Journal of Neural Engineering

PAPER

Predicting cognitive load with EEG using Riemannian geometry-based features

Iris Kremer[1,2], Wissam Halimi[1], Andy Walshe[1], Moran Cerf[3] and Pablo Mainar[1,*]

[1] Logitech, Lausanne, Switzerland
[2] École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
[3] Columbia University, New York, NY, United States of America
[*] Author to whom any correspondence should be addressed.

E-mail: pmainar@logitech.com, ikremer@logitech.com, whalimi@logitech.com, awalshe@logitech.com and moran@morancerf.com

RECEIVED
23 January 2024

REVISED
18 July 2024

ACCEPTED FOR PUBLICATION
26 July 2024

PUBLISHED
3 September 2024

Keywords: EEG, cognitive load, Riemannian geometry, transfer learning, symmetric positive definite matrices

Abstract

Objective. We show that electroencephalography (EEG)-based cognitive load (CL) prediction using Riemannian geometry features outperforms existing models. The performance is estimated using Riemannian Procrustes Analysis (RPA) with a test set of subjects unseen during training. *Approach*. Performance is evaluated by using the Minimum Distance to Riemannian Mean model trained on CL classification. The baseline performance is established using spatial covariance matrices of the signal as features. Various novel features are explored and analyzed in depth, including spatial covariance and correlation matrices computed on the EEG signal and its first-order derivative. Furthermore, each RPA step effect on the performance is investigated, and the generalization performance of RPA is compared against a few different generalization methods. *Main results*. Performances are greatly improved by using the spatial covariance matrix of the first-order derivative of the signal as features. Furthermore, this work highlights both the importance and efficiency of RPA for CL prediction: it achieves good generalizability with little amounts of calibration data and largely outperforms all the comparison methods. *Significance*. CL prediction using RPA for generalizability across subjects is an approach worth exploring further, especially for real-world applications where calibration time is limited. Furthermore, the feature exploration uncovers new, promising features that can be used and further experimented within any Riemannian geometry setting.

1. Introduction

Under strenuous conditions of increased demands, i.e. extensive need for memory resources, the brain experiences high so-called cognitive load (CL). The CL Theory provides a model of the brain under such working memory capacity limitation [5]. In this context, CL is defined as the amount of working memory resources used at a particular point in time. It can further be seen as a bottleneck for learning processes [61]. Accurate real-time CL estimates thereby allow for a better understanding of the pathways that enable learning in the brain [52], and for an expanded understanding of some learning disorders and disabilities [21].

Numerous works have attempted to predict the CL experienced by an individual using electroencephalography (EEG) [5, 22, 50]. EEG is useful in this endeavor as it is not invasive, cheap, accessible, and affords high temporal resolution.

Despite the abundance of prior works aiming at predicting CL, prediction performances remain low. This is primarily due to challenges stemming from the high signal variance across subjects and recording sessions, which make the divergence high [23]. Advances in feature extraction methods coupled with machine learning (ML) models have attempted to address the variance challenges [14, 51]. Furthermore, recent emerging techniques in the realm of transfer learning (TL) offer additional solutions to the inter-subject variability challenges [55].

The goal of this work is to investigate and evaluate different feature extraction methods and model configurations to predict CL. We focus on Riemannian Procrustes Analysis (RPA) [46], a TL approach that matches different subjects' data distributions in

Riemannian space. The method performs unsupervised centering and stretching operations, followed by a supervised rotation step that needs some calibration data for each new subject. Riemannian geometry has been demonstrated to work well in a number of EEG-related tasks that use spatial covariance matrices as features spanned over the EEG channels [1, 18, 57]. We assess the generalizability of the model by predicting CL on data from a subject unseen during training. Finally, we investigate the effect of different features on the classification performance and compare our classifiers to traditional non-TL approaches.

We formalize our contributions by addressing the following research questions:

(i) **RQ1**: Is a supervised calibration step (rotation) required to achieve high accuracy using RPA?
(ii) **RQ2**: Are covariance matrices the best features for RPA?
(iii) **RQ3**: What is the trade-off between the amount of supervised calibration data necessary and the RPA accuracy?
(iv) **RQ4**: How does this trade-off compare to other modeling approaches for classifiers, when tested on unseen subjects?

Our work provides an experimental demonstration of RPA application in CL prediction. We carry out an extensive exploration of features suitable for this method, identifying novel representations that improve results obtained using RPA. Finally, we provide a comparison of the performances obtained using RPA with that of other models using non-TL approaches.

2. Related work

2.1. CL measurement and EEG

Increased CL can be identified through various means. Those include physiological measures (e.g. pupil dilation [24], or heart-rate variability [49]), task performance decrease, or reaction time, to name a few [5]. Attempts to measure or predict CL using neural signals have focused on EEG readouts [22, 50] due to their high temporal resolution and non-invasive nature. A large subset of the measures use signal frequency fluctuations as a direct CL estimate [4], whereas others use the EEG as an outcome variable reflecting the emergence of CL [41, 50].

The ability to detect and characterize various mental states using EEG is an extensively researched topic. Among the mental states frequently investigated are those depicting high and low levels of CL. The main frequency bands implicated with CL are: **theta** $[3.5, 7.5]$ Hz, **alpha** $[7.5, 12.5]$ Hz and **beta** $[12.5, 30]$ Hz. Frontal theta band activity was found to increase proportional to the difficulty of the task [5, 17, 48], but decrease when new information is processed [34]. The alpha band activity was shown to strongly decrease when subjects shift from eyes closed to eyes open, making it a useful feature for distinguishing between these two states [11, 39]. The alpha band was also shown to correlate with perceived CL through a decrease in activity in the regions involved in the processing of a task, notably the fronto-temporal region [16, 17, 33, 48]. Finally, studies have shown that increased beta power is linked with increased CL [16, 34, 48] but the functional role of this band is not clear [17].

One challenge with EEG data is its inter-subject variability [23]. This variability makes it difficult to infer EEG properties that are valid across subjects. Furthermore, training ML models on EEG such that they can generalize to unseen subjects has proven challenging. This, in turn, leads researchers to often train individual models for each subject [22, 50]. The latter approach has the natural drawback of being idiosyncratic and thereby less suitable for real-world applications.

2.2. TL

TL is a general term for a set of techniques aiming to improve the performance of an ML model on a task by using knowledge from a related, previously learned task [53]. A variety of TL methods have recently been explored and used in EEG research [55].

As one aim of this work is to overcome the challenge of inter-subject variability, the usage of TL becomes essential. Specifically, we use subspace learning to find a model that generalizes across subjects, by identifying a transformation between each subject's data that maximizes the inter-subject similarity in terms of class distribution. To intuit the concept, we offer the following explanation: in Euclidean space, the Procrustes Analysis algorithm [25] uses centering, stretching, and rotation operations to transform datasets. In the context of brain-computer interfaces (BCIs), a RPA method extends the Euclidean algorithm to the Riemannian manifold [46] $\mathcal{P}(n)$ using symmetric positive definite (SPD) matrices C

$$\mathcal{P}(n) = \left\{ C \in \mathbb{R}^{n \times n} \mid C^T = C, \boldsymbol{x}^T C \boldsymbol{x} > 0, \forall \boldsymbol{x} \in \mathbb{R}^n \right\}. \tag{1}$$

Simply, each subject's data is regarded as a different dataset $D \in \mathcal{D}$ (the set of all subjects). Unlike in the original RPA definition provided by [46], there are more than two datasets involved. The source space $\mathcal{S}$ is now the union of all datasets except for one selected subject D_T. This subject's dataset is considered the target space $\mathcal{T}$

$$\begin{aligned} \mathcal{S} &= \{(C_i, y_i) \text{ for } i = 1, \ldots, |D_S| \text{ for } D_S \in \mathcal{D} \backslash \{D_T\}\}, \\ \mathcal{T} &= \left\{ \left(\widetilde{C}_i, \widetilde{y}_i\right) \text{ for } i = 1, \ldots, |D_T| \right\}, \end{aligned} \tag{2}$$

where $C_i, \widetilde{C}_i \in \mathbb{R}^{n \times n}$ are data points, and $y_i, \widetilde{y}_i \in \{1, \ldots, L\}$ are their class labels. All datasets in both source and target spaces are first re-centered around

