Deep transfer learning for real-time upper-limb motor imagery decoding with a low-cost EEG device

Tim de Boer[1,2]
Supervisors: José M. Azorín[1], Eduardo Iáñez[1], Mario Ortíz García[1], Mark Hoogendoorn[2]

[1] Miguel Hernández University of Elche, Spain
[2] Vrije Universiteit Amsterdam, The Netherlands

Abstract. Brain-computer interfaces (BCIs) can assist stroke or spinal cord injury patients in their rehabilitation process. A BCI system collects data, preprocesses it, and makes predictions about the mental state of the user using machine learning (ML). BCIs have already been used to control robotic arms and exoskeletons, using noninvasive electroencephalography (EEG) to capture brain signals of motor imagery (MI) tasks. However, costs of such systems are high, and practicability is low due to long setup and cleanup times of the EEG device, and time needed to calibrate the ML model before each session. In this study, the goal was to develop a BCI system using a low-cost EEG device with low setup and cleanup times. Additionally, deep learning with cross-subject transfer learning (DTL) was implemented to reduce model calibration time. Data was collected from healthy subjects performing relax, right arm MI, and left arm MI. DTL showed to outperform traditional ML methods, achieving an average accuracy across subjects of 58.7%, with a range between 37.7% to 86.6% for lowest and highest performing subjects. Then, experiments using DTL with real-time feedback were conducted for 4 subjects, on 3 consecutive days. Average model accuracy of DTL after training was 63.2%, and following closed-loop performance averaged 63.4%. Next, subjects played a Dodge game, and reported to feel in control of the system during the game. These results indicate that real-time MI decoding is possible with a low-cost device with low setup and cleanup times, while also reducing model calibration time using DTL, and pave the way towards usage of BCIs for the general public.

Keywords: Brain-Computer Interface · Motor Imagery · Real-Time · Deep Transfer Learning

List of abbreviations

General terms

BCI	Brain-computer interface
MI	Motor imagery
MI-BCI	Motor imagery based brain-computer interface
EEG	Electroencephalography
SNR	Signal-to-noise ratio
ERS	Event-related synchronization
ERD	Event-related desynchronization
ERP	Event-related potential

Preprocessing terms

SP	Butterworth state space filter
filtfilt	Butterworth forward and backward filter
FB	Filter bank
CSP	Common spatial pattern
FBCSP	Filter bank common spatial pattern
RG	Riemannian geometry
TG	Tangent space in Riemannian geometry

Machine learning terms

ML	Machine learning
LDA	Linear discriminant analysis
SVM	Support vector machine
RF	Random forest
MDM	Minimum distance to mean classifier
DL	Deep learning
TL	Transfer learning
DTL	Deep transfer learning
CNN	Convolutional neural network
fz	Filter size
D	Filter depth parameter
lr	Learning rate
pdrop	Dropout probability
ELU	Exponential linear unit
ConFT	Continuous fine-tuning
GenFT	General fine-tuning

1 Introduction

Loss of motor control due to a stroke or spinal cord injury is increasingly present in modern society [1]. Brain-computer interfaces (BCIs) have attracted a lot of attention as a way to assist the patients in their rehabilitation process [2, 3]. A BCI system consists of a user performing some cognitive task, while brain activity is measured by sensors. This is followed by preprocessing of the signals, after which a classification prediction is made of the user's cognitive state. An offline BCI system generally is an open-loop system, where data is captured and saved for preprocessing and classification at a later stage. An online BCI system generally is a closed-loop system. Here, data collection, preprocessing and classification are all done in real-time [4]. In the continuation of this paper, an offline BCI system is referred to when talking about open-loop experiments, and an online BCI system is referred to when talking about closed-loop experiments.

Earlier research demonstrated promising results for closed-loop control of robotic arms and lower-limb exoskeletons using a BCI with noninvasive electroencephalography (EEG) to capture brain signals of motor imagery (MI) (e.g., [5–8]). EEG is regarded as the most practical measurement device for BCIs as of now, due to the ease of setup, removal, and transportability, as well as the relatively low cost when compared to invasive methods, or imaging methods [9]. MI is one of the most common used cognitive tasks for control paradigms in BCIs (MI-BCIs) [10]. MI is defined as the mental process of imagining a motion without executing any movement [11]. MI produces similar brain patterns to the ones associated with the execution of the intended movement, and MI-BCIs are therefore developed to classify those intended movements [12, 13].

However, practicability of the previously used EEG systems is still low, as costs of the devices can still reach €30.000, and movement possibilities are limited due to wires. Moreover, setup and cleanup times can take around 30 minutes each, as often more than 20 electrodes are used, which have to be prepared beforehand, and cleaned afterwards [14]. Besides this, traditional machine learning (ML) algorithms for EEG are heavily reliant on session-specific data, as EEG signals vary both subject-to-subject and day-to-day [15]. For this reason, performance of traditional ML drops considerably across sessions [15]. To re-train a ML model from scratch prior to each session, an obligatory training session is needed, which can last an hour [8].

Deep learning (DL) models have been proposed as a method to create subject-generalizable models (e.g., [16–18]). However, applying the general model directly on data of unseen subjects has shown suboptimal perform [17]. With transfer learning (TL), EEG data from all subjects is used to train a general model, before fine-tuning the model to the data of a unseen subject [19]. When combining DL with cross-subject TL (DTL), only a short training session is needed prior to adapting the general model to the specific session. DTL has been experimented with for open-loop experiments (e.g., [20–23]). To the best of the author's knowledge, DTL has not been applied earlier for closed-loop experiments.

The aim of this work was twofold. First, the goal was to study the feasibility of using a low-cost, wireless BCI system, which also reduces setup and cleanup

times. Here, feasibility of the BCI system is defined as achieving an accuracy higher than 70%, which is generally seen to be the threshold for feasible usage of the BCI system [24]. For the low-cost and wireless BCI system, the Unicorn EEG device was used (Unicorn Hybrid Black, g.tec neurotechnology GmbH, Austria). The Unicorn has a cost of only €1.000, is wireless due to a Bluetooth connection, and has only 8 electrodes. The second goal was to use DTL to reduce training session time prior to closed-loop experiments for upper-limb MI.

Concerning the setup for this study, firstly an open-loop experiment was executed to collect data, and DTL was compared to traditional ML algorithms. As the electrodes of the Unicorn can be used both with or without gel (i.e., dry electrodes), a small side experiment was conducted to investigate the feasibility of using dry EEG electrodes, to further reduce setup and cleanup time. Second, closed-loop experiments were executed using DTL, to evaluate DTL performance using only a training session of 3 trials of 3 minutes each. Here, closed-loop experiments were done for 3 consecutive days to assess effect of subject training on classification performance, as well as the effect of increase of subject-specific, but different session, data on DTL performance.

2 General background

The general steps of an MI-BCI system using EEG follow the order of data collection, data preprocessing, feature extraction, and classification predictions [25]. In this chapter, these steps will be explained in more detail, with current state-of-the-art methods and algorithms.

2.1 Data collection

The most popular method for measuring brain activity in MI-BCIs is with EEG, which measures the weak brain signal registered by an electrode from outside of the scalp in relation to a reference electrode. As EEG measures activity from outside the scalp, spatial resolution is low, meaning individual neurons cannot be distinguished, and mostly activity of groups of neurons around the area beneath the electrode are indicated.

EEG data represents the brain activity as waves of varying frequency, amplitude, and shape. An example of EEG data is given in Figure 1. One can spot similarities in the dynamics of the EEG data between channels (i.e., same signal values captured by different electrodes), indicating big groups of neurons firing together in areas beneath multiple electrodes. One can also spot differences in EEG data between the channels, indicating a smaller group of neurons firing together in the area beneath a specific electrode.

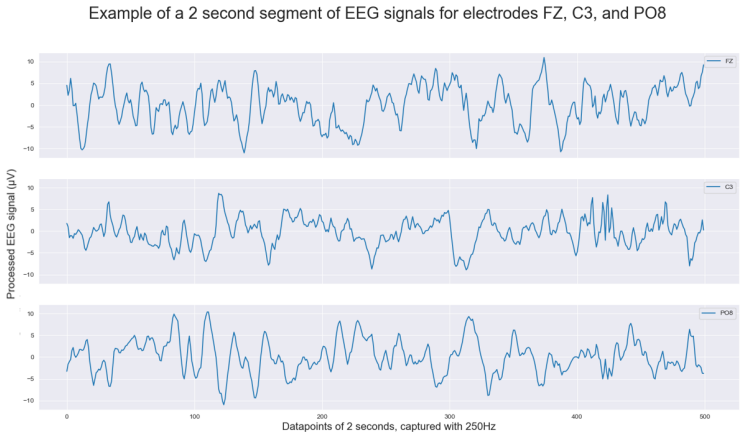


Fig. 1. An example of EEG data captured by electrodes at positions FZ, C3, and PO8 with Unicorn. Signals were bandpass filtered over 1 to 100Hz to reduce noise.

By comparing characteristics of the EEG signal over time, and between the electrodes, patterns and differences can be found which can be used to identify cognitive states. To compare EEG data across studies, the standard 10-20

international system, visible in Figure 2, is used for comparable electrode placement. The highlighted area in Figure 2 contains the electrodes which are placed above the area of the motor cortex. Various motor actions are mapped to different areas of the motor cortex. Therefore, during specific motor actions and MI states, changes in brain activity can be seen around their mapped areas. Thus for MI-BCIs, the electrodes in the highlighted area are of most importance, as differences in brain activity during MI can be captured best by differences in brain activity of the motor cortex, best captured by these electrodes [26].

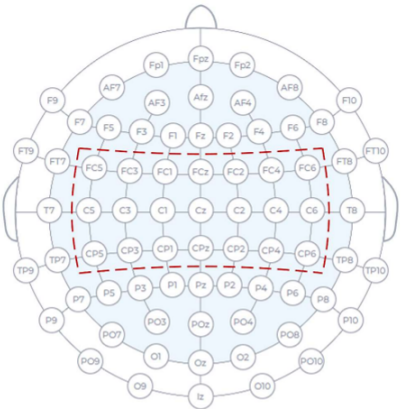


Fig. 2. The standard 10-20 international system of electrode placement. Highlighted area indicate electrodes commonly used for motor imagery tasks. Adapted from [27].

2.2 Preprocessing

Due to the fact that EEG electrodes pick up general electrical noise, muscle activation signals from eye- or neck muscles, and noise due to electrode displacement during experiments, EEG data is regarded to have a low signal-to-noise ratio (SNR). Therefore, approaches have been developed to preprocess data to remove unwanted noise in EEG data, and increase the SNR. A general preprocessing pipeline consists of applying a notch filter, artifact detection, and re-referencing.

As EEG signals often contain power line noise (50Hz in Europe), the common first step is to apply a notch filter, which is a band-pass filter used to remove a narrow band of frequencies, to remove the power line noise.

Due to noise, EEG signals can have sudden peaks in the signal as well, called artifacts. Thus, the second preprocessing step for most BCI pipelines is artifact detection and removal to clean up the data, or to reject bad trials entirely [4]. In a review regarding lower-limb BCIs, it was reported that in 29% of investigated papers, a manual approach of detecting artifacts was used [28]. As is it easy to

manually spot big artifacts or loss of signal, manual removal of those channels is indeed possible and justifiable for these cases. However, when small general noise is present in all channels, or when noise is only sparsely present, manual removal becomes highly subjective. Moreover, during real-time control, manual removal is not feasible due to lack of time. An automatic artifact removal technique based on statistical analysis of the segments has been proposed earlier [29]. Here, a segment was considered to contain an artifact when the power, standard deviation, or amplitude of a trial was 2.5 times higher than the mean signal value of all trials, and such segments were rejected. This statistical approach has been adapted with slight variations to work with real-time experiments [30–32]. In these studies, segments were discarded when the absolute amplitude of the segment would exceed 125 μV, or when the kurtosis of a segment would exceed the standard deviation of the segment by four times. Here, the authors have claimed the method to be successful, although no comparison was provided between this method and other artifact removal methods, or no method at all. In the current study, this claim was assumed to hold. Next to this, the processing time of other methods is often considered too long for real-time processing [33]. Due to the simplicity of above statistical method, processing time is short, making it suitable for real-time processing.

