xu, J., Jayaram, V., Schölkopf, B., and Grosse-Wentrup, M. (2019). Feature extraction from the Hermitian manifold for Brain-Computer Interfaces. In *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 965–968. IEEE.
- [Xu et al., 2021] Xu, J., Markham, A., Meunier, A., Raggam, P., and Grosse-Wentrup, M. (2021). Distance covariance: A nonlinear extension of Riemannian geometry for EEG-based Brain-Computer Interfacing. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2000–2005. IEEE.
- [Yair et al., 2019] Yair, O., Ben-Chen, M., and Talmon, R. (2019). Parallel transport on the cone manifold of SPD matrices for domain adaptation. *IEEE Transactions on Signal Processing*, 67(7):1797–1811.
- [Yamamoto et al., 2021] Yamamoto, M. S., Yger, F., and Chevallier, S. (2021). Subspace oddity-optimization on product of Stiefel manifolds for EEG data. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1080–1084. IEEE.
- [Yang et al., 2018] Yang, H.-M., Zhang, X.-Y., Yin, F., and Liu, C.-L. (2018). Robust classification with convolutional prototype learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3474–3482.
- [Yger et al., 2016] Yger, F., Berar, M., and Lotte, F. (2016). Riemannian approaches in Brain-Computer Interfaces: A review. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(10):1753–1762.
- [Yin et al., 2017] Yin, Z., Zhao, M., Wang, Y., Yang, J., and Zhang, J. (2017). Recognition of emotions using multimodal physiological signals and an ensemble deep learning model. *Computer Methods and Programs in Biomedicine*, 140:93–110.

Bibliography

- [Yu et al., 2020] Yu, H., Fan, C., and Zhang, Y. (2020). Epilepsy detection in EEG using Grassmann discriminant analysis method. *Computational and Mathematical Methods in Medicine*, 2020.
- [Yuan et al., 2016] Yuan, X., Huang, W., Absil, P.-A., and Gallivan, K. A. (2016). A Riemannian limited-memory BFGS algorithm for computing the matrix geometric mean. *Procedia Computer Science*, 80:2147–2157.
- [Zanini et al., 2017] Zanini, P., Congedo, M., Jutten, C., Said, S., and Berthoumieu, Y. (2017). Transfer learning: A Riemannian geometry framework with applications to Brain-Computer Interfaces. *IEEE Transactions on Biomedical Engineering*, 65(5):1107–1116.
- [Zhang et al., 2022] Zhang, X.-N., Meng, Q.-H., and Zeng, M. (2022). A novel channel selection scheme for olfactory EEG signal classification on Riemannian manifolds. *Journal of Neural Engineering*, 19(4):046006.
- [Zheng and Tse, 2002] Zheng, L. and Tse, D. N. C. (2002). Communication on the Grassmann manifold: A geometric approach to the noncoherent multiple-antenna channel. *IEEE Transactions on Information Theory*, 48(2):359–383.