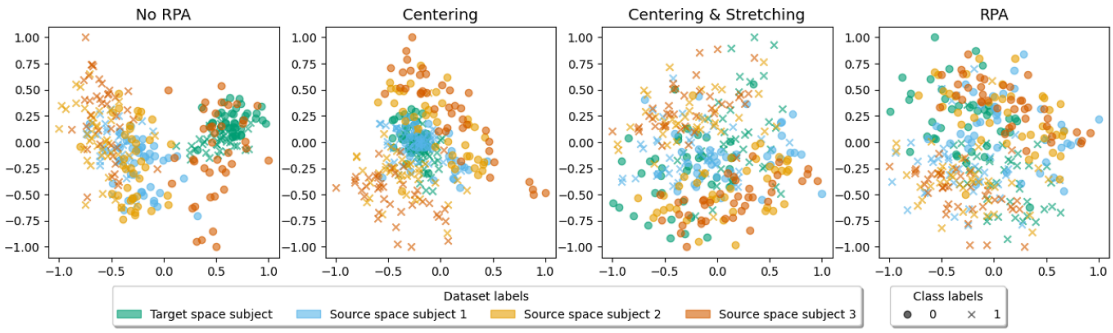

Figure 1. Overview of RPA processing results after each step. This example imagines a situation with four subjects (colors) and two classes (markers). Each subject has 40 randomly generated 2×2 SPD matrices as data points per class. The green subject is selected as target space and the other three subjects are part of the source space. The data points are projected to a 2-dimensional plane using spectral embedding for the purpose of visualisation, but all RPA operations are performed in the Riemannian space. The left-most plot shows the initial state of the data. The second plot visualises the data recentered around the identity matrix. Next comes the data after the stretching step. Finally, the right plot is the data obtained after the rotation step.

the origin of a Riemannian space, which is the identity matrix I_n, using the geometric mean M_D of each dataset:

$$C_i^{(\mathrm{rct})} = M_D^{-1/2} C_i M_D^{-1/2} \quad \text{for } i = 1, \ldots, |D| \text{ for } D \in \mathcal{D}. \tag{3}$$

Following, each set is stretched to reach a unit dispersion:

$$\delta_R^2\left(C_i^{(\mathrm{str})}, I_n\right) = \frac{1}{s}\delta_R^2\left(C_i^{(\mathrm{rct})}, I_n\right) \quad \text{for } i = 1, \ldots, |D| \text{ for } D \in \mathcal{D}, \tag{4}$$

where $\delta_R^2\left(C_i, C_j\right) = \sum_{k=1}^{n} \log^2\left(\lambda_k\right)$ is the Riemannian distance between $C_i, C_j \in \mathcal{P}(n)$ defined using the eigenvalues λ_k of $C_i^{-1}C_j$, and the dispersion $s = \sum_{C_i \in D} \delta_R^2\left(M_D, C_i\right)$. Note that this stretching to unit dispersion differs from the original RPA algorithm proposed in [46] to accommodate a source space composed of multiple datasets. Finally, each dataset from the source space is rotated such that its class distributions reach maximum overlap with the target space's class distributions:

$$C_i^{(\mathrm{rot})} = U_D^T C_i^{(\mathrm{str})} U_D \quad \text{for } i = 1, \ldots, |D| \text{ for } D \in \mathcal{S}, \tag{5}$$

where the orthogonal matrix U_D is obtained by optimizing the following objective:

$$\underset{U_D^T U_D = I_n}{\text{minimize}} \sum_{k=1}^{L} \delta_R^2\left(\widetilde{G}_k, U_D G_k U_D^T\right) \tag{6}$$

where $k \in L$ are all classes present in the data. G_k and $\widetilde{G}_k$ are defined using the geometric mean M of all recentered and stretched matrices from D, $\widetilde{M}$ of recentered and stretched matrices of $\mathcal{T}$, and the class-wise means M_k and $\widetilde{M}_k$ defined as follow:

$$\begin{aligned} M_k &= \mathcal{G}\left(C_i^{(\mathrm{str})} | \text{ for } i = 1, \ldots, |D| \text{ and } y_i = k\right), \\ \widetilde{M}_k &= \mathcal{G}\left(\widetilde{C}_i^{(\mathrm{str})} | \text{ for } i = 1, \ldots, |\mathcal{T}| \text{ and } y_i = k\right), \end{aligned} \tag{7}$$

so that

$$\begin{aligned} G_k &= M^{-1/2} M_k M^{-1/2}, \\ \widetilde{G}_k &= \widetilde{M}^{-1/2} \widetilde{M}_k \widetilde{M}^{-1/2}. \end{aligned} \tag{8}$$

Intuitively, equation (6) minimizes the Riemannian distance between the geometric mean of the same class in D and $\mathcal{T}$, averaged across all classes. Note that in the original RPA definition from [46], a weighting term $w_k \in [0, 1]$ in equation (6) allows to weight each class separately. Also, unlike in the original RPA where the target space is rotated to match the source space, here all datasets within the source space are rotated individually to match the target space. Any dataset from the source space could have been chosen as reference instead of the target space, it would have been mathematically and conceptually equivalent.

As previously mentioned, all equations presented in this section are taken from [46] and adapted for the case where the source space is composed of multiple datasets, as implemented in the python library Pyriemann [8] used in this work to perform RPA. For further details, the mathematical basis of RPA is defined extensively in [46]. Figure 1 visualises all RPA steps performed on randomly generated data for example purposes.

Riemannian geometry has recently achieved impressive performance and robustness in BCI applications [1, 18, 29] as well as in CL evaluation [57]. Here, we explore the possibility of using RPA in the Riemannian space as the TL method to align the subjects, in turn easing the classifier's task of predict CL on an unseen subject.

2.3. EEG features

2.3.1. Standard EEG features

The most common practice in EEG decoding is to utilize statistical features derived from the raw input signals. This is despite the fact that some deep neural

models (i.e. EEGNet) actually use the raw or filtered time-series signals as direct input [37].

The typical features extracted from the raw signal reflect time-domain properties, frequency-domain properties (i.e. absolute and relative powers extracted using multitaper analysis) [17, 32], or time-frequency-domain ones [14].

While these features are typically derived directly from the raw EEG signal, they can also be extracted from a transformed signal (e.g. a derivative of the EEG input) [28, 54].

2.3.2. Riemannian features

RPA requires SPD matrices as features, commonly referred to as Riemannian features. The standard choice for Riemannian space EEG features are the spatial covariance matrices [35, 47]. These matrices have proven to be efficient in EEG decoding both when using Riemannian geometry [10, 46, 62] as well as Common Spatial Patterns [51, 60]. Notably, these covariance matrices do not hold any temporal information since shuffling the spatial signal over the time dimension does not change the covariance. Given that ERP classifications are time-dependent, this loss of temporal information could prove challenging in some analyses [18], but does not pose any problem in our work because CL is not time-locked but rather evolves over continuous time windows.

Several works have tested optimal kernels for the projection of SPD matrices to related spaces [13, 40, 59]. Alternatively, contending algorithms have tried to simplify the covariance matrices by using specific signal processing techniques [58] or post-processing of methods [60].

3. Methods

This section provides all information about methods used in this work to acquire and pre-process data for the analyses and experiments, which are summarized in figure 2. It also provides all information concerning experimental designs and model training procedures.

3.1. Experimental design

Neural signals were acquired as part of a broad endeavor to generate a large corpus of data for various analyses. The endeavor, termed 'Mantis' was a collaboration between Logitech and the Liminal Collective[4] with the aid of the Xperi research group. We recorded 100 subjects, but due to the large dataset size, only a randomly selected subset of 40 subjects is used for the analyses and experiments carried out. The analysis of the remaining subjects is left for future work. Among the 40 randomly selected subjects, there are 23 men and 17 women. The youngest subject is 20 years old and the oldest 71. The median age is 27, the average age 30.7 with a standard deviation of 11.4 years. Handedness of subjects was not collected during acquisition.

Subjects sat in a driving simulator and performed cognitive tasks (resting baseline, Flanker [56] for attention, N-back [30] for memory, simulated driving, NASA-TLX [27] for subjective CL reporting) during four distinct sessions. Each session lasted 40 min (table 1), but only the first session's data for each subject, and only the resting baseline and N-back tasks, were used for the analysis and experiments.

The 4 min baseline (2 min **eyes-closed** and 2 min **eyes-open**) in the beginning of the sessions was used for calibration and control in further analyses. The main CL task, N-back, aimed at exercising working memory at increasing levels of difficulty [30]. Increasing CL was introduced through the escalating task demands. The subject was shown a sequence of patterns (100 milliseconds stimulus presentation followed by a 3 s fixation cross) and is asked to determine whether the most recent stimulus matches the one shown N ($\in 0,1,2,3,4,5$) positions earlier in the sequence. $N = 0$ meant that the subject had to compare each pattern to the very first one in the sequence. We perform 20 continuous trials for each value of N, each trial consisting of displaying one of the 8 possible N-back patterns shown in figure 3 and recording the subject's match/no match response. Subjects answered by pressing one of two buttons (match/no match) for all N values, in order to reduce the muscular and ocular artifacts differences between conditions. Subjects passed the N-back tests in two blocks: first $N \in \{0,2,4\}$, then $N \in \{1,3,5\}$, as shown in table 1. In other words, they performed 20 tasks of 0-back, followed by 20 tasks of 2-back and 20 tasks of 4-back. After a break they did the same for $N \in \{1,3,5\}$.