Another common step in preprocessing of EEG data is to re-reference the data using a common average referencing filter (CAR). Although EEG systems measure brain activity relative to a reference electrode, this reference electrode can still record some brain signals, introducing general noise for each electrode. CAR is an simple spatial noise filter, which has been shown to reduce this general noise present in all channels, by subtracting the mean of all channels for each datapoint. CAR has shown to improve the specific signal component of the brain region under each individual EEG electrode [34].

2.3 Feature extraction

Extracting features from EEG data can be divided in two commonly used approaches, namely extracting timepoint features, or spectral features [25]. Timepoint features represent a concatenation of the EEG signals of all channels at one timepoint. Spectral filtering gives spectral features that represent the power (energy) of the EEG data for a given frequency band, over a certain time window. The spectral features capture information of oscillatory activity, such as changes in EEG rhythm amplitudes. When MI is executed, sensorimotor rhythms appear, which are oscillatory events originating in different areas of the brain [35]. An increase in the power of an EEG signal in a certain frequency band is called an event-related synchronization (ERS), and a decrease of EEG signal power an event-related desynchronization (ERD) [36]. Right hand MI leads to a contralateral ERD in the motor cortex (i.e., in the left motor cortex for right hand movement), and to an ERS just after the MI. This phenomenon is illustrated in Figure 3, where the decrease of signal power, an ERD, is clearly visible during the MI task, and immediately after the task, signal power increases, indicating

an ERS. Changes in the signal power during MI is most notable in the μ ($\pm$ 8-12Hz) and β ($\pm$ 16-24Hz) frequency bands [36]. In order to broadly capture the relevant EEG signals from MI, spectral filters for MI-BCIs are therefore often chosen between 4 and 40Hz.

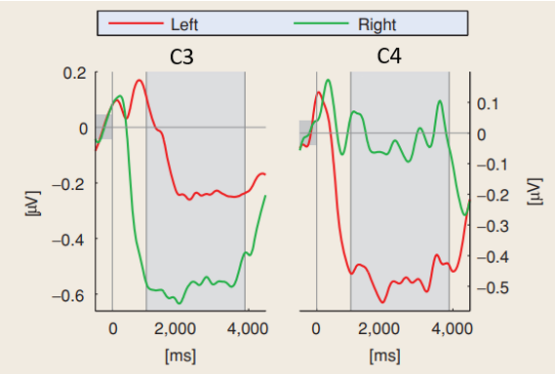


Fig. 3. EEG signals during MI of the left and the right hand. Raw EEG signals of one subject have been band-pass filtered between 9–13 Hz. EEG signals are shown for channels C3 and C4 at the left and right graph, respectively. The shaded area indicate the time period of performing the MI task. Figure altered from [37].

A common next step is to apply a form of spatial filtering [25]. As EEG has poor spatial resolution, underlying brain signals can be spread around several EEG channels. A spatial filter combines information of EEG signals from several channels, often using weighted linear combinations. Thereby, spatial filtering can recover the original signal by gathering the relevant information which is spread over the different channels [38, 39].

By combining spectral and spatial filtering, relevant signals concerning ERD and ERS can be found with spectral filtering, and the brain area from where the signal originate can be identified with spatial filtering. Two commonly used spatial filtering methods are Common Spatial Filter (CSP) and Riemannian Geometry (RG) [37,40]. Both have reached state-of-the-art performance on various BCI tasks (e.g., [37, 40–42]). In the following sections, both methods will be explained. As both methods are not the main aim of this study, only an intuitive explanation will be given, and the interested reader is referred to the original papers (CSP [37], RG [42]) for more details.

Common Spatial Pattern The classic algorithm for spatial filtering in BCIs is CSP [37]. CSP transforms the EEG data by minimizing the variance of one class, and simultaneously maximizing the variance of the other class [37].

An example of the data transformation by CSP for two-channel EEG data is presented in Figure 4. The axes in the left graph represent electrical potential

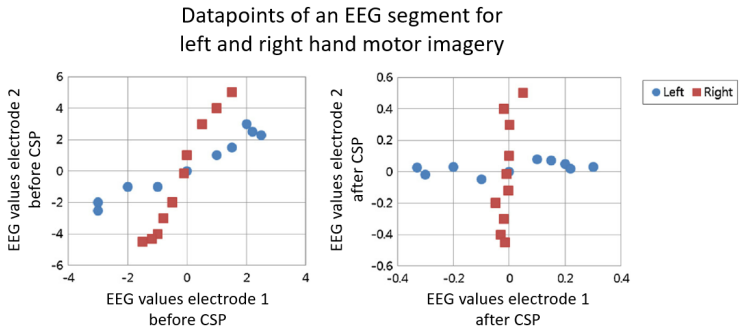


Fig. 4. Example of common spatial pattern (CSP) filtering for binary classification of a segment of EEG data. Adapted from [43].

values measured by two EEG channels, with each point representing a datapoint of one segment of EEG data (e.g., a segment of 1 second). In the right graph, the datapoints are presented in the new space after applying the CSP filters.

CSP can also act as a dimensionality reduction algorithm, as data from multiple electrode channels can be transformed to any chosen number of axes, which are ordered from high to low by the difference in variance between classes. One can choose the number of returned axes as the n components parameter in CSP, and this is often chosen as a lower amount than the amount of electrode channels [37]. Lastly, CSP has been developed for binary classification, but extensions of CSP have been developed for multiclass classification [44].

As subjects exhibit brain activity in different frequency bands, an extension to CSP has been proposed for user-generalizable performance. Rather than using only one spectral filter before CSP, a filter bank (FB) with multiple smaller frequency bands is applied [41]. Combining FB with CSP (FBCSP) allows CSP to choose which frequency bands are most discriminative for the specific subject.

Riemannian Geometry More recently, data transformation using RG has gained more interest in the BCI field. RG has reached state-of-the-art performance for multiple BCI tasks while not having any parameters to tune [25].

RG maps data on a geometrical curved space, called a manifold [40]. Mapping EEG data to a RG manifold entails calculating a covariance matrix for data of each segment (Figure 5a, 5b and 5c). Using labeled training data, a mean covariance matrix can be constructed for each class, visually presented in the manifold in Figure 5c as G. A property of this manifold is that it can be locally approximated by a hyperplane, which is named a tangent space (TG). Intuitively, an image of earth from space can be seen as a manifold, and a 2D satellite image of a part of the earth can be seen as a TG. The TG is useful, as traditional ML algorithms cannot work in a RG manifold, as they assume that data is presented in a hyperplane.

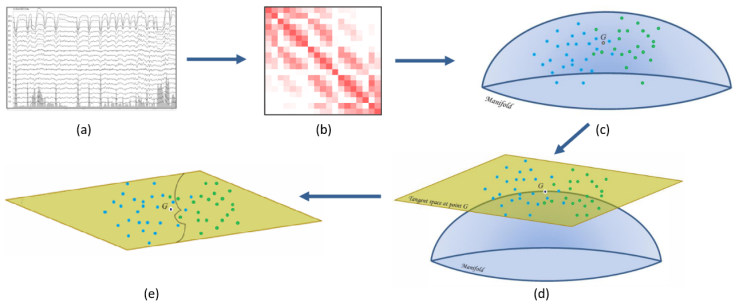


Fig. 5. A pipeline of RG. **a)** A segment of EEG data. **b)** The covariance matrix of the segment of EEG data. **c)** Data of covariance matrices of multiple segments is plotted on the manifold, presented as points, with colors to indicate true label. The mean of all datapoints is indicated with G. **d)** A tangent space on the manifold around the mean of all datapoints (G). Data is mapped from the manifold to the tangent space. **e)** A ML model has decided upon a boundary line, distinguishing data from the two classes. Figures adapted from [45].

2.4 Classification

After spatial filtering, the next step in the MI-BCI pipeline is classification of the MI state using ML.

Common Spatial Pattern After applying FBCSP, traditional ML algorithms can be applied, like linear discriminant analysis (LDA) [46], Support Vector Machine (SVM) [47], and Random Forest (RF) [48]. These ML methods are not the main aim of this study, and therefore only a short explanation will be given. The interested reader is referred to the respective papers for more details. LDA focuses on finding a lower-dimensional feature subspace that maximizes the separability between classes [46]. The main task of SVM is to find the linear hyperplane that can divide the maximum space of the data space [47]. Lastly, RF makes predictions by combining the results from many individual decision trees trained on subsets of the data [48]. Here, decision trees make decision split branches (e.g., if a feature is above or below a certain value).

Riemannian Geometry For the RG Minimum Distance to Mean (MDM) classifier, classification can be done directly in the RG manifold presented in Figure 5c by calculating the distance between an unlabeled datapoint and the mean of the classes, and the unlabeled datapoint is assigned to the class with minimum distance. ML algorithms like LDA, SVM, and RF, can be applied using the EEG data mapped to the TG of RG as presented in Figure 5d [40]. In Figure 5e, the boundary line resulting from training the ML model is presented in the TG.

2.5 Combined filtering and classification with end-to-end deep learning

Deep learning DL methods have seen an enormous increase in popularity over recent years, and have been experimented with in the BCI field as well. A proposed advantage of DL pipelines over FBCSP and RG is the fact that DL can extract features itself from near raw EEG data, meaning data can be pre-processed without filters manually chosen by the researcher [16, 20, 22]. Oftentimes, only a band-pass filter is used in a broad range of 1-100Hz to minimize noise artifacts (e.g., [22]). Without hand-designed features, DL is an end-to-end approach, where feature extraction and classification are combined in the model. The most prominent example of DL in the BCI field is the application of convolutional neural networks (CNNs), which were originally used for computer vision tasks like images, but also for audio signals (e.g., [49–51]). Here, a short overview of CNNs and their application in the BCI field is given. A detailed explanation of the DL model used in this study is provided in Methods, Section 4.2.

Images and audio signals often have a hierarchical structure, where nearby features are important for the current feature, and far away features less so. Using convolutions and non-linearities, a CNN can learn local non-linear features and local patterns in the data. When EEG data is seen as a 2D-array, with the number of time steps as the width, and the number of electrodes as the height, EEG data has similar characteristics to an image or audio signal. Here, data of nearby timepoints is important for the current datapoint, as well as datapoints from the other channels at the same timepoint. A CNN works by using a kernel, which is a sliding window over the data which scans from left to right, and from top to to bottom. For each scan, the dot product of the values of the part of data and the values of the kernel is calculated, essentially summarizing information of the data in that window.

The most popular DL networks for MI-BCIs with EEG data were developed using temporal and spatial convolutions. A temporal convolution has a kernel size of 1 channel to a chosen timeframe, and therefore summarizes the EEG data over a certain timeframe for each channel. A spatial convolution is applied over all channels, for each timepoint, and thus summarizes information over all channels. The convolutions can be applied multiple times with different kernel values, creating different types of summaries of the original data (called feature maps). Here, the goal was to represent the CSP method by representing spectral filtering with temporal convolutions, and spatial filtering with spatial convolutions [16].

Transfer learning Just as CSP and RG, DL suffers from the variability phenomena in EEG data for cross-session and cross-subject data, caused by day-to-day neural variability within subjects, neural variability between subjects, variability due to differences in placement of the EEG cap, and variability in noise signals during sessions for different environments [15]. It has been shown that pre-training a DL model on cross-subject data to learn general features is possible, but directly applying the cross-subject DL model for new subjects may lead to unsatisfactory results [17].

For this reason, recent research in DL for EEG data has been moving towards using DTL techniques (e.g., [21, 28, 52]). DTL for EEG data consists of pre-training a model on cross-subject data. Afterwards, starting from the learned weights of the pre-trained model, the model can be fine-tuned on data of a specific subject or session, using only a small amount of data of that specific subject or session [19]. Multiple DTL strategies exist, where often only weights of the last couple of layers are further trained, while keeping the weights of earlier layers the same as the pre-trained model.

3 Related work

Most earlier work regarding MI-BCIs has been done for open-loop systems. For this reason, open-loop results of earlier implemented pipelines will be discussed extensively. Discussed open-loop techniques will be evaluated with respect to their usability in closed-loop BCI systems, as some open-loop techniques need a complete dataset beforehand and thus cannot be used for a closed-loop BCI system, where data is processed segment by segment in real-time. Next, results using DTL for closed-loop experiments will be discussed.