Physiological data was acquired simultaneously. The data comprised of biometrics time series and high-quality video. The biometric data included EEG, EKG, EOG, SpO2, GSR, and temperature time series. Only EEG data is used in this work.

EEG signals were recorded at 500 Hz sampling rate with a wireless Enobio 32 EEG system (Neuroelectrics) [20]. This device consists of 32 gel-based electrodes, and has been used in previous studies related to CL prediction [2, 3]. The channel used for the analyses are: *Fz*, *Pz*, *F*7, *P*7, *F*8, *P*8, *C*3, *C*4, *F*3, *F*4, *P*3, *P*4, *Fp*1, *O*1, *Fp*2, *O*2. Those electrodes were selected to ensure a coverage across the entire scalp [23, 31].

All subjects signed an informed consent form according to the declaration of Helsinki. The study protocol was approved by the Xperi Research Ethics Committee.

3.2. EEG signals pre-processing

EEG data was first downsampled to 250 Hz. Following, a notch-filter at 50 Hz (FIR filter, 6.6 s

[4] www.liminalcollective.co/.

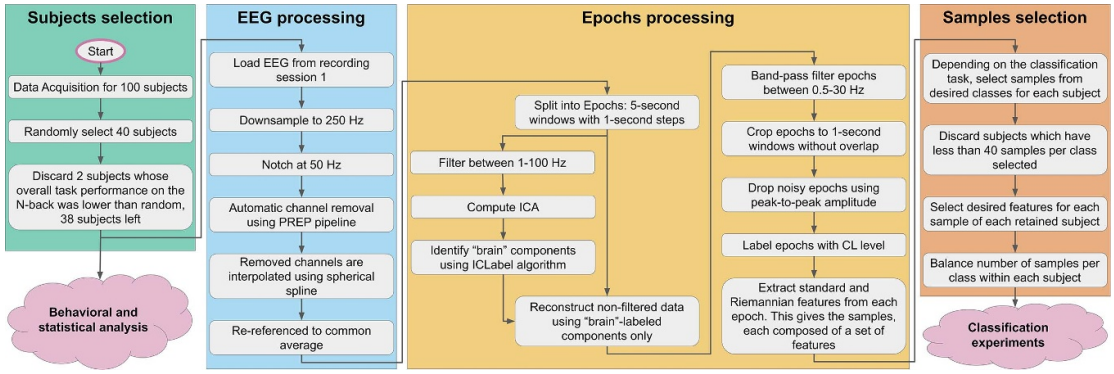

Figure 2. Overview of all data acquisition and pre-processing carried out in this work. The data processing steps are split in several categories for easier understanding. The figure also shows from which processing step the data is taken for the analyses and various classification experiments.

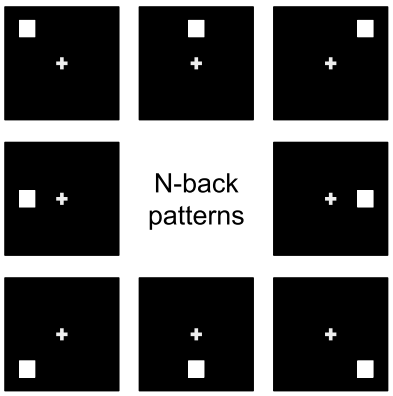

Figure 3. An N-back pattern consists of a white square appearing at a given location within the black square. This figure displaying the 8 possible N-back patterns the subject can be presented with at each N-back trial. During fixation time between pattern display, only the black square with the white fixation cross in the middle remain on screen.

Table 1. First session tasks and the time each of them lasts. Time flows from the top to the bottom of the table.

Task	Time (min.)
Eyes closed	2
Eyes open	2
Break	2
Flanker	1
Break	2
N-back ($N \in \{0,2,4\}$)	3
Break	2
N-Back ($N \in \{1,3,5\}$)	3
Break	2
Driving	20
NASA-TLX	1
Total	40

length) was applied and a PREP pipeline [12] for automatic channel removal was executed. An average of 3.9 out of 32 channels were rejected per subject, with a standard deviation of 2.5. Rejected channels were then interpolated with spherical spline interpolation from surrounding channels. Finally, signals were re-referenced to a common average.

To prepare the data for ML modeling, it was split into epochs (5 s window, 1 s step). These epochs are evenly spaced and do not match with particular N-back trial timings. For example, an epoch might contain the final seconds of one trial and the beginning seconds of the next trial. An Independent Component Analysis (ICA) over a band-pass filtered copy of the data (in the 1–100 Hz. range, FIR filter, 3.3 s length) was calculated. ICLabel [44] was then used to identify brain and artifact components. Next, the original data was reconstructed using the 'brain'-labeled components alone, effectively eliminating as many artifacts from muscular, ocular and other undesired sources as possible. The epochs were band-pass filtered (0.5–30 Hz. range, FIR filter, 6.6 s length) and cropped to 1 s windows with no overlap by trimming away two seconds from the start and the end of the signal respectively. Finally, noisy epochs were removed if their peak-to-peak amplitude was higher than 150 uV in any of the channels.

3.3. Labelling

To determine the CL ground truth we used the objective task difficulty estimate (rather than the subjective NASA-TLX reporting [5]). We assumed that more difficult tasks yield higher CL, as was shown in prior works [57]. Specifically, we used the difficulty of the N-back task as CL score. N-back conditions with $N \in \{0,1\}$ were deemed **low CL**, $N \in \{2,3\}$ were deemed **medium CL**, and $N \in \{4,5\}$ were deemed **high CL**.

To validate the labelling, we estimated the N-back accuracy (see section 4.1). Overall, the median accuracy in % for the tasks was 85.00, 92.50, 67.50, 65.00, 60.00, 56.32 for $N \in \{0,1,2,3,4,5\}$ respectively. Two subjects performed below 12.5% (which corresponds to chance-level performance) on all N-back tasks combined and were excluded from the analyses, as such poor performance indicates they might have misunderstood the task. This leaves 38 subjects

for the analyses and experiments. The accuracy differences between the low, medium and high CL conditions was tested with a paired Wilcoxon signed-rank test, given that the accuracy distributions were skewed. The Bonferroni correction for multiple comparisons was used.

3.4. Features

Two types of features were extracted for the analyses: (1) standard features (i.e. time-domain features), and (2) Riemannian features, i.e. SPD matrices. The latter ones are necessary for the RPA. Table 2 lists all the standard features and table 3 the Riemannian ones as well as the motivation behind the choice of each.

The standard features chosen were selected due to their common usage for prediction using EEG in various contexts.

Spatial covariance matrices were computed in line with existing protocols [10, 46, 60, 62]. Correlation matrices (normalized to [−1,1]) were generated by dividing the covariance matrices by the standard deviation of each signal.

The raw signal's derivative was obtained from the difference between consecutive recording samples in each channel: $x_i' = x_{i+1} - x_i$. We repeated this derivation method once, twice or three times for the first, second and third order derivative of the signal respectively. This derivative operation could be interpreted as a filtering operation where high frequencies are emphasized.

To combine multiple Riemannian features, the SPD matrices of each individual feature were combined into block matrices by placing the SPD matrices in the top-left to bottom-right diagonal of a generalized matrix, and filling the remaining entries with zeros. The resulting block matrix was guaranteed to be SPD as well since it is a square matrix and the bag of eigenvalues of the block matrix is the union of the bags of eigenvalues of each matrix composing it. The block matrix allowed for an investigation of the information complementarity in the signal covariance and its derivative. Similar block matrix was generated from the correlation matrices.

Riemannian features on specific power bands were extracted from the band-pass filtered raw signal in the theta, alpha and beta bands, respectively. Band-pass filtering occurred before the derivation process. Delta band was not investigated because the output was too close to symmetric positive *semi-definite*, which computationally violates the RPA algorithm requirements.

3.5. Training

3.5.1. Models

Minimum Distance to Riemannian Mean model (MDM [9]) was used for the training. This metric derives the Riemannian mean of each class and assigns every data point to the nearest class mean. Class means are computed using all data points belonging to that class from all datasets of the source space after alignment with the target space. The points from the target space were not used in the class means computation. This classification ensures that the output remains within the Riemannian space, requires little computation, and is robust to noise thanks to the usage of geometric means [19]. This provides a useful baseline for the RPA performances.

To assess the MDM performance, we compared the classification to two traditional methods: 1) support vector machine (SVM) with a radial basis function (RBF) kernel and balanced class weights, and 2) EEGNet [37] with parameters $F_1 = 8, D = 2, F_2 = 16, C = 16, T = 250, p = 0.5$, kernel length= 64, batch size= 256. EEGNet was used twice, with different tuning steps (see section 3.5.4).

3.5.2. Experiments

Multiple tests were conducted to assess the performance obtained using RPA, evaluate each model's generalisability across subjects, and address the four research questions stated in section 1.

Specifically, to address **RQ1**, we performed a ***processing experiment*** where we compared the performance of various RPA operations on the standard covariance of the signal samples. To address **RQ2** we performed a ***features experiment*** where we used the RPA model with the various features extracted.

Finally, to address **RQ3** and **RQ4** we conducted a ***calibration experiment*** where we used the covariance of the signal derivatives as features. Here, we trained the models using varying amounts of samples for each class. Then, we compared the output to the ones obtained by including this calibration data in the training procedure of other models.

Performance was assessed by testing the accuracy of the classification of the following states displayed in table 4: (1) baseline eyes open versus baseline eyes closed (named reference task), (2) baseline eyes open versus any of the CL states (named activity presence task), (3) low versus high CL, (4) low versus medium CL and 5) low versus medium versus high CL. For tasks (3) and (4), only data from N-back with $N \in \{0, 2, 4\}$ is used for classification, because samples acquired in the same block, i.e. acquired closely in time, are harder to distinguish from each other and a challenge closer to real device usage which we chose to address. For task (5), there were very few subjects that had at least 40 samples in all three tasks if only data from $N \in \{0, 2, 4\}$ was used. Therefore, data from all $N \in \{0, 1, 2, 3, 4, 5\}$ was used in this case. We computed chance level for comparison by randomly shuffling the labels of samples before starting the training procedure, and averaging the testing accuracy over all subjects. The chance level is computed 20 times and the 95th percentile is reported. This was done for all tasks using the most performant features in the ***features experiment***.

Table 2. Table listing all standard EEG features extracted from clean EEG Epochs for classification. The number of features per sample depends on which kinds of features (listed in the 'Feature names' column) are extracted and from how many channels or channel pairs (listed in the 'Channels' column) they are extracted.

Category	Feature names	Frequency ranges	Channels	#Feat. / sample
Time-Domain (TD)	Mean, Variance, Skewness, Kurtosis, Sample Entropy (SE) [45], Hjorth parameters (activity, complexity & mobility) [28], Detrended Fluctuation Analysis (DFA) [15]	[0.5, 30] Hz	Each of the 16 channels	144 (= 64 for the four moments of distribution + 16 for SE + 48 for Hjorth + 16 for DFA)
Frequency-Domain (FD)	Absolute power bands, Relative power bands	Delta: [0.5, 3.5] Hz Theta: [3.5, 7.5] Hz Alpha: [7.5, 12.5] Hz Beta [12.5, 30] Hz	Each of the 16 channels	128
Dual Channel	Pearson's R coefficient, Time-lagged cross-correlation, (TLCC) offset [15], TLCC maximum Pearson's R [15], Dynamic Time Warping (DTW) [15]	[0.5, 30] Hz	(Fz, Pz), (F7, P7), (F8, P8), (C3, C4), (F3, F4), (P3, P4), (Fp1, O1), (Fp2, O2)	32

Table 3. Table listing all Riemannian features extracted from clean EEG Epochs for classification. Each Riemannian feature named is computed over all 16 channels at once and producing a single matrix per sample.

Feature names	Frequency ranges	Motivation
Covariance (Cov) of the signal	[0.5, 30] Hz	Default SPD matrix features for RPA.
Correlation (Corr) of the signal	[0.5, 30] Hz	'Standardized' covariance matrix.
Cov of first order signal derivative Corr of first order signal derivative Cov of second order signal derivative Cov of third order signal derivative	[0.5, 30] Hz	Extract variance information from signal changes instead of the plain signals.
Cov of the signal + Cov of its first order derivative Corr of the signal + Corr of its first order derivative	[0.5, 30] Hz	Investigate complementary information from signals and their derivatives.
Cov of the first order derivative of the signal on Theta band Cov of the first order derivative of the signal on Alpha band Cov of the first order derivative of the signal on Beta band	[3.5, 7.5] Hz [7.5, 12.5] Hz [12.5, 30] Hz	Investigate the role of each band for CL prediction. Could not be run for Delta band due to matrices being too close to symmetric positive *semi-definite*.

The Python code for the classification was built primarily on the following libraries: (1) MNE [26, 36], Pyprep [6] and EEGLib [15] for basic EEG signal processing, (2) ICA-Label [38] for ICA calculation, (3) Pyriemann [8] for covariance matrices computations as well as RPA and the MDM model in the Riemannian space, (4) scikit-learn [43] for the ML computations outside the Riemannian space, (5) Tensorflow [7] for EEGNet implementation, and (6) Numpy, Pandas and Seaborn for statistical analyses and visualizations.

3.5.3. Processing and features experiments procedure

The training procedure was designed to mimic a likely real device usage. We performed a Leave-One-Out Cross-Validation (LOOCV) training, i.e. training on $n-1$ subject and testing on the last subject, using each subject once as a test subject. Reported

Table 4. Table listing names and classes included for each classification tasks. It also shows how many subjects and samples were involved in total in each classification task after discarding subjects with less than 40 samples per class.

Name	Classes	# Subjects	# Samples
Reference	1. Baseline eyes closed 2. Baseline eyes open	33	7496
Activity presence	1. Baseline eyes open 2. Any CL state	33	7558
Low vs. High CL	1. Low CL (0-back) 2. High CL (4-back)	20	2220
Low vs. Medium CL	1. Low CL (0-back) 2. Medium CL (2-back)	24	2616
Low vs. Medium vs. High CL	1. Low CL (0&1-back) 2. Medium CL (2&3-back) 3. High CL (4&5 back)	33	9135

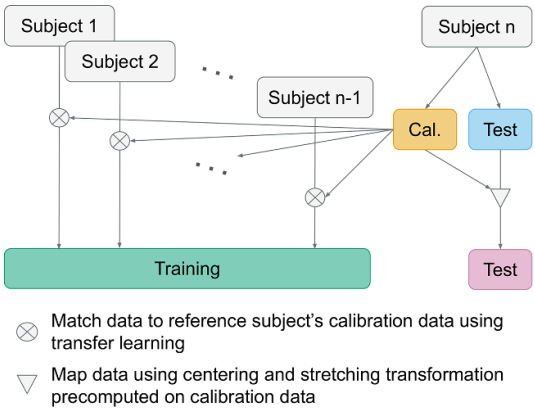

Figure 4. Diagram depicting the training setup in the case where the nth subject is used as test subject, and subjects 1 to $n-1$ are used as training set. Note that the calibration subset from subject n, which corresponds to the target space, is used to identify the RPA transformation to apply on the datasets in the source space. The centering and stretching operations computed for the calibration data are also applied to the samples in the test subset of subject n, resulting in the final test set. The calibration subset is not used for testing.

accuracies are the average performance across all runs with a given configuration.

The training set, i.e. the source space, is therefore composed of $n-1$ subjects, each bringing one dataset. The last subject is the test subject. Since TL requires calibration for each new subject, we split the test subject's data into two sets. The first set acted as the calibration set, equivalent to the target space. It was used to compute the transformations to apply on the data of each subject from the training set to match the test subject's data. The second set was used for testing, and only the centering and stretching operations computed for the calibration set were applied on that testing set, no rotation is performed (figure 4). The calibration set is neither used for training nor for testing. It is used solely to compute the mappings to perform on the datasets in the training set and on the testing set. Inverse transformations between each source space and the target space could be found with identical results, ensuring that the choice of using the calibration set as target space rather than any of the training set subjects did not affect the performance.

In the processing and features experiments, a fixed number of 10 randomly selected samples from each class is used as calibration set. As the data contains varying numbers of samples per subject and class, each classification task was evaluated independently. For a given task, subjects with less than 40 samples in each class were excluded from that task to avoid the risk of overfitting. Since 10 samples per class from the test subject's data are used as calibration set, it must be ensured that there are enough samples remaining in the test set to avoid the risk of overfitting the test set when the training subjects are matched to the calibration set. A test set that is too small is more likely to have extremely well overlapping data distributions with the calibration set. The 40 samples per class constraint resulted in varying numbers of subjects across tasks, as some subjects had enough data for some tasks but not for others. The number of samples was balanced within each subject but not across subjects, meaning some subjects have more data than others for the same task. The final number of subjects and samples for each task is listed in table 4.