3.1 Open-loop

CSP, RG, and DL pipelines have been the most commonly researched pipelines for MI-BCIs, achieving state-of-the-art results. For this reason, these approaches will be discussed. Earlier work regarding CSP, RG, and DL approaches is summarized in Table 1. All presented studies in Table 1 are tested on the BCI IV-2a dataset [53] with 4 classes: left hand, right hand, tongue, and feet, to ensure fair comparison between presented studies. The BCI IV-2a dataset is considered the gold standard dataset to evaluate new methods and algorithms for open-loop MI classification. The list of earlier work is not exhaustive, but rather limited to key papers in the development of the discussed algorithms over recent years. Next to this, all included earlier work was tested using the BCI IV-2a, to ensure fair comparison between the earlier works themselves, as well as between earlier work and the current study. Although the classes in the BCI IV-2a dataset (left hand, right hand, tongue, foot) are not the same as the current study (left hand, right hand, relax), the dataset does contain left and right hand MI, making the results of the BCI IV-2a dataset relatively representative to the current study.

Using FBCSP, it was shown that accuracy performance of the full pipeline increased from 50.3% with CSP to 57.2% with FBCSP [41], and from 57.9% to 68.1% in [18]. CSP and FBCSP have strong assumptions about the data. For this reason, performance drops considerably when applied to unseen subjects. In [17], performance of FBCSP trained on multiple sessions of one subject was 68.0% when applied to a new session of same subject, but only 33.0% when applied to a new subject.

The RG approach MDM has been shown to achieve similar performance to FBCSP (63.2% to 65.1%), and a RG-TG with LDA pipeline outperformed FBCSP (70.2% to 65.1%) [40]. RG has also been used to transform cross-subject data to allow a form of cross-subject TL using RG, showing an increase in accuracy performance from 64.0% to 78.0% [54]. However, for the RG TL approach, the full dataset must be available. Therefore, RG TL is not applicable for closed-loop BCI, and was not further considered in the current study.

Regarding DL, EEGNET showed similar performance to FBCSP for within-subject classification (67.0% and 68.0% accuracy, respectively), and outperformed FBCSP for cross-subject classification (39.0% and 33.0% accuracy, respectively). EEGNET showed similar performance for MI classification when compared to a CNN with only a temporal and spatial convolution, but with

more feature maps in each layer (67.0% accuracy for EEGNET, 69.0% CNN for within-subjects, and both 39.0% for cross-subject). Here, EEGNET had only 1716 parameters, compared to 40644 of the other CNN, which was mostly due to using less feature maps in each layer [17]. Having less parameters is an advantage for real-time application, both in terms of lower inference time, as well as needing less training data to calibrate the parameters when compared to having a network with more parameters. Recent work has experimented with bigger networks as well (e.g., MI-EEGNET in [18], showing to outperform EEGNET (69.7% versus 73.0%, respectively), but these networks were not considered in the current study, as inference time would increase for real-time processing, and the data available for pre-training was considered to be too small for proper calibration of the bigger networks.

Performance of EEGNET drops when a model is applied directly to an unseen subject, which can be seen in [17], where EEGNET dropped from 67.0% to 39.0% accuracy between a within-subject design and applying EEGNET to an unseen subject. This can be explained by the variability phenomena in EEG data between subjects and sessions [15]. In [55], applying DTL outperformed an inter-subject DL model with 74.8% accuracy compared to 72.5%. Here, after pre-training, the weights of the first layer of the DL network were kept the same after cross-subject pre-training, while randomly re-initializing the weights of other layers before training those weights with data of a new subject. The DTL model also outperformed inter-subject FBCSP with 74.8% compared to 67.8%. The authors of [23] used DTL by pre-training a DL model on cross-subject data of 8 subjects. After, only the weights of the last two fully-connected layers were randomly re-initialized and trained, while other weights were transferred from the pre-trained model and kept the same. DTL showed higher performance than an inter-subject DL model, achieving 81.0% accuracy compared to 78.0%.

3.2 Closed-loop

Closed-loop performance is often lower when compared to open-loop experiments (e.g., [8, 56]). The drop in performance may be due to the distractions by real-time feedback, causing event-related potentials (ERPs) for the subject [57]. An ERP is a brain response, resulting of a sensory, cognitive, or motor event. During open-loop experiments, distractions are limited, thus decreasing the chance of ERP occurrence.

Closed-loop experiments for left hand versus right hand MI have been done often using CSP and RG based approaches (e.g., [56, 58, 59]). However, due to varying experimental designs, pipelines, and metrics, comparing results is considered not practical. To the best of the author's knowledge, DL models have not been used in closed-loop experiments, owing to the considered long training time for DL models, with would let subjects and users wait for an infeasible amount of time [25]. However, using a small network, combined with using DTL, training time can be greatly decreased [17, 19]. The current study explores the use of DTL for closed-loop upper-limb MI experiments, being the first of its kind.

Table 1. An overview of earlier classification accuracy for the BCI IV-2a dataset with 4 MI classes: left hand, right hand, tongue, and feet [53]. All 22 electrodes were used in each paper.

Study	Frequency band (Hz)	Evaluation design	Pipeline	Accuracy %	Takeaway
Ang et al., 2012 [41]	4-40	i-sub 10-fold CV	CSP + NBPW	50.3	FBCSP outperformed CSP.
	FB: 4-8, 8-12... 36-40		FBCSP + NBPW	57.2	
Barachant et al., 2012 [40]	FB*	i-sub 30-fold CV	FBCSP + LDA	65.1	RG performed equal or better than FBCSP.
	4-30		RG MDM	63.2	
			RG + LDA	70.2	
Rodrigues et al., 2019 [54]	4-40	c-sub 10-fold CV	RG MDM	64.0	RG-TL outperformed RG, but cannot be used for closed-loop.
			RG MDM with TL	78.0	
Lawhern et al., 2018 [17]	FB: 4-8, 8-12... 36-40	i-sub 4-fold CV, test on new session	FBCSP + LR	68.0	First tests with EEGNET showed equal performance to FBCSP and other CNN. Both DL models scored low, but outperformed FBCSP for a cross-subject task.
	4-38		CNN	69.0	
			EEGNET	67.0	
	FB: 4-8, 8-12... 36-40	c-sub 4-fold CV, test on new subject	FBCSP + LR	33.0	
	4-38		CNN	39.0	
			EEGNET	39.0	
Riyad et al., 2021 [18]	4-38	c-sub 5-fold CV	CSP + LR	57.9	A more advanced EEGNET outperformed standard EEGNET, but has too many parameters to use for small dataset of current study.
	FB: 4-8, 8-12... 36-40		FBCSP + LR	68.1	
	8-30		RG + LR	66.3	
	8-38		EEGNET	69.6	
	8-38		MI-EEGNET	73.0	
Zhao et al., 2020 [55]	FB: 4-8, 8-12... 36-40	i-sub, i-ses	FBCSP + ML*	67.8	Using pre-trained DL outperformed FBCSP, and DL without pre-training.
	8-38		DL	72.5	
		c-sub + i-sub, i-ses	DTL	74.8	
Zhang et al., 2021 [23]	FB: 4-8, 8-12... 36-40	i-sub	DL	78.0	Using DTL with FT outperformed DL without TL.
		c-sub + i-sub	DTL with FT	81.0	

Definitions of abbreviations. FB: Filter bank. **i-sub:** Train and test set of same subject. **k-fold CV:** MI data from all sessions are merged and then randomly divided into k equal sets. **NBPW:** Naïve Bayesian Parzen Window classifier. **c-sub:** Train and test set of multiple subjects together. **i-ses:** model is evaluated using data from different sessions (session-independent). **TL:** Transfer Learning. **LR:** Logistic Regression classifier. **d-use:** Transfer learning by directly applying pre-trained model on new subject. **FT:** Transfer learning by fine-tuning on new subject. ***approach not mentioned in paper.

4 Methods open-loop experiments

4.1 Experiment design

Subjects A total of 9 subjects (4 men, mean ± SD age: 22.88 ± 1.36 years, 1 left handed) volunteered to participate. None of the subjects had prior experience with BCIs. The procedures conformed to the ethical committee of the Miguel Hernández University of Elche. Prior to measurements, written informed consent of all subjects was obtained. All collected data was anonymized. Participants did not receive monetary compensation.

Subjects were told to avoid intake of alcohol, caffeine or other drugs 24 hours prior to the experiment to avoid differences in resting state EEG due to these substances [60]. Next to this, subjects were told to not have any substance in their hair (e.g., hair gel, hairspray) during the experiment to avoid disturbances in the connection between the electrodes and scalp.

Overview A full overview of the open-loop experiments is given in Figure 6. A full session consisted of environmental noise capture, a motor execution practice trial, a MI practice trial, and 10 official trials. Based on initial experiments, 10 trials was the maximum amount before subjective attention levels of the subjects dropped too much. A full session lasted for around 75 minutes, including the 10 minutes needed for the set-up of the cap. During the experiment, participants sat in a comfortable chair, facing a computer screen.

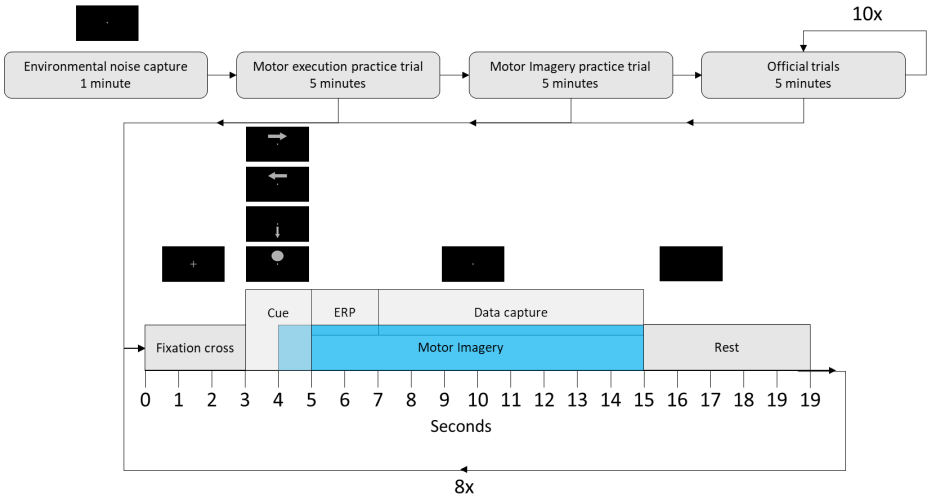


Fig. 6. A full overview of an open-loop experiment, including screenshots of the computer screen presented during the experiments.

At the start of each session, environmental noise was captured for 1 minute for possible usage of artifact removal algorithms. The subject was instructed to stay relaxed, and to move as little as possible during the 1 minute. A small grey dot with black background was presented on the screen, and the subject was instructed to keep their gaze fixed on this dot to avoid sudden eye movement.

Both the practice trials as the official trials lasted for 3 minutes, followed by a 2 minute break. Each trial started with 10 seconds of relax in order to let the subject prepare for the trial and to have a stable EEG signal. During these 10 seconds, the same grey dot as during the environmental noise capture was presented on the screen. During each trial, 8 loops were executed, in which each of the 4 MI tasks were presented 2 times, in random order. The set-up of the loop was inspired by earlier works in MI-BCIs (e.g., [53, 61, 62]).

A loop consisted of a period of gaze fixation and preparation during 3 seconds, in which a fixation cross was presented. Next, a cue for a task appeared together with the small grey dot for 2 seconds. This could be a cue for left arm MI (left arrow), right arm MI (right arrow), legs MI (down arrow) or relax (big circle), see Figure 6. The subject was instructed to start the task immediately after appearance of the cue. After 2 seconds, the cue disappeared, and the small grey dot stayed for 10 seconds during which the subject performed the task. Data was captured starting from the 3rd second of the task performance in order to avoid capturing data of ERPs, which was not of interest in this study. An ERP could appear due to the disappearance of the cue image, or due to starting performing the task. After the task, a black screen appeared, giving the subject time to relax.

Task instructions For the relax task, the subject was asked to fully concentrate on their breathing. MI tasks consisted of imagining squeezing a ball with the right arm or left arm, or pedaling using the legs with a sitting pedal exerciser.

During the first practice trial, the subject executed the motor tasks instead of only imagining. This practice trial served multiple purposes. First, the subjects could practice and get familiar with the tasks. Second, the subjects could be guided towards a kinesthetic experience (i.e., imagining the execution of the task from first person view) of MI rather than a visual experience (i.e., imagining the task from a 3rd person view), which has been shown to result in higher classification accuracies and clearer spatial pattern of brain activity [63]. For this purpose, the subject was asked to imagine the same task, and experience the same sensation, as during the motor execution. Lastly, by performing the motor execution, all subjects were guided towards the same MI task, with the goal of having comparable EEG data between subjects.

During all trials, both hands of the subject rested on their legs with palms upward. The subject was instructed to blink as little as possible from the moment the cue appeared until the end of the task. Additionally, the subject was instructed to avoid jaw clenching, sudden eye movements, and swallowing. During the black screen, the subject was allowed to perform any of above. During

the fixation cross, blinking was still allowed, and advised, in order to reduce the chances of blinking during the task.

Additional remarks During the open-loop sessions, the grey dot was presented in the middle of the screen during the MI tasks. The purpose of this dot was two-fold. First, the dot was used to guide the subject's gaze towards the same position, limiting artifacts due to eye movements. Secondly, the dot was used to have an experiment design as similar as possible to the closed-loop sessions. There, the dot would be used to give feedback to the user regarding the decoding of their MI stage.

Note that data of legs MI was captured in the experiments, but was not further used in this study. As preliminary results showed low classification performance predicting all classes (accuracy $< 50\%$ for almost all subjects), focus was shifted to upper-limb MI only, as such low performance is unfeasible for use in later closed-loop experiments.

4.2 Brain-computer interface pipelines

Pipelines for open-loop experiments were developed for a CSP, RG, and DL approach, with general first steps and approach-specific last steps. A general overview of the pipelines is shown in Figure 7.

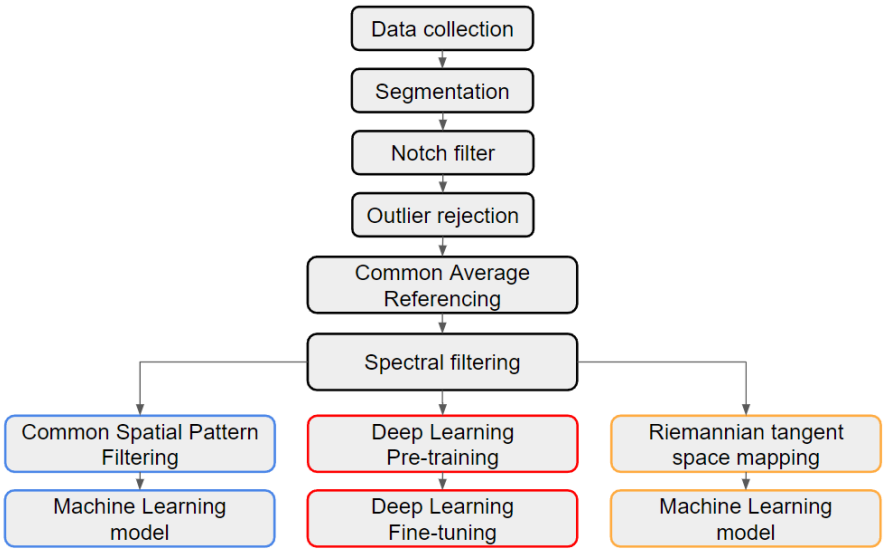


Fig. 7. General overview of implemented pipelines for the brain-computer interface.

Implementation details Code was written in Python 3.9. ML models were constructed using scikit-learn [64] and MNE-Python [65]. DL models were written in PyTorch [66]. Experiments were conducted using PsychoPy [67], and real-time data collection was done using the lab streaming layer LSL (Swartz Center for Computational Neuroscience, UCSD) extension of the Unicorn Recorder software.

Data collection EEG signals were recorded using the Unicorn with 8 gel-based electrodes (Unicorn Hybrid Black, g.tec neurotechnology GmbH, Austria), at a sampling frequency of 250Hz. The standard electrode configuration of the Unicorn was used, with electrodes placed at positions Fz, C3, Cz, C4, Pz, PO7, Oz, and PO8, using the 10-20 international system (Figure 8). Gel was applied between scalp and electrode to reduce impedance, although a side experiment without the use of gel was conducted for one subject. Two reference electrodes were used, with the placement being on the mastoid bones behind the left and right ears.

Segmentation The pipelines were designed in order to simulate the real-time processing needed in the closed-loop experiments. A small amount of data is needed for the pipeline to function, while limiting the delay between user input and system output. For this reason, the first step was segmentation of the data in small windows.

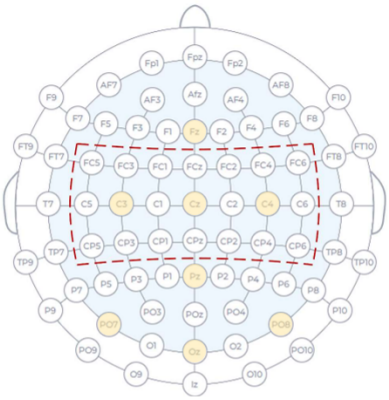


Fig. 8. Unicorn BCI electrode configuration presented in orange for the 10-20 international system. Electrodes of the Unicorn BCI are highlighted in green. Original image adapted from [27].

Notch filter After segmentation, a second-order IIR notch digital filter was applied at 50Hz to suppress power line noise.

Outlier rejection Next, statistical outlier rejection was applied, adapted from [30–32]. A segment was marked as outlier when the absolute amplitude of the segment exceeded 125 μV, or when the kurtosis of the segment exceeded the standard deviation of the segment by four times. When a segment was considered an outlier, the segment was rejected and not taken into account in further data analysis.

Common Average Referencing The next step in the pipelines was applying average referencing using CAR to reduce general noise. CAR was implemented by subtracting the mean of the signals from all channels at each timepoint from the original signals.

Spectral filtering The following step was spectral filtering, either using a state space filter (SP), or a Butterworth forward- and backward filter (filtfilt). SP remembers the last state of the filter from the previous segment, but is only applied in a forward manner and thus introduces phase shift. Filtfilt re-initializes at the start of each segment, thereby destroying the first part of the signal in each segment, but is applied both forward and backward, thereby not introducing phase shift.

Common spatial filtering After spectral filtering, the FBCSP pipeline continues with the applying of CSP, followed by classification using ML with a LDA, SVM, or RF algorithm.

Riemannian geometry For the RG pipeline, data was mapped into a Riemannian manifold using RG. Then, the RG MDM classifier was applied, Also, data was mapped into the tangent space after which LDA, SVM, or RF was applied.

Deep learning Regarding DL, the network architecture was adapted from EEGNET [17]. Optimal parameter values were found with hyperparameter search. As the authors of EEGNET claim that default kernel sizes of the network were found to perform most optimal, kernel sizes of the original EEGNET were only altered whenever this was needed to work with the data shape used in the current study, without having loss of information due to remainders. EEGNET consists of a temporal and spatial convolution, but also added another form of convolution, namely a separable convolution. An overview of the architecture is given in Figure 9 and will be described next.

The first layer of the network was a temporal convolution. The kernel size of the convolution was kept the same as the original EEGNET, at 1x64. The amount of feature maps in this layer, named filter size (fz), was chosen based on hyperparameter search. Each convolution layer applied batch normalization after the convolution, to normalize the output from the previous layer to ensure a normalized input for the following layer [68].

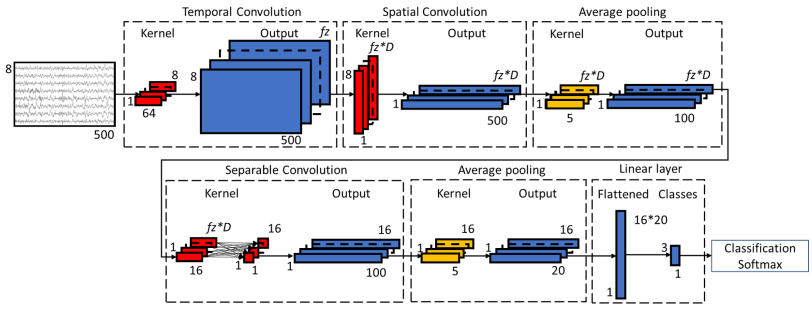


Fig. 9. Network architecture of adapted EEGNET. Parameters fz and D were later found with hyperparameter search.

The second layer was a spatial convolution of size 8x1. Here, the first dimension was equal to the amount of electrode channels. The amount of feature maps from the previous layer was multiplied with a depth parameter (D), which was also chosen based on hyperparameter search. After applying batch normalization, a non-linearity was implemented with an exponential linear unit (ELU) [69]. ELU keeps output x the same when $x > 0$, and transforms x by $\exp(x) - 1$ for $x < 0$.

Then, temporal average pooling with a kernel size of 5x1 with a stride of 5 was applied, averaging data from each 5 timepoints to reduce dimensionality. Here, the size of 5 was chosen, instead of the default value of 8, to not have remainders, as the input size in the current study was divisible by 5, and not by 8. Average pooling was followed by a dropout layer. During training, dropout randomly zeroes some of the elements of the input with a certain probability (*pdrop*). This prevents overfitting on training data by decreasing the dependency of specific nodes to nodes in earlier layers, as having high dependency between node pairs could lead to overfitting on specific features in the training data, which would not be present in the validation and test data [70]. The value of *pdrop* was found by hyperparameter search.

Next, a separable convolution layer was applied, which consisted of a temporal convolution with kernel size 1x16 as used in the original EEGNET, directly followed by a 1x1 conv over the kernels from previous layer grouped over all the feature maps, essentially summarizing the output of the temporal convolution over the feature maps. In this layer, another batch normalization and ELU was applied.

After, another 5x1 average pooling was applied, followed by a dropout layer. Lastly, data was flattened, and a linear layer was applied. As in the original EEGNET, all convolution layers were applied with a stride of 1, and no bias. For temporal convolutions, 'same' padding was used, where zeros are added to the left and right of the input to have same output size after the convolution.

As optimization method, the Adam optimizer was used [71]. The learning rate (lr) of Adam was found by hyperparameter search. Early stopping was

implemented to terminate runs of which the validation loss did not decrease for 5 epochs, to prevent overfitting on training data, and to early terminate bad runs to reduce compute.

4.3 Experimental setup

Experiments were done for each applicable step in the pipelines of 7 to decide for the highest performing available option. Experiments were done by comparing average leave-one-trial-out cross-validation accuracy, unless specified otherwise.

Segmentation Experiments were done with window sizes of 1, 1.5, and 2 seconds, each with a 0.5 second stride. Larger windows were not considered as delay caused by the time to collect the data was considered too large for windows above 2 seconds. This delay has previously been reported to cause frustration by users [72]. To reduce delay between each consecutive prediction, smaller strides were considered as well. However, to ensure that overhead during real-time processing would not happen, overlap between segments was kept at 0.5 seconds, following previous papers of closed-loop experiments with lower-limb exoskeletons [8, 73].

Common Average Referencing Experiments were done to see if applying CAR resulted in better or worse results when compared to not applying CAR.

Spectral Filtering A comparison was made between SP and filtfilt. Next, the order of the filter was considered. Here, experiments were done between a 2nd, 3rd, and 4th order filter. Any higher order filter was considered to affect the signal too much. In the last step regarding bandpass filtering, experiments were done for different frequency bands for RG and DL, and for FBCSP, different FBs were compared. The chosen frequency bands were loosely based on earlier literature, where these bands were used as well (e.g., [22, 40, 73, 74]).

Common spatial pattern For FBCSP, as well as for the ML models, grid search was applied for parameter setting to choose the best performing model, for each run. Here, values for search spaces for hyperparameter search were based on initial experiments. For CSP, the search space for amount of returned axes was $\{2, 4, 8\}$. Regarding ML, for LDA, default settings reached highest performance during initial experiments. For SVM, search space was for C : $\{1, 10, 100\}$, γ : $\{0.1, 0.01, 0.001\}$, and *kernel:* $\{$*rbf, linear*$\}$. For RF, *minimal samples per leaf:* $\{1, 2, 50, 100\}$, *number of estimators:* $\{10, 50, 100, 200\}$, and *criterion:* $\{gini, entropy\}$.

Riemannian geometry As RG does not have any parameters to tune, no hyperparameter search was needed. For ML models, the same grid search as for the FBCSP pipeline was used.