Given that we used an MDM model, the inter-subject imbalance of samples did not impact the results' robustness. Intra-subject imbalance of samples does not have a large impact either, as RPA and the MDM model are computed using class means, but the number of samples per class was still balanced within each subjects to follow good ML practices. Finally, also note that as soon as there are more than a few subjects in a training set, the exact number of subjects has little impact on the classification performance obtained with the MDM model on the test set. This is because the data from each subject is matched by maximising overlap of class distributions, therefore all training subjects have similar distribution in data.

3.5.4. Calibration experiment procedure

A calibration experiment was conducted for each number of samples $n\_cal \in \{1, 3, 5, 7, 10, 15, 20\}$ per

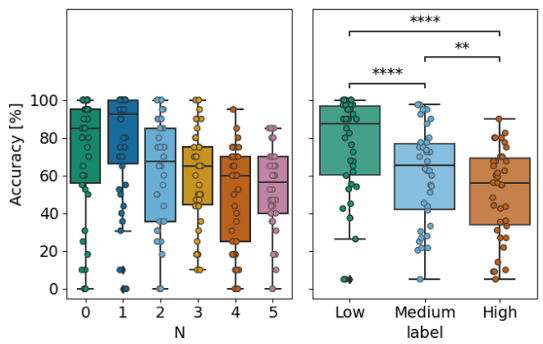

Figure 5. N-back performance for the 38 subjects by N value (left) and CL label (right). Bonferroni-corrected significance levels in the right plot, computed with a Wilcoxon signed rank-sum test, denote ns: $p > 0.05$, $*$: $0.05 > p > 0.01$, $**$: $0.01 > p > 0.001$, $***$: $0.001 > p > 0.0001$, $****$: $0.0001 > p$.

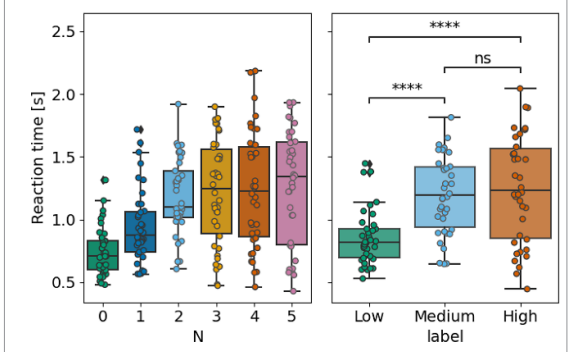

Figure 6. Mean reaction time for each subject in the N-back task by N value (left) and by CL (right). Significance level labels are the same as in figure 5.

class. The limit of 20 samples ensured no over-fitting (with a minimum of 40 samples per class).

LOOCV was conducted for each *n_cal* value, along with the following steps for each *n_cal* and test subject pair. Specifically, subjects and samples were first selected and the covariance of the signal derivatives were computed. Following, the samples were calibrated using RPA and the MDM model was trained and tested. Next, the standard EEG features were extracted from the same samples and the data was normalized. The training and calibration sets were used to fit an SVM model and the performance was calculated using the test set. Finally, the raw EEG signals were used for training, tuning, and testing of the two EEGNet models. Both EEGNet models were trained on the training set with a learning rate of 0.001 over 50 epochs. The fine-tuned EEGNet was subsequently trained on the calibration set with a learning rate of 0.0005 over 10 epochs. The over-fitted EEGNet was tuned in an identical fashion, but over 25 epochs.

Notably, MDM and RPA were not compared to the single-subject models as commonly done in assessment of inter-subject variability [22, 23, 50]. This is because the aim of the work was to produce a generalizable model rather than optimize the solution for a single subject.

4. Results

4.1. Behavioral data analysis

Subjects median accuracies with standard deviation in % were 85.00 ± 32.02, 92.50 ± 27.83, 67.5 ± 29.07, 65.00 ± 25.44, 60.00 ± 26.64, 56.32 ± 24.38 for $N \in \{0, 1, 2, 3, 4, 5\}$ respectively. The mean accuracy for each CL level was computed for each subject, giving 87.50 ± 26.47, 65.51 ± 25.43 and 56.17 ± 23.78 for CL $\in$ Low, Medium, High. Using the Wilcoxon signed rank-sum test with Bonferroni-correction showed a significant difference across all pairwise comparisons of CL levels. The most significant accuracy difference was seen between low and high CL ($T = 31.0$, $p = 0.00\,006$) and between low and medium CL ($T = 73.5$, $p = 0.00\,082$). A significant difference was also noted between medium and high CL ($T = 129.5$, $p = 0.024$). The accuracy in the tasks decreased with difficulty, in line with prior behavioral works.

Subjects median reaction times with standard deviation in seconds were 0.71 ± 0.20, 0.88 ± 0.30, 1.10 ± 0.31, 1.25 ± 0.42, 1.23 ± 0.46, 1.34 ± 0.46 for $N \in 0, 1, 2, 3, 4, 5$ respectively. The mean reaction time for each CL level was computed for each subject, giving 0.82 ± 0.22, 1.19 ± 0.31 and 1.24 ± 0.43 for CL $\in$ Low, Medium, High. The reaction time showed a similar trend as accuracies with significance differences between low and high CL ($T = 53.0$, $p < 10^6$) and between low and medium CL ($T = 73.5$, $p = 0.00\,082$). The reaction time difference between the medium and high CL was not statistically significant ($T = 299.0$, $p = 0.92$). This last observation can be due to the fact that the participants had a time limit to give an answer. All measures and statistical test results are reported in figure 6.

The differences found between the three conditions motivated the distinction between the proposed CL levels. We assume that participants undergo these three different states during the experiment, and attempt to classify them based on their EEG.

4.2. Evaluation of the method

We conducted three experiments aimed at addressing the four research questions investigated in this work. The resutls of the ***processing experiment*** (table 5) indicated that the maximal improvement was achieved when including the last RPA step (rotation).

The results of the ***features experiment*** (tables 6–8) suggest that using the covariance of the first-order derivative of the signal as features instead of the covariance of the non-derived signal strongly improves the model's performance on all tasks.

Table 5. ***Processing experiment*** results. Reported values are the mean ± std in % of the model accuracies obtained using each available subject once as test subject in a LOOCV fashion. Bold results depict the highest mean accuracy obtained in each column, indicating maximal improvement is obtained when performing all RPA steps.

Processing \ Task	Reference	Activity presence	Low vs. High CL	Low vs. Medium CL	Low vs. Medium vs. High CL
No RPA step	58.80 ± 13.13	52.61 ± 4.82	49.11 ± 8.55	50.55 ± 4.76	34.55 ± 2.64
Centering	75.35 ± 14.02	52.38 ± 13.38	52.80 ± 15.90	55.04 ± 9.91	36.30 ± 6.45
Centering + stretching	76.23 ± 13.91	52.94 ± 13.33	52.68 ± 14.40	55.84 ± 9.98	35.64 ± 5.92
RPA (using cov)	**68.81 ± 11.90**	**84.02 ± 10.22**	**66.08 ± 9.92**	**62.20 ± 11.26**	**42.47 ± 3.21**

Table 6. ***Features experiment*** results for covariance, correlation, the sames on the signal's first order derivative and the combination of features extracted from the signal and its derivative. Reported values are the mean ± std of model accuracies, in %, obtained using each available subject once as test subject in a LOOCV fashion. Bolded results indicate the best mean accuracy obtained in each column, to highlight maximum improvement in each task. It shows in all tasks that the covariance of the first order derivative yields the best results.

Features \ Task	Reference	Activity presence	Low vs. High CL	Low vs. Medium CL	Low vs. Medium vs. High CL
Cov	84.02 ± 10.22	68.81 ± 11.90	66.08 ± 9.92	62.20 ± 11.26	42.47 ± 6.21
Corr	76.02 ± 11.97	61.71 ± 8.35	58.55 ± 10.27	56.39 ± 7.08	37.73 ± 5.03
Cov 1st order	**87.80 ± 9.38**	**83.77 ± 10.22**	**78.59 ± 14.17**	**66.75 ± 12.48**	**46.60 ± 10.10**
Corr 1st order	84.59 ± 9.23	76.10 ± 12.59	71.29 ± 12.33	63.30 ± 11.14	41.83 ± 7.00
Cov + Cov 1st order	86.35 ± 11.01	78.61 ± 11.03	71.22 ± 11.72	66.66 ± 10.87	44.93 ± 7.48
Corr + Corr 1st order	81.97 ± 12.33	10.89 ± 11.27	70.62 ± 10.74	62.44 ± 10.79	40.84 ± 7.90

Table 7. ***Features experiment*** results obtained with the covariance on different derivative orders of the signal. Reported values are the mean ± std of model accuracies, in %, obtained again using each available subject once as test subject in a LOOCV fashion. It shows that derivative order of the signal does not strongly impact the effectiveness of using the covariance as features.