Deep learning Hyperparameter search was applied to find the combination of hyperparameters with highest performance. The developers of EEGNET found that EEG reached highest performance for their dataset with lr was 0.001, for $fz : 8$, for $D : 2$, and for $pdrop : 0.25$, and argue that optimal parameters for all EEG datasets could differ, but are around the same search space [17]. Therefore, search space for hyperparameter search in the current study was done around in the search space around the parameter settings of [17]. For lr, the search space was $\{0.001, 0.005\}$, for $fz : \{4, 8, 16\}$, for $D : \{1, 2, 3\}$, and for $pdrop : \{0.1, 0.25, 0.4\}$. A grid search was implemented, with each run having a maximum of 30 epochs. The search was done by evaluating the validation loss averaged for 3 different configurations of training and validation set, to get an idea of performance for the general population, while limiting compute. Experiments were done with either subject X03, X05 or X08 in the validation set, as these subjects were considered representative for the average population, showing high (X05), average (X03), and low performance (X08) in initial experiments. Data of remaining subjects were put in the training set. The model was trained on the training set. After each epoch, the model was fine-tuned for 5 epochs on the first 4 trials of the validation subject, to get an idea of fine-tune performance, while limiting compute by fine-tuning for only 5 epochs. To get numbers for the validation loss and accuracy, the model was evaluated after the fine-tuning each epoch with the remaining trials of the validation subject.

An overview of the pre-training and fine-tuning process is given in Figure 10, and will be described in detail in the following sections.

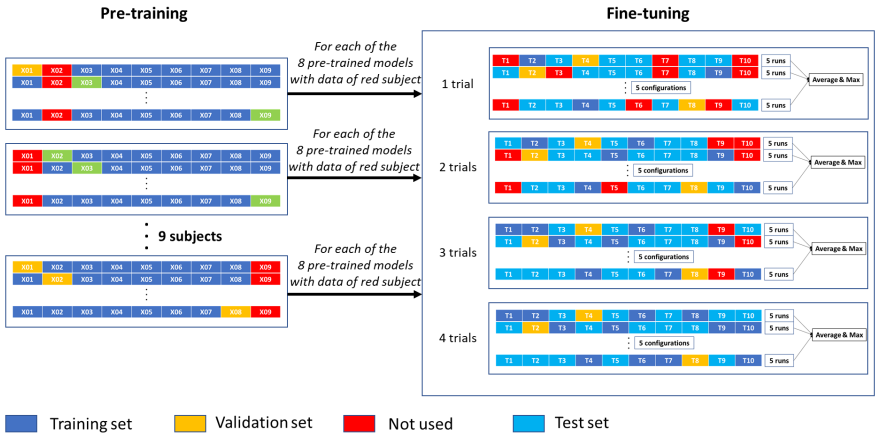


Fig. 10. An overview of pre-training and fine-tuning. In the pre-training phase, blocks indicate data of each subject. In the fine-tuning phase, blocks indicate trials of the specific subject for which fine-tuning was done. Actual fine-tuning configurations were chosen randomly and may not correspond with what is shown in the figure.

EEGNET was pre-trained in a cross-subject design. For each subject, the subject was left out of the dataset and EEGNET was pre-trained 8 times, each time with one of the remaining 8 subjects in the validation set and all remaining trials in the training set. The training approach used for pre-training was the same as the training approach used for hyperparameter search, described above.

For each of the subjects, all 8 pre-trained models were compared for fine-tuning performance. Here, each model was fine-tuned for maximum 20 epochs using either 1, 2, 3 or 4 trials, with 1 trial in the validation set, and 5 trials in the test set. For each pre-trained DL model, experiments were run for 5 different random configurations of training, validation and test splits. Moreover, for each configuration, models were trained for 5 runs to get average performance score, as DL models are initialized randomly and can therefore give different results for each run.

Initial experiments showed that fine-tuning the weights of the entire network lead to highest performance when compared to other methods of fine-tuning. Later, a formal ablation study was conducted to confirm these results. Here, a comparison was made between fine-tuning only the weights of the last layer (linear layer), the last two layers (separable convolution and linear layer), and weights of the entire network. The first two methods were executed either by re-initializing the weights, or starting from pre-trained weights.

4.4 Pipeline comparison setup

For each configuration of subject and trial amount, the highest performing DL model after fine-tuning was noted. Then, FBCSP and RG pipelines were also trained using only data of the specific subject. The same configurations were applied, thus using either 1, 2, 3 or 4 trials, with 1 trial in the validation set, and 5 trials in the test set. Experiments were run for the same 5 different configurations of training, validation and test splits as DL.

5 Results

5.1 Exploratory data analysis

Less than 1% of segments in open-loop trials were considered an outlier. This is probably due to the static nature of the open-loop experiment, which limits the possibility of movement artifacts. All of these segments were marked as outliers due to them exceeding the amplitude threshold. The total number of rejected outliers was low, and evenly distributed regarding classes. Therefore, it was considered that outlier rejection did not introduce any significant bias to the dataset. In Figure 11, an example of a segment marked as outlier is given, in which the artifact is clearly visible in multiple channels.

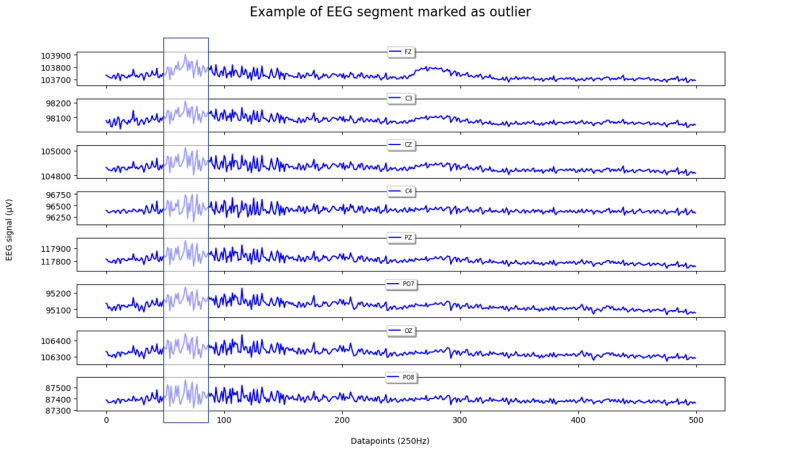


Fig. 11. A segment of EEG data marked as outlier due to exceeding the amplitude threshold. The part of the segment exceeding the threshold is highlighted.

Analysis of EEG signals shows that EEG signals can be more easily distinguished for some subjects than for other subjects, which can be best presented by the spatial filters resulting from CSP. Patterns of spatial filters of CSP can presented in a topographical map, illustrating how the presumed sources project to the scalp. To simplify matters, a CSP with only one bandpass filter of 5 to 25Hz has been implemented, for left hand MI versus right hand MI. In Figure 12, the average topographical pattern of all segments from the top two spatial filters, meaning the two axes with the highest variance between classes, are presented for subjects X02, X05, X08 and X09. Here, for X02, for most segments the filters are able to clearly distinguish left and right arm MI, as the topographical patterns clearly show an ERD on either side of the brain, indicating MI on the other side of the body as explained in Section 2.3. For X05, filters seem to focus on detecting ERS at either side of the brain. However, for X08 and X09, no clear

patterns are visible, indicating that the spatial filters are not able to detect any constant ERD of ERS over all segments, and thus are less able to distinguish left versus right arm MI. These results indicate differences in brain activity for the same classes, thus indicating that ML models would achieve lower performance for subjects X08 and X09 when compared to subject X02 and X05. Initial results (e.g., results in the Appendix, Section 10.1), also confirmed this.

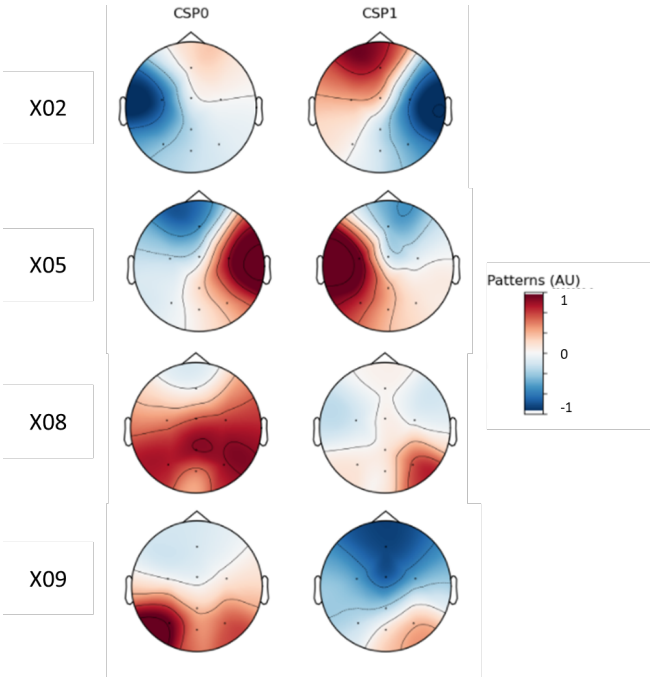


Fig. 12. Normalized topographical patterns resulting from CSP spatial filters of left hand MI versus right hand MI, for subjects X02, X05, X08, and X09. Blue and red areas indicate lower or higher values for topographical patterns, respectively.

5.2 Pipeline experiments

A summary of the results for pipeline experiments is presented in Table 2. Final choices are visible in an overview of the pipelines in Figure 13. Additional explanation about the experimental setup of the pipeline experiments is presented in the Appendix, Section 10.1. Explanation and extensive results for DTL related experiments can be found in Section 10.2.

Segmentation Experiments showed higher average performance for 2 second windows (56.7%) when compared to 1.5 (55.0%) and 1 second (53.2%) windows.

Table 2. Averaged leave-one-trial-out validation accuracy for finding optimal pipeline parameters. Best results per parameter are highlighted in bold.

Name	Parameters	FBCSP	RG	DL
Window size*	1 sec	-	53.2	-
	1.5 sec	-	55.0	-
	2sec	-	**56.7**	-
CAR*	Without	**61.1**	**58.0**	41.2
	With	60.8	55.2	**50.1**
Filter type	SP	-	61.3	-
	filtfilt	-	**61.4**	-
Filter order*	2	**64.4**	**56.7**	-
	3	63.3	56.3	-
	4	64.2	56.6	-
FB bands	5-25Hz in bands of 5Hz	**63.9**	/	/
	5-35Hz in bands of 5Hz	62.0	/	/
	4-40Hz in bands of 4Hz	62.8	/	/
Filter band	8-35	/	**59.8**	**58.0**
	4-40	/	59.1	56.4
	1-100	/	57.1	54.0
ML models	MDM	/	59.8	/
	LDA	**63.9**	59.7	/
	SVM	63.2	**61.8**	/
	RF	62.5	59.3	/

Meaning of symbols. -: Experiments not done for this pipeline due to time restrictions, results from other pipelines taken as indication. /: Parameter not relevant for this pipeline. *: Experiments ran for subjects X01, X02 and X03 taken as indication for all subjects to limit compute.

Higher performance for the bigger window sizes when compared to lower window sizes was expected, as a bigger window size results in individual segments containing more information about the oscillatory activity, and predictions are therefore done more informed.

Common Average Referencing CAR was only applied for DL, as DL performed better using CAR (50.1%) when compared to not using CAR (41.2%). Running CAR before FBCSP resulting in minimally worse results when compared to not using CAR (60.8% and 61.1%, respectively), and RG performed considerably worse using CAR when compared to not using CAR (55.2% and 58.0%, respectively). Lower results for FBCSP and RG can be due to the assumption of both models that EEG data from different channels is independent [75], and by using CAR data is shared across channels, and this assumption is not met.