Features \ Task	Reference	Activity presence	Low vs. High CL	Low vs. Medium CL	Low vs. Medium vs. High CL
Cov 1st order	87.80 ± 9.38	83.77 ± 10.22	78.59 ± 14.17	66.75 ± 12.48	46.60 ± 10.10
Cov 2nd order	88.76 ± 7.87	82.31 ± 9.81	78.91 ± 13.18	67.01 ± 12.08	48.73 ± 9.05
Cov 3rd order	84.09 ± 10.32	84.80 ± 10.67	77.73 ± 14.38	67.56 ± 13.10	48.14 ± 9.00

Table 8. ***Features experiment*** results obtained with the covariance first order derivative of the signal on the full signal (first row) vs. on specific bands (other three rows). Reported values are the mean ± std of model accuracies, in %, once more obtained using each available subject once as test subject in a LOOCV fashion. It shows that using the covariance on the first order derivative of the beta band gives results closest to those obtained with the same feature computed on the full signal.

Features \ Task	Reference	Activity presence	Low vs. High CL	Low vs. Medium CL	Low vs. Medium vs. High CL
Cov 1st order (over all bands)	87.80 ± 9.38	83.77 ± 10.22	78.59 ± 14.17	66.75 ± 12.48	46.60 ± 10.10
Cov 1st order theta	74.96 ± 13.60	60.04 ± 9.18	57.87 ± 10.53	56.30 ± 5.64	37.24 ± 4.38
Cov 1st order alpha	85.81 ± 11.94	65.12 ± 10.92	59.01 ± 10.23	55.93 ± 6.14	38.11 ± 5.61
Cov 1st order beta	87.04 ± 8.57	83.65 ± 11.13	77.84 ± 12.95	68.46 ± 11.10	46.09 ± 7.42

However, using higher-order derivatives of the signal does not further improve the results. Finally, we observe that most of the CL information retained by these features come from the beta band of the signal.

Additionally, when considering the distribution of accuracies across all LOOCV models in each task (figure 7) we see that the choice of test subject strongly affects the performance of the model.

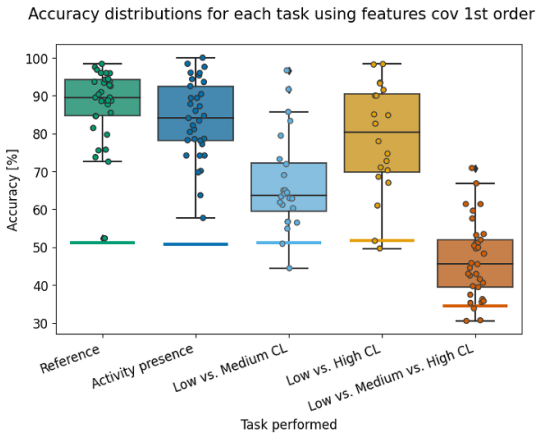

Figure 7. Distributions of accuracies by task obtained during the ***features experiment***, when using the covariance of the derivative of the signal as features. For each task, each available subject is used once as test subject. A bold, colored horizontal line across each box entry column indicates the 95th percentile of the chance level testing accuracy across all subjects for each task, provided for comparison of the results to the chance baseline.

The results of the ***calibration experiment*** (figure 8) show that only around 10 samples per class allow to obtain good results. Furthermore, the comparison with other models shows that RPA allows much better generalization to new subjects than the other approaches investigated.

5. Discussion

We attempt to classify CL using novel Riemannian features in a series of experiments using EEG data obtained from subjects participating in a series of cognitive tasks (namely, memory task in these analyses). We show that RPA combined with MDM models achieves good generalization to new subjects and largely surpasses performances obtained with our comparison models. Furthermore, our analysis uncovers that the feature choice is crucial, and that using the covariance of the first-order derivative of the signal yields the best results, with an average performance of 83.77% on the activity presence task. Evaluation of the method was done by addressing four questions using three experiments, which we denote Processing, Features, and Calibration experiments.

5.1. Processing experiment

In the processing experiment, detailed evaluation of the drivers of the accuracy shows that not performing any RPA step on the samples yields close to random accuracy on all tasks including the reference (table 5). Interestingly, performing only the centering step or both centering and stretching (i.e. standardization in the Riemannian space) without rotation does not improve that performance (For 'activity presence' task: −0.23% on the mean accuracy for the former, +0.33% for the latter compared to no RPA step). On the contrary, applying full RPA, including the rotation step, to the data improves these results a lot (+16.20% for 'activity presence' task).

The centering and stretching steps do not require calibration data and therefore do not adapt specifically to the target space. This can explain the big difference in performance when the rotation step is included.

5.2. Features experiment

The features experiment investigated different kinds of yet mostly unexplored Riemannian features. Table 6 shows that the model accuracies using the correlation as features consistently result in worse performance than those using the covariance of the same signal (for 'activity presence' task: −7.1% when computed on the initial signal, −7.67% when computed on the first-order derivative). No improvement was expected from using the correlation alone, as RPA already performs normalization in the Riemannian space with its first two steps, centering and stretching. However, the results show that standardization in the Euclidean space before normalization in the Riemannian space actually yields poorer performance.

Using the signal's first-order derivatives to compute features largely outperforms using the initial signal across tasks. This suggests that the signal's first-order derivative carries information related to the CL states. The highest performance is shown using the covariance of the first-order derivative (+14.96% for 'activity presence' task).

Using both the covariance of the signal and its derivative does not further improve the results (for 'activity presence' task: −5.16% when using both instead of only covariance on the first-order derivative). Rather, it produces accuracies that lie between those obtained using the signal and its derivative, separately. The same holds for the correlation of the signal and its derivative. This suggests that the combined feature sets are not informatively complementary.

Finally, the large standard deviations observed highlight how differently models are performing depending on the test subject used (figure 7). There are a number of possible explanations for this variation. Recordings might contain variable levels of noise, subjects can have more or less distinctly different brain activities over varying CL states, or they may experience different intensity levels during the acquisition [23]. These variations are likely not caused by the trained models as the MDM model are robust to those variations.

Using the features computed on higher-order derivatives (table 7), we deduce that no large variation in model accuracy can be achieved when increasing the order (for 'activity presence' task: −1.46% for the second-order compared to first-order, +1.03%

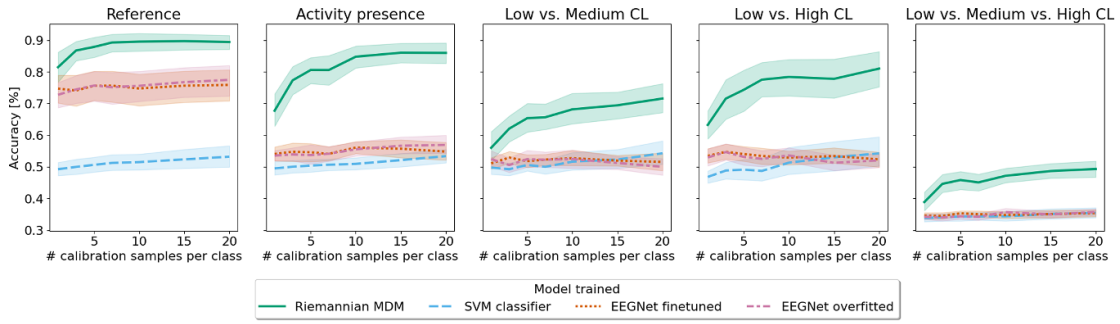

Figure 8. ***Calibration experiment*** results. The number of calibration samples per class is depicted on the x-axis and the average accuracy reached over all subjects used as test subject once (following LOOCV) on the y-axis. Each plot denotes results for one task. Colors and line type correspond to different models. The covariance of the derivative of the full signal (i.e. over $[0.5, 30]$ Hz) was used as features for the MDM classifier. Error margins are 95% confidence intervals computed from the results obtained across different subjects for the same n_cal value and model.

for the third-order). In other words, further emphasizing higher frequencies through additional derivative operations neither improves nor reduces classification performances. The accuracy remains stable across all tasks when the order of the derivative is increased. There only is a big gain in accuracy when using the first derivative compared to the covariance of the original signal, and this gain is maintained in higher-order derivatives.

We observe in table 8 that the frequency band has a significant impact on the model accuracies obtained for each classification task. Because the derivative of a signal could be interpreted as a high-pass filtering operation, it is expected that the information carried in lower frequency bands is going to be affected. We attempt to provide explanations for our results using literature observations and our knowledge of the features.

First, we notice that performances obtained using the theta band are poor (for 'activity presence' task: −23.73% compared to the covariance on all bands), often falling close to random accuracies for all CL tasks. We also observe rather poor performance on the reference task compared to what can be achieved with all bands (−12.84%) and even with standard features such as band powers, suggesting our features are simply inadequate to use on the theta band. While literature observes theta power increases during complex tasks [17, 34, 48], our features are not able to capture this difference sufficiently well for CL classification. A possible explanation is that variation in the theta band are too weak to be properly perceived in the covariance. The length of our epochs is also a likely factor, as one second is very short to observe meaningful information in the slow theta frequencies. Additionally, lower frequencies are attenuated by the derivative operation, likely decreasing the information that the theta band might carry.

Using the alpha band, we observe an interesting behavior: on the reference task, we are able to reach model accuracies comparable to those we have using all bands together (−1.99%). From literature we know that the eyes closed and eyes open states strongly differ in their alpha band [11, 39] manifestation. Hence, it makes sense that distinguishing these two states using the alpha band is a successful approach. However, we see that the performance strongly decreases when it comes to identifying CL levels (for 'activity presence' task: 65.12%). This appears to contradict literature claims, as the alpha band's suppression was found to be a good indicator of CL [5, 39, 48]. However, the suppression of the alpha band during mentally demanding tasks suggests that it does not contribute much to these tasks. So while its power was observed in literature to be indicative of CL, the actual signal and its derivative do not appear to carry much cognitive activity information that the covariance can properly capture. Therefore, our results are in fact coherent with literature observations.

Finally, the results we obtain using the beta band are comparable to those obtained with features from the signal from all bands together (for 'activity presence' task: −0.76% when using features from the beta band only). This suggests that the majority of the CL information we are able to extract using our features comes from the beta band. This is further enhanced by the derivative operation which emphasizes the high frequencies of the signal. It is also consistent with literature that has shown an increased activity in this band to be associated with a working state and higher CL [34, 48].

Higher beta band activity has been related to external muscular or ocular activity [42]. In our experiment we tried to minimize muscular and ocular contamination by keeping the different experimental conditions constant in terms of movements. For all CL tasks the subjects are asked to press one of two buttons (indicating if the pattern matched or not with the pattern N steps back) and to look at the center of the screen. Of course a particular subject can still move more during one particular condition, but this will be

averaged out over time and participants. In addition, during the preprocessing steps, we did ICA and IC-Label to detect and remove components that are not originated in the brain.

5.3. The calibration experiment

The calibration experiment showed that 10 samples are sufficient for accurate calibration (yellow/solid lines in figure 8, which are the results obtained with the MDM model training on RPA-mapped data).

The more distinct two CL classes are (e.g. low versus high) the more valuable are additional samples in the calibration data. This is expected, as more challenging tasks require more precise calibration. That said, even with only 10 samples (corresponding to 10 s of EEG data), we are reaching close to peak accuracies.

Comparing the performance obtained with RPA and MDM model to traditional approaches, when including the calibration data in the training process, shows that all classifiers (SVM, and two EEGNet classifiers) yield poor performance that is virtually at chance level (figure 8). This is true for all but the reference task where an improvement of around 25% is observed with EEGNet models compared to random accuracy. Still, these performances are much lower that those obtained with RPA (about 20% difference) suggesting that RPA is superior to traditional methods, when used with small calibration samples.

6. Conclusions

Our study provides experimental insight into the performances and benefits of using RPA for CL prediction across different subjects.

We demonstrate that generalization performance on CL tasks significantly benefits from the rotation RPA step. This step is used to align all subjects' class distributions.

Exploring a variety of new, easily computable Riemannian features compatible with RPA, our work shows that using the covariance of the EEG derivatives yields improved performances compared to the baseline of using the covariance of the initial signal. Using higher-order derivatives of the signal does not further improve the performance. Also, the CL information relevant to the classification using these features appears to be mostly contained in the EEG beta band.

Few samples per class—in our case 10 - are sufficient to obtain nearly peak classification accuracy, although more samples contribute positively to increased performance.

Finally, comparing different classification approaches proves RPA to be superior, with the alternatives (SVM and EEGNet) showing near chance performance.

One limitation of our work was the choice to disregard the time-series aspect of the signal in the training/testing samples, so one can estimate a potential improvement in future work. Implementation in real-world environments could benefit from personalized calibration, which will likely yield enhanced accuracy in decoding one's mental state. The choice to work solely with RPA method and MDM models may have caped our performance further. There exist a multitude of other models working in the Riemannian or other related spaces. It has not escaped our notice that other methods involving Riemannian geometry can benefit from using the features presented in this work for the decoding of other mental states.

Data availability statement

The data cannot be made publicly available upon publication because they contain commercially sensitive information. The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgment

The authors would like to thank Jocelyn Philippe, Sandro Barissi and Aline Cretenoud for the helpful discussions and suggestions concerning the analyses carried our in this paper. The authors declare that they have no conflicts of interest. Iris Kremer did this work while being an intern at Logitech and a student at EPFL.

ORCID iDs

Iris Kremer https://orcid.org/0009-0001-6314-8702
Moran Cerf https://orcid.org/0000-0002-2012-3177
Pablo Mainar https://orcid.org/0009-0005-9662-492X

References

[1] Abdel-Ghaffar E A, Wu Y and Daoudi M 2022 Subject-dependent emotion recognition system based on multidimensional electroencephalographic signals: a riemannian geometry approach *IEEE Access* **10** 14993–5006
[2] Ahmadi M, Bai H, Chatburn A, Najatabadi M A, Wünsche B C and Billinghurst M 2023 Comparison of physiological cues for cognitive load measures in VR *2023 IEEE Conf. on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)* pp 837–8
[3] Ahmadi M, Michalka S W, Lenzoni S, Ahmadi Najafabadi M, Bai H, Sumich A, Wuensche B and Billinghurst M 2023 Cognitive load measurement with physiological sensors in virtual reality during physical activity *Proc. 29th ACM Symp. on Virtual Reality Software and Technology* (*VRST '23*) (Association for Computing Machinery) (available at: https://doi.org/10.1145/3611659.3615704)