Spectral filtering Little to no difference was found comparing the two different spectral filtering methods SP and filtfilt (61.3% and 61.4%, respectively). SP was

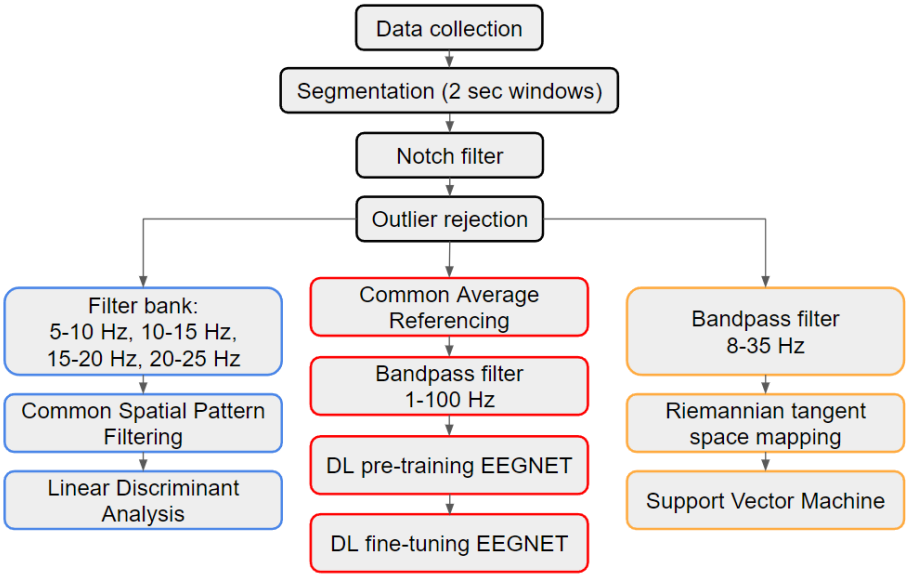


Fig. 13. Final implemented pipelines, after experimentation.

preferred over filtfilt as re-initialization at the start of each segment is not needed for SP. The phase shift introduced by SP was not considered less of a problem, as the oscillatory information of MI was not affected by the phase shift.

Following experiments showed that for both FBCSP as for RG, a second order SP filter slightly outperformed a third and fourth order SP filter (FBCSP: 64.4% to 63.3% and 64.2%, respectively, and RG: 56.7% to 56.3% and 56.6%, respectively). Here, results are very similar, indicating that this choice did not have much impact on final results.

Next, experiments were done for different frequency bands. For FBCSP, the 5-25 FB with steps of 5 performed highest (63.9%) when compared to 5-35Hz with steps of 5 (62.0%), and 4-40Hz with steps of 4 (62.8%). For RG, a 8-35Hz bandpass filter outperformed (59.8%) a 4-40Hz (59.1%) and 1-100Hz (57.1%). For DL, 8-35Hz performed highest (58.0%) when compared to 4-40Hz (56.4%) and 1-100Hz (54.0%). However, the 1-100Hz bandpass filter was chosen, as using this filter was shown to achieve higher cross-subject performance in initial cross-subject pre-training experiments. It should be noted that for each pipeline, the highest performing filter band differed much between subjects, which is in line with previous literature showing higher accuracy when filter bands are chosen per subject (e.g., [76]). Using individualized filter bands was not experimented with in the current study due to time restrictions.

Common spatial filtering For FBCSP, LDA outperformed (63.9%) RF (62.5%) and SVM (63.2%), and was chosen. As with the filter bands, the best performing

model differed much between individuals, and no clear explanation can be given why one model outperformed the others.

Riemannian geometry For RG, SVM (61.8%) outperformed RF (59.3%), LDA (59.7%) and MDM (59.8%), and was chosen.

Deep learning Regarding DTL hyperparameter search, experiments showed best performance for EEGNET with $lr : 0.001, fz : 8, D : 2, pdrop : 0.4$. Results are in line with the optimal parameter values found in the original EEGNET, with only *pdrop* differing from the 0.25 used in the original EEGNET [17]. The amount of data used in the current study is lower than used in the EEGNET study, which may explain that a higher *pdrop* resulted in better validation accuracy due to preventing overfitting of the model on the smaller amount of data used in the current study when compared to the EEGNET study. However, it should be noted that difference in performance between *pdrop* of 0.25 and 0.4 were minimal (49.7% and 50.0%, respectively).

On average, pre-trained models did not achieve a validation accuracy above 50%. However, pre-trained models with X02 or X05 as validation subject scored highest validation accuracy, with scores reaching around 70%. This can indicate that subjects X02 and X05 have more consistent, distinguishable data, making the training process smooth, while for other subjects, data is less consistent, making the training process noisy and less effective.

Regarding fine-tuning, results of the ablation study showed highest fine-tune performance when fine-tuning weights of the entire network starting from learned weights of the pre-trained model. All pre-trained models were fine-tuned, and results in the following section are from the model which achieved highest validation accuracy after fine-tuning.

5.3 Pipeline comparison

Results of comparing fine-tuned EEGNET using DTL to RG and FBCSP using either 1, 2, 3, or 4 trials as training set, are presented in Table 3. DTL outperformed FBCSP and RG for most of the subjects, scoring an overall average over subjects and trials of 58.7% compared to 54.7% and 54.1% for FBCSP and RG, respectively. Statistical significance of performance between DTL and the other models was determined with the Wilcoxon signed rank test. Differences in performance between DTL and RG, and DTL and FBCSP, were significant ($p = 0.014$, and $p = 0.020$, respectively), when comparing results averaged over all trials for each subject.

Looking into Table 3, we see that accuracy for subjects X02, X04, and X05 was considerably higher when compared to all other subjects, reaching 80%, while accuracy for other subjects did not exceed 60%. Performance of all models generally increased when more training trials were available, although the increase was not linear over trials for each subject. Sometimes, performance decreased (e.g., subject X03 DTL performance with 1 trial was higher (52.4%)

than with 4 trials (52.3%)). Next to this, average increase of performance over trials was lower than expected, often showing no more than a 5% increase from trial 1 to 4, which can also be seen by the low standard deviations of the average accuracy over trials, often being around 1%. Another thing to see is that generally when a model performs highest using 1 trial when compared to the other models, average performance of this model is also highest, showing consistency of model performance over trials.

Table 3. Average classification accuracy ± σ of 5 runs on test set, using 1, 2, 3 or 4 trials as training set, 1 trial as validation set, and 5 trials as test set. Averages of averages ± σ are presented for pipelines over trials, and for all trials over pipelines. Highest average accuracy ± σ for each subject is highlighted in bold.

		Number of training trials				
Subject	**Pipeline**	**1**	**2**	**3**	**4**	**Average**
	RG	44.2 ± 1.6	44.7 ± 0.5	44.5 ± 0.2	44.7 ± 0.1	44.5 ± 0.2
X01	FBCSP	43.7 ± 2.8	45.3 ± 0.8	45.1 ± 0.4	45.8 ± 0.2	**45.0 ± 0.8**
	DTL	39.3 ± 2.1	39.9 ± 1.2	39.7 ± 4.4	40.9 ± 3.6	40.0 ± 0.6
	RG	70.5 ± 3.0	70.6 ± 1.0	72.2 ± 0.7	73.5 ± 0.1	71.7 ± 1.2
X02	FBCSP	63.3 ± 1.8	68.6 ± 1.2	71.8 ± 1.0	74.3 ± 0.5	69.5 ± 4.1
	DTL	79.2 ± 4.9	78.8 ± 2.6	80.7 ± 2.8	80.8 ± 2.2	**79.9 ± 0.9**
	RG	40.4 ± 1.6	39.9 ± 1.1	39.9 ± 0.3	40.6 ± 0.1	40.2 ± 0.3
X03	FBCSP	42.7 ± 2.8	41.5 ± 1.6	42.3 ± 0.3	44.2 ± 0.7	42.7 ± 1.0
	DTL	52.4 ± 3.4	50.9 ± 2.4	52.6 ± 3.2	52.3 ± 3.8	**52.1 ± 0.7**
	RG	62.0 ± 1.5	64.9 ± 1.7	67.8 ± 0.6	69.2 ± 0.4	66.0 ± 2.8
X04	FBCSP	65.9 ± 1.3	69.9 ± 1.9	74.4 ± 0.9	76.3 ± 0.6	71.6 ± 4.0
	DTL	70.7 ± 2.9	79.7 ± 3.3	81.1 ± 1.2	80.4 ± 1.3	**78.0 ± 4.2**
	RG	79.9 ± 1.8	79.6 ± 0.6	81.3 ± 0.3	82.8 ± 0.3	80.9 ± 1.3
X05	FBCSP	74.2 ± 0.9	77.0 ± 1.1	79.4 ± 0.8	81.3 ± 0.2	78.0 ± 2.7
	DTL	83.0 ± 1.6	86.9 ± 1.0	87.0 ± 1.7	86.3 ± 2.0	**86.6 ± 1.6**
	RG	53.0 ± 1.4	53.9 ± 0.3	54.1 ± 0.6	56.3 ± 0.6	54.3 ± 1.2
X06	FBCSP	52.1 ± 1.3	55.0 ± 0.6	55.4 ± 0.3	56.5 ± 1.6	54.8 ± 1.6
	DTL	57.2 ± 4.0	57.9 ± 6.4	57.0 ± 6.6	63.9 ± 1.7	**59.0 ± 2.8**
	RG	36.4 ± 1.6	38.1 ± 0.6	39.5 ± 0.2	39.9 ± 1.0	**38.5 ± 1.4**
X07	FBCSP	36.1 ± 2.5	35.8 ± 0.6	38.1 ± 0.6	39.0 ± 1.0	37.3 ± 1.3
	DTL	35.4 ± 3.7	36.0 ± 2.3	38.9 ± 3.2	40.4 ± 2.0	37.7 ± 2.1
	RG	51.0 ± 2.0	49.0 ± 0.3	50.1 ± 0.5	51.9 ± 0.3	50.5 ± 1.1
X08	FBCSP	52.1 ± 2.5	51.2 ± 0.9	52.7 ± 0.3	54.1 ± 0.6	52.5 ± 1.1
	DTL	49.5 ± 4.2	50.9 ± 3.0	56.7 ± 4.6	59.5 ± 1.6	**54.2 ± 4.1**
	RG	43.7 ± 3.1	41.4 ± 0.5	40.6 ± 0.8	39.8 ± 0.0	41.4 ± 1.5
X09	FBCSP	42.2 ± 3.6	39.1 ± 0.7	39.4 ± 0.2	39.3 ± 0.2	40.0 ± 1.3
	DTL	40.5 ± 5.7	41.7 ± 2.9	42.5 ± 3.2	42.8 ± 1.8	**41.9 ± 0.9**
	RG	53.4 ± 13.8	53.3 ± 14.3	54.3 ± 14.9	55.3 ± 15.4	54.1 ± 14.6
Average	FBCSP	52.5 ± 12.0	54.0 ± 13.8	55.6 ± 14.9	56.9 ± 15.5	54.7 ± 14.0
	DTL	56.4 ± 16.6	58.1 ± 18.0	59.6 ± 17.8	60.8 ± 17.2	**58.7 ± 17.3**

5.4 Side experiment dry electrodes

Results of model performance for X02 using either data captured by wet or dry electrodes is presented in Table 4. Performance for dry electrodes follows the same trend as for wet electrodes, with DTL outperforming (67.9%) FBCSP (60.8%) and RG (61.0%), but absolute performance drops by around 10% for almost each configuration of trials and models.

Table 4. Comparison between model performance, as conducted in Table 3, using data of X02 captured by either wet, or dry electrodes.

		Number of training trials				
Subject	**Pipeline**	**1**	**2**	**3**	**4**	**Average**
X02 wet	RG	70.5 ± 3.0	70.6 ± 1.0	72.2 ± 0.7	73.5 ± 0.1	71.7 ± 1.2
	FBCSP	63.3 ± 1.8	68.6 ± 1.2	71.8 ± 1.0	74.3 ± 0.5	69.5 ± 4.1
	DTL	79.2 ± 4.9	78.8 ± 2.6	80.7 ± 2.8	80.8 ± 2.2	**79.9 ± 0.9**
X02 dry	RG	62.8 ± 0.5	57.9 ± 1.5	60.4 ± 1.3	62.8 ± 0.5	61.0 ± 2.0
	FBCSP	63.4 ± 0.8	55.3 ± 2.0	61.1 ± 1.2	63.4 ± 0.8	60.8 ± 3.3
	DTL	63.5 ± 5.7	67.5 ± 3.7	70.5 ± 5.5	70.0 ± 3.8	**67.9 ± 2.8**

6 Methods closed-loop

6.1 Experiment design

Subjects Subjects X01, X02, X03, and X04 volunteered to participate in the closed-loop experiments. Subjects were chosen based on open-loop performance to have subjects with previous high performance (X02, X04), medium performance (X03), and low performance (X01), to assess results for all groups.

Overview Each experiment was repeated on 3 consecutive days, to assess the effect of practice on the performance, and to measure the development of fine-tune performance of the DL model when more inter-subject, cross-session data became available.

A session consisted of environmental noise capture, a motor execution practice trial, a MI practice trial, 3 open-loop trials, 5 pseudo-closed-loop trials, and 3 closed-loop trials in a game environment. A session lasted for around 75 minutes, including the 10 minutes needed for the set-up of the cap.

The same procedure as described in Section 4.1 was used for environmental noise capture, the practice trials, and open-loop trials.

Experiment design for the pseudo-closed-loop trials followed nearly the same pattern as for open-loop trials, but with two main differences. Firstly, as only left arm, right arm, and relax were chosen as classes, legs MI was left out. Second, real-time feedback was given to the user during the MI tasks. Inference using the DTL was used to make real-time predictions. Feedback was given in the form of replacement of the grey dot. The dot moved to the right when right arm MI was predicted, and to the left for left arm MI. The dot increased in size when relax was predicted. In an earlier study, it was found that only giving feedback for correct predictions resulted in better subject performance of MI compared to giving both feedback for correct as for incorrect predictions [77]. Therefore, during a task, the dot only moved for correct predictions (e.g., to the left for a left arm MI task), and did not move for incorrect predictions.

For the closed-loop game trials, the game Dodge was used. Dodge is a game in which the player moves a small block horizontally to avoid hitting bigger blocks falling from above. A screenshot of the game is given in Figure 14. Real-time predictions of the MI-BCI were used as input for moving the player block. Here, left arm MI was used to move the player block to the left, right arm MI to move the player block to the right, and relax to stop the player block from moving. The background color was set to black, and colors of the player block and other blocks were set to gray, to have a similar environment in terms of color compared to the earlier trials. To work with the update of predictions every 0.5 seconds, positions of the player block and other blocks were only updated every 0.5 seconds. The player started with a score of 0. For each avoided block, the player earned one point. The game trial was over when the player hit one of the blocks, or hit the border at the left or right of the screen. At the start of the game, the player could get used to moving the player block for 5 seconds, before blocks started falling down.

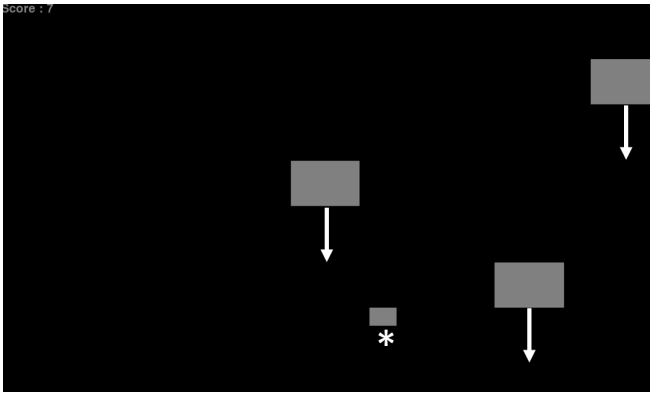


Fig. 14. A screenshot of the game Dodge. The player block is located at the bottom (indicated by *). Movement direction of the blocks is indicated by arrows.

After the last session, the following questions were asked regarding the subjective experience of the experiments, and their satisfaction of the system:

- On a scale of 1 to 10, how satisfied were you...
  - With the feeling of control during closed-loop trials?
  - With the wearing comfort of the Unicorn BCI cap?
  - With the feeling of control during the game?
- What was the main inconvenience of the system?

Task instructions The same instructions as given during open-loop experiments were given for the MI tasks. Further, during the closed-loop trials, subjects were instructed to limit shifting their focus to the results of the real-time feedback, to limit the influence of changes in the EEG signals due to ERPs, which were not present in the training data, and could therefore negatively influence the predictions [57]. Before the closed-loop game trials, the game was explained and the control of the player block using MI was explained, and one example game was shown.

6.2 Experimental setup

For each subject, a comparison was made between two methods of fine-tuning. For the first method, the fine-tuned model with highest performance for open-loop data was used (see Appendix, Section 10.2), which then was continuously fine-tuned and saved again for data of new sessions (ConFT), visible in Figure 15a. For the second method, the general pre-trained model with highest performance for open-loop data was used (see Appendix, Section 10.2), and was only fine-tuned for the current session (GenFT), visible in Figure 15b. For both methods, fine-tuning was done by updating all weights of the network, as was

done for open-loop experiments. Using ConFT gives an indication about difference in model performance when more subject-specific, but cross-session, data is available. After experiments were finished, ConFT and GenFT were compared to FBCSP and RG using the open-loop data.

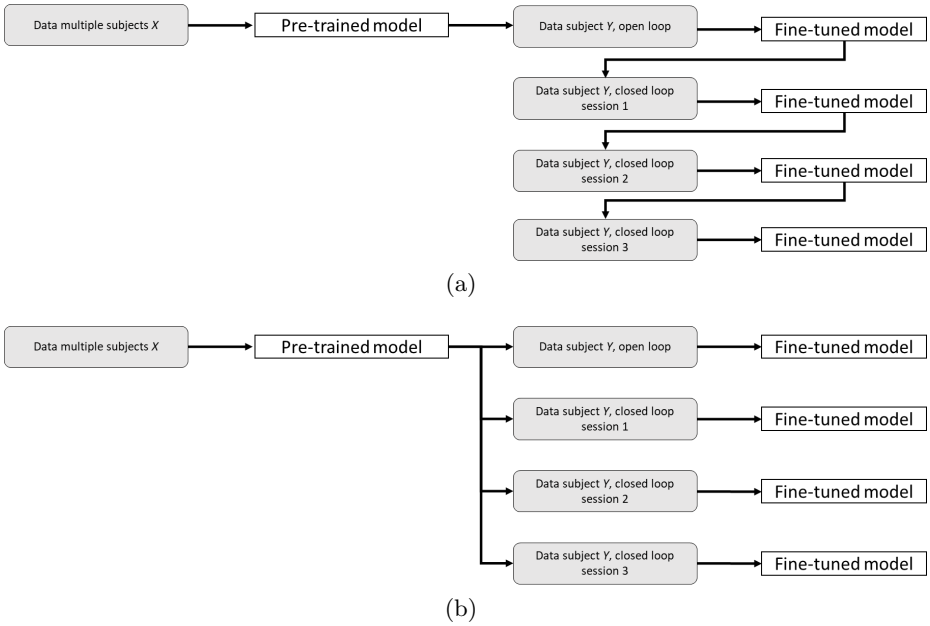


Fig. 15. Two methods of fine-tuning for data of new sessions of the same subject. **a)** ConFT: the fine-tuned model is continuously fine-tuned and saved for data of new sessions. **b)** GenFT: the model is fine-tuned starting from parameters of the general pre-trained model.

For each session, the model with highest fine-tune performance was used for inference during closed-loop trials of that session. Accuracy performance of closed-loop trials was calculated by comparing model predictions to the presented cues as ground truth labels. Regarding the closed-loop game trials, no ground truth labels were available. Here, performance was assessed by the game score achieved at the end of the trial.

7 Results closed-loop

7.1 Open-loop fine-tuning

Results of fine-tuning GenFT and ConFT, together with closed loop results, are presented in Figure 16. ConFT outperformed GenFT for almost all sessions of each subject, averaging 63.42% accuracy versus 60.94% for GenFT. Using the Wilcoxon signed rank test, differences were found to be significant ($p = 0.026$), when comparing results for all sessions of all subjects. As first experiments all showed superiority for ConFT compared to GenFT, it was decided to always choose ConFT as model for closed-loop trials for consistency.

Model performance of ConFT did not show a clear pattern of increase over sessions. Thus, feeding more inter-session data to the model does not necessarily improve performance, which can be explained by inter-session variability of the data. The fact that GenFT also did not show consistent increase of performance over sessions indicates that the effect of learning the task by the subjects was low.

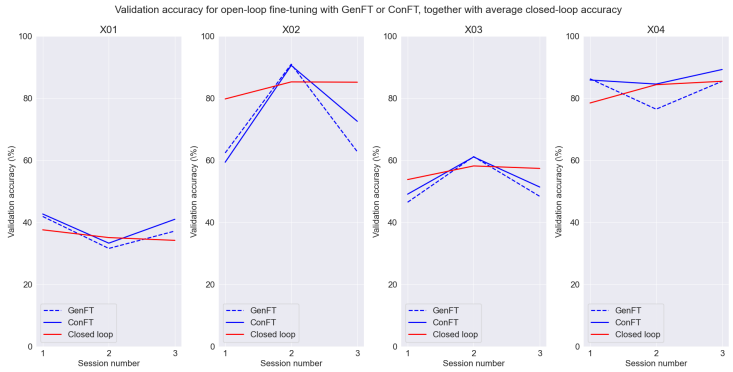


Fig. 16. Fine-tune validation accuracies of GenFT and ConFT for 3 open-loop trials, together with average accuracy of 5 closed-loop trials.

7.2 Closed-loop

As shown in Figure 16, closed-loop performance was lowest for X01, X03 scored medium, and X02 and X04 showed highest performance. Closed-loop performance increased between session 1 and 2 for subjects X02, X03, and X04, but not for X01, and was only significant for X02 ($p = 0.03$) as found with a Wilcoxon signed rank test. Performance of all subjects did not differ significantly between session 2 and 3.

Closed-loop performance was similar when compared to open-loop fine-tune performance. High closed-loop performance indicated that differences between

data from open-loop and closed-loop experiments was low. Closed-loop performance even showed a close to 20% increase in session 1 of subject X02, indicating that subjects may have performed with more attention during the closed-loop experiments. Subjects also reported to feel more alert and motivated when positive feedback was given, which gave them confirmation of doing MI correctly, as compared to open-loop experiments where no feedback was given.

Confusion matrices of closed-loop trials, presented in Figure 17, reveal a low number of true positives for the relax class for subjects X01 and X03, when compared to right arm MI and left arm MI. For subjects X02 and X04, class predictions were more evenly distributed when compared to subjects X01 and X03. The relatively low amount of correct predictions for relax when compared to left arm MI and right arm MI indicates that performing the relax task does not result in consistent brain activity across segments, causing the model to have troubles correctly predicting this class.

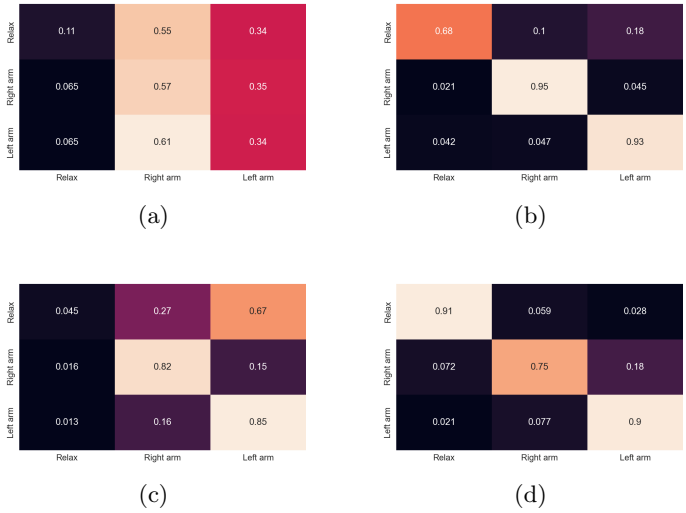


Fig. 17. Confusion matrices from predictions of ConFT for all closed-loop trials for **a)** X01, **b)** X02, **c)** X03, **d)** X04. True labels are presented on the vertical axis, and predictions on the horizontal axis. Values in the matrices are normalized for true labels.

7.3 Game trials

Game results, presented in Table 5, show lowest performance for X01, and highest performance for X04. Performance of X02 and X03 were similar. Average game performance increased between session 1 and 2 for all subjects, but decreased between session 2 and 3 for subjects X02, X03, and X04, only increasing for X01.

The order of performance of the subjects during closed-loop trials did not necessarily correlate with the order of performance during game trials, with subject X03 performing higher than X01 and X04 during session 1, and higher than X02 during session 2. As playing the game relies on many factors including but not limited to game tactics, the randomness of the blocks falling, and the need to constantly switch between the mental states, the difference in results between game trials and earlier experiments can be attributed to any of these factors. Here, the lower performance of X03 in open-loop and closed-loop experiments when compared to X02 and X04 was mostly due to low accuracy of the relax class (visible in the confusion matrices in Figure 17). During the game, using relax was not necessary, as the player could survive by constantly moving to the left or right.