[4] Antonenko P D and Niederhauser D S 2010 The influence of leads on cognitive load and learning in a hypertext environment *Comput. Hum. Behav.* **26** 140–50
[5] Antonenko P, Paas F, Grabner R and Van Gog T 2010 Using electroencephalography to measure cognitive load *Educ. Psychol. Rev.* **22** 425–38
[6] Appelhoff S, Hurst A J, Lawrence A, Li A, Mantilla Ramos Y J, O'Reilly C, Xiang L and Dancker J 2022 PyPREP: a Python implementation of the preprocessing pipeline (PREP) for EEG data (available at: https://doi.org/10.5281/zenodo.6363576)
[7] Abadi M *et al* 2015 TensorFlow: large-scale machine learning on heterogeneous systems (available at: https://github.com/tensorflow/tensorflow)
[8] Barachant A *et al* 2023 pyriemann/pyriemann: v0.5 (available at: https://pyriemann.readthedocs.io/)
[9] Barachant A, Bonnet S, Congedo M and Jutten C 2011 Multiclass brain–computer interface classification by Riemannian geometry *IEEE Trans. Biomed. Eng.* **59** 920–8
[10] Barachant A, Bonnet S, Congedo M and Jutten C 2013 Classification of covariance matrices using a Riemannian-based kernel for BCI applications *Neurocomputing* **112** 172–8
[11] Barry R J, Clarke A R, Johnstone S J, Magee C A and Rushby J A 2007 EEG differences between eyes-closed and eyes-open resting conditions *Clin. Neurophysiol.* **118** 2765–73
[12] Bigdely-Shamlo N, Mullen T, Kothe C, Su K-M and Robbins K A 2015 The PREP pipeline: standardized preprocessing for large-scale EEG analysis *Front. Neuroinform.* **9** 16
[13] Bleuzé A, Mattout J and Congedo M 2021 Transfer learning for the Riemannian tangent space: applications to brain-computer interfaces *2021 Int. Conf. on Engineering and Emerging Technologies (ICEET)* (IEEE) pp 1–6
[14] Boonyakitanont P, Lek-Uthai A, Chomtho K and Songsiri J 2020 A review of feature extraction and performance evaluation in epileptic seizure detection using EEG *Biomed. Signal Process. Control* **57** 101702
[15] Cabanero-Gomez L, Hervas R, Gonzalez I and Rodriguez-Benitez L 2021 EEGLIB: a python module for EEG feature extraction *SoftwareX* **15** 100745
[16] Chen Y and Huang X 2016 Modulation of alpha and beta oscillations during an n-back task with varying temporal memory load *Front. Psychol.* **6** 2031
[17] Chikhi S, Matton N and Blanchet S 2022 EEG power spectral measures of cognitive workload: a meta-analysis *Psychophysiology* **59** e14009
[18] Congedo M, Barachant A and Bhatia R 2017 Riemannian geometry for EEG-based brain-computer interfaces; a primer and a review *Brain-Comput. Interfaces* **4** 155–74
[19] Congedo M, Rodrigues P L C and Jutten C 2019 The riemannian minimum distance to means field classifier *BCI 2019-8th Int. Brain-Computer Interface Conf.* (https://doi.org/10.3217/978-3-85125-682-6-02)
[20] Enobio 32 2023 (available at: www.neuroelectrics.com/solutions/enobio/32) (Accessed 16 October 2023)
[21] Fraga F J, Mamani G Q, Johns E, Tavares G, Falk T H and Phillips N A 2018 Early diagnosis of mild cognitive impairment and alzheimer's with event-related potentials and event-related desynchronization in n-back working memory tasks *Comput. Methods Programs Biomed.* **164** 1–13
[22] Friedman N, Fekete T, Gal K and Shriki O 2019 EEG-based prediction of cognitive load in intelligence tests *Front. Hum. Neurosci.* **13** 191
[23] Gómez L C, Hervas R, Gonzalez I and Villarreal V 2021 Studying the generalisability of cognitive load measured with EEG *Biomed. Signal Process. Control* **70** 103032
[24] Gavas R, Chatterjee D and Sinha A 2017 Estimation of cognitive load based on the pupil size dilation *2017 IEEE Int. Conf. on Systems, Man and Cybernetics (SMC)* pp 1499–504
[25] Gower J C and Dijksterhuis G B 2004 *Procrustes Problems* vol 30 (OUP Oxford)
[26] Gramfort A *et al* 2013 MEG and EEG data analysis with MNE-Python *Front. Neurosci.* **7** 1–13
[27] Hart S G and Staveland L E 1988 Development of NASA-TLX (task load index): results of empirical and theoretical research *Advances in Psychology* vol 52 (Elsevier) pp 139–83
[28] Hjorth B 1970 EEG analysis based on time domain properties *Electroencephalogr. Clin. Neurophysiol.* **29** 306–10
[29] Huang X, Xu Y, Hua J, Yi W, Yin H, Hu R and Wang S 2021 A review on signal processing approaches to reduce calibration time in EEG-based brain–computer interface *Front. Neurosci.* **15** 733546
[30] Jonides J, Schumacher E H, Smith E E, Lauber E J, Awh E, Minoshima S and Koeppe R A 1997 Verbal working memory load affects regional brain activation as measured by PET *J. Cogn. Neurosci.* **9** 462–75
[31] Keskin M, Ooms K, Dogru A O and De Maeyer P 2020 Exploring the cognitive load of expert and novice map users using EEG and eye tracking *ISPRS Int. J. Geo-Inf.* **9** 429
[32] Kim S-E, Behr M K, Ba D and Brown E N 2018 State-space multitaper time-frequency analysis *Proc. Natl Acad. Sci.* **115** E5–E14
[33] Klimesch W, Sauseng P and Hanslmayr S 2007 EEG alpha oscillations: the inhibition-timing hypothesis *Brain Res. Rev.* **53** 63–88
[34] Kumar N and Kumar J 2016 Measurement of cognitive load in HCI systems using EEG power spectrum: an experimental study *Proc. Comput. Sci.* **84** 70–78
[35] Lahav A and Talmon R 2023 Procrustes analysis on the manifold of SPSD matrices for data sets alignment *IEEE Trans. Signal Process.* **71** 1907–21
[36] Larson E *et al* 2023 Mne-python (available at: https://doi.org/10.5281/zenodo.8262486)
[37] Lawhern V J, Solon A J, Waytowich N R, Gordon S M, Hung C P and Lance B J 2018 EEGNET: a compact convolutional neural network for EEG-based brain–computer interfaces *J. Neural Eng.* **15** 056013
[38] Li A, Feitelberg J, Saini A P, Höchenberger R and Scheltienne M 2022 MNE-ICALabel: automatically annotating ICA components with iclabel in python *J. Open Source Softw.* **7** 4484
[39] Li L 2010 The differences among eyes-closed, eyes-open and attention states: an EEG study *2010 6th Int. Conf. on Wireless Communications Networking and Mobile Computing (WiCOM)* (IEEE) pp 1–4
[40] Lin Y-W E, Kluger Y and Talmon R 2021 Hyperbolic procrustes analysis using riemannian geometry *Advances in Neural Information Processing Systems* vol 34 pp 5959–71 (available at: https://proceedings.neurips.cc/paper_files/paper/2021/file/2ed80f6311c1825feb854d78fa969d34-Paper.pdf)
[41] Mills C, Fridman I, Soussou W, Waghray D, Olney A M and D'Mello S K 2017 Put your thinking cap on: detecting cognitive load using EEG during learning *Proc. 7th. Learning Analytics and Knowledge Conf.* pp 80–89
[42] Muthukumaraswamy S 2013 High-frequency brain activity and muscle artifacts in MEG/EEG: a review and recommendations *Front. Hum. Neurosci.* **7** 138
[43] Pedregosa F *et al* 2011 Scikit-learn: machine learning in Python *J. Mach. Learn. Res.* **12** 2825–30
[44] Pion-Tonachini L, Kreutz-Delgado K and Makeig S 2019 ICLabel: an automated electroencephalographic independent component classifier, dataset and website *NeuroImage* **198** 181–97
[45] Richman J S and Moorman J R 2000 Physiological time-series analysis using approximate entropy and sample entropy *Am. J. Physiol. Heart Circ. Physiol.* **278** H2039–49
[46] Rodrigues P L C, Jutten C and Congedo M 2018 Riemannian procrustes analysis: transfer learning for brain–computer interfaces *IEEE Trans. Biomed. Eng.* **66** 2390–401
[47] Rodrigues P L, Congedo M and Jutten C 2020 Dimensionality transcending: a method for merging BCI datasets with different dimensionalities *IEEE Trans. Biomed. Eng.* **68** 673–84

[48] Schapkin S, Raggatz J, Hillmert M and Böckelmann I 2020 EEG correlates of cognitive load in a multiple choice reaction task *Acta Neurobiol. Exp.* **80** 76–89
[49] Solhjoo S, Haigney M C, McBee E, van Merrienboer J J G, Schuwirth L, Artino A R, Battista A, Ratcliffe T A, Lee H D and Durning S J 2019 Heart rate and heart rate variability correlate with clinical reasoning performance and self-reported measures of cognitive load *Sci. Rep.* **9** 14668
[50] Stewart A X, Nuthmann A and Sanguinetti G 2014 Single-trial classification of EEG in a visual object task using ICA and machine learning *J. Neurosci. Methods* **228** 1–14
[51] Sun S and Zhou J 2014 A review of adaptive feature extraction and classification methods for EEG-based brain-computer interfaces *2014 Int. Joint Conf. on Neural Networks (IJCNN)* (IEEE) pp 1746–53
[52] Sweller J 1988 Cognitive load during problem solving: effects on learning *Cogn. Sci.* **12** 257–85
[53] Torrey L and Shavlik J 2010 Transfer learning *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods and Techniques* (IGI global) pp 242–64
[54] Vidaurre C, Krämer N, Blankertz B and Schlögl A 2009 Time domain parameters as a feature for EEG-based brain–computer interfaces *Neural Netw.* **22** 1313–9
[55] 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
[56] Wei H and Zhou R 2020 High working memory load impairs selective attention: EEG signatures *Psychophysiology* **57** e13643
[57] Wriessnegger S C, Raggam P, Kostoglou K and Müller-Putz G R 2021 Mental state detection using riemannian geometry on electroencephalogram brain signals *Front. Hum. Neurosci.* **15** 746081
[58] Wu D, Lance B J, Lawhern V J, Gordon S, Jung T-P and Lin C-T 2017 EEG-based user reaction time estimation using riemannian geometry features *IEEE Trans. Neural Syst. Rehabil. Eng.* **25** 2157–68
[59] Yger F 2013 A review of kernels on covariance matrices for BCI applications *2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)* (IEEE) pp 1–6
[60] Yger F, Lotte F and Sugiyama M 2015 Averaging covariance matrices for EEG signal classification based on the CSP: an empirical study *2015 23rd European Signal Processing Conf. (EUSIPCO)* (IEEE) pp 2721–5
[61] Young J Q, Van Merrienboer J, Durning S and Ten Cate O 2014 Cognitive load theory: implications for medical education: AMEE guide no. 86 *Med. Teacher* **36** 371–84
[62] 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 Trans. Biomed. Eng.* **65** 1107–16