During session 3, subjects X02 and X03 both reported to feel tired. The effect of the tiredness was already visible in the closed-loop experiments of session 3, and was also visible by their low performance during the game trials.

Table 5. Game scores of playing the game 3 times in each session, together with the average for each session.

	Session 1				**Session 2**				**Session 3**			
Subject	**1**	**2**	**3**	**Average**	**1**	**2**	**3**	**Average**	**1**	**2**	**3**	**Average**
X01	0	0	0	0.0	4	9	1	4.7	2	12	1	5.0
X02	0	9	2	3.7	22	10	21	17.7	8	7	6	7.0
X03	0	2	19	7.0	11	22	27	20.0	9	3	6	6.0
X04	0	0	9	3.0	19	10	35	21.3	30	0	27	19.0

7.4 Subjective experience

Results of subjective experience are presented in Table 6. Comparing the subjective scores to the performance, the feeling of control directly corresponds to the achieved scores, where X04 reported the highest feeling of control and had highest performance, followed by X02, X03, and lastly X01 in same order of their performance scores.

Table 6. Subjective experience during closed-loop trials and game trials.

Question	**X01**	**X02**	**X03**	**X04**
On a scale of 1 to 10, how satisfied were you:				
a) With the feeling of control during closed-loop trials?	6	8	7	9
b) With the wearing comfort of the Unicorn BCI cap?	8.5	8	9	10
c) With the feeling of control during the game?	7	8	8	9
Which was the main inconvenience of the system?	Use of gel	Restriction of movement	Data collection delay	Data collection delay

7.5 Post-experiment model comparison

Post-experiment comparison between fine-tuning ConFT and GenFT with training of FBCSP and RG are presented as tabular form in Table 7. Contrary to earlier open-loop experiments, FBCSP (68.5% ± 16.5) outperformed ConFT for most of the subjects and sessions, although the difference was not significant ($p = 0.09$). For subject X01, differences are most notable, with GenFT and ConFT performing around 40%, while FBCSP and RG score around 50 to 60%.

Table 7. Average validation accuracy ± σ for each session for 3 open-loop trials, using leave-one-trial-out cross-validation. Highest accuracy per configuration of subject and session is highlighted in bold.

Subject	Session	FBCSP	RG	GenFT	ConFT
	1	**50.0 ± 11.5**	44.9 ± 1.0	41.9 ± 3.2	42.7 ± 3.0
X01	**2**	56.4 ± 17.3	**57.3 ± 10.7**	31.6 ± 1.5	33.3 ± 2.2
	3	60.7 ± 7.1	**62.0 ± 4.9**	37.2 ± 0.1	41.0 ± 9.2
	1	**69.2 ± 4.6**	59.4 ± 4.0	62.4 ± 1.5	59.4 ± 0.7
X02	**2**	**93.9 ± 1.7**	84.7 ± 9.8	91.0 ± 0.0	90.6 ± 0.7
	3	60.8 ± 15.4	58.2 ± 0.0	62.8 ± 1.3	**72.6 ± 0.7**
	1	36.1 ± 1.7	40.0 ± 6.0	46.5 ± 14.4	**49.1 ± 13.7**
X03	**2**	**68.9 ± 7.9**	53.6 ± 4.0	61.2 ± 4.8	61.1 ± 8.2
	3	**67.8 ± 1.6**	42.6 ± 3.4	48.4 ± 10.8	51.4 ± 9.4
	1	**88.0 ± 8.9**	78.6 ± 5.4	86.3 ± 0.7	85.9 ± 0.0
X04	**2**	**86.3 ± 5.8**	82.5 ± 4.0	76.5 ± 0.7	84.6 ± 0.0
	3	84.2 ± 2.6	87.2 ± 4.2	85.5 ± 0.7	**89.3 ± 1.5**

8 Discussion

In this study, performance of decoding the mental states relax, left arm MI, and right arm MI, using a BCI system with a low-cost EEG device was investigated. A cross-subject DTL approach with fine-tuning was proposed using EEGNET, which performed significantly higher for open-loop experiments when compared to FBCSP and RG pipelines. Performance of DTL was also investigated for closed-loop experiments using two methods of fine-tuning, where ConFT significantly outperformed GenFT. Additionally, subjects played a game of Dodge using the BCI system, and reported positively regarding the feeling of control during the game.

8.1 Open-loop

Performance of DTL was compared with FBCSP and RG pipelines for low-resource circumstances, having only 1 to 4 trials of each 3 minutes as training data. Open-loop accuracy averaged over trials and subjects for DTL was 58.7%, for FBCSP 54.7%, and for RG 54.1%. The finding that DTL outperforms FBCSP has also been shown in earlier work [55]. Performance generally increased when the amount of training data increased, with an average DTL performance using 1 trial of 56.4% increasing to 60.8% using 4 trials. This finding was also found in earlier work of DTL in ERP-based BCIs, where fine-tuning on a bigger subject-specific set of data also resulted in higher performance when compared to fine-tuning on a smaller set of data [78]. Performance differed much between subjects, with one subject scoring an accuracy for open-loop of 86.6%, and other subjects performing as low as 37.7%. These results are in line with earlier work, as inter-subject variability is a well-known phenomenon in the BCI field [15]. Results of the small side-experiment using dry electrodes with subject X02 showed same performance trends for FBCSP, RG, and DTL when compared to using wet electrodes, but absolute performance dropped by around 10%. This trend in performance drop was also shown in a study of decoding reach-and-grasp actions, were gel-based classification accuracy was 61.3%, compared to 56.4% for dry electrodes [79].

Above results indicate that by cross-subject pre-training, a DL model is able to learn general discriminative features which can be exploited by fine-tuning on a small dataset of a specific subject, outperforming FBCSP and RG trained only on the available training data of the specific subject. Thus, DTL can be an effective method to reduce calibration time. Moreover, these results indicate that MI decoding of relax, right arm MI, and left arm MI is possible using a low-cost EEG device, with only 8 electrodes.

However, 6 out of 9 subjects achieved an accuracy lower than 70%, thereby having performance below the threshold for feasible usage of the BCI system [24]. Various factors can influence the performance of the system used in current study. Here, the most notable factor is the use of only 8 electrodes, as it has previously been shown that decoding performance of reach-and-grasp actions

increases from 54.0% when 5 electrodes were used, to 66.9% when 61 electrodes were used [30].

8.2 Closed-loop

For closed-loop experiments, it was shown that ConFT almost always outperformed GenFT, and had significantly higher average accuracy across subjects and sessions when compared to GenFT (63.42% $\pm$ 0.026 versus 60.94% $\pm$ 0.014, $p = 0.026$). These results are comparable to earlier work, where a closed-loop BCI system was developed using an adaptive LDA model which retrains for new data over multiple sessions, which outperformed a LDA model trained solely on data of the current session [80]. Contrary to open-loop experiment results, FBCSP performed highest (68.5% $\pm$ 16.5) for almost all subjects and sessions when compared to RG (62.6 $\pm$ 16.1), ConFT, and GenFT. Regarding closed-loop performance in session 1, accuracy was similar when compared to earlier open-loop experiments, contrary to earlier work, where often a decrease in performance is reported (e.g., [8, 81]). Model performance of ConFT did not show a clear pattern over sessions, while for closed-loop performance of subjects X02, X03, and X04, performance increased between session 1 and 2, but not between session 2 and 3. This trend corresponds with game performance, where highest scores for subjects X02, X03, and X04 were achieved in session 2. For subject X01, performance stayed constantly low across all sessions. In other work where 3 sessions were also done, performance did increase for each new session, although not significant [82]. The subjective experience of the users regarding control during closed-loop experiments and the comfort of the cap, was generally positive, with main inconveniences being the use of gel, the restriction of movement, and the data collection delay. Both the use of gel and the delay have been mentioned in earlier work as inconveniences of the BCI systems [29, 72].

For most subjects and sessions, ConFT outperformed GenFT. However, for some cases, GenFT showed higher performance. Multiple underlying mechanisms can influence these differences. One well-known mechanism is the fact that humans and computers learn to adapt to each other in different ways, called the two-learner problem [83]. When the user is adapting to the BCI by learning how to change his way of performing MI, this new data might change the way the model makes predictions, and this in turn can lead to different feedback which can be confusing for the user [25]. When either the user or model adapt too quickly, performance drops [83]. Another factor which influences model learning, is the variability of brain activity of the subject between sessions [15]. If data between sessions differs too much, the model might need to move to completely different minima than the minima it was previously converged to. Next to this, levels of attention of the subject during the experiments can greatly influence the performance as well. The level of attention is a well-known topic in the BCI field, and experiments are done taking into account the level of attention (e.g., by not predicting the positive class when attention levels are low) [8]. In the current study, subjects X02, X03, and X04 all reported to feel tired during session 3, which can indicate low attention levels, thereby giving a cause for the

irregularities in performance. For subject X01, performance stayed low across all sessions.

Closed-loop performance was not always lower when compared to open-loop performance, contrary to earlier reported results (e.g., [8, 56, 81]). All subjects did report to feel more alert and excited during closed-loop trials due to given feedback. This alertness and excitement may have risen the attention levels, possibly explaining the high closed-loop performance. Moreover, given feedback was minimal, and feedback was only given for correct predictions, limiting the level of frustration and distraction of the subject. This type of feedback has previously been reported to lead to higher performance compared to giving feedback for incorrect predictions as well [77]. In [56], the setup of experiments was similar to the current study, but feedback was given for correct and incorrect predictions, possibly explaining the drop of performance between closed-loop and open-loop. In studies regarding external control (e.g., lower-limb exoskeletons in [8, 81]), feedback during closed-loop trials is also given for both correct and incorrect predictions. Moreover, all predictions trigger a change of state of the exoskeleton. Thereby, an incorrect prediction can cause movement of the exoskeleton, which can distract and frustrate the user to an even higher level when compared to feedback presented on a computer screen as in the current study and [56].

The feeling of control for subjects X01 and X03 was higher during the game than during the closed-loop trials, while ratings between game and closed-loop trials were the same for X02 and X04. This can be explained by observing the class prediction distribution in the confusion matrices in Figure 17. The relax class was badly represented for subjects X01 and X03, causing frustration when relax was not predicted during the closed-loop trials. However, the relax class was not necessary during the game, as one could keep moving left to right using the MI classes, causing less frustration and thus a higher feeling of control. For subjects X02 and X04, all classes were well represented, and thus there was no big difference between closed-loop trials and the game.

8.3 Limitations of pipeline choices

During this project, various choices were made regarding the pipelines and model learning procedures. For future work, some improvement can be made regarding these choices. Early on, it was decided to use a window size of 2 seconds. Experiments showed superior performance for 2 second windows when compared to 1.5, or 1 second windows. However, during closed-loop experiments, subjects X03 and X04 reported that the delay caused by initial data collection was the main inconvenience of the system. As the first two seconds of the MI tasks were not used for data collection due to possible ERPs in the data, and after that another 2 seconds were needed for data collection, the total initial delay was 4 seconds. Besides the frustration for the user caused by this delay, earlier work also found that during a 4 second MI task, MI classification for their BCI system performed higher for the first 2 seconds of the MI task than the last two seconds [84]. Thus, for future work, smaller window sizes and a smaller window for ERP should be investigated to reduce the initial delay of the system.

Besides the window size, early in the project the window hop was chosen to be 0.5 seconds, to ensure processing overhead for later closed-loop experiments would not occur. However, during closed-loop experiments, the processing overhead was found to be lower than 0.1 seconds. A window hop of 0.5 seconds adds a small delay between thought and prediction. Moreover, the frequency of predictions is only 2 times per second, thereby making playing the game less enjoyable due to a frame rate of only 2Hz. Due to time restrictions of this project, a smaller window hop was not implemented. For future work, this window hop can easily be decreased.

For fine-tuning the DL models for closed-loop trials, 2 trials were chosen for training data, and 1 trial as validation data. It can be derived from open-loop experiments that using more trials for training and validation would have increased model performance. However, this also would have increased training session time. Here, future work could try to find optimal balance between calibration time, and performance.

For the DL pipeline, an architecture adapted from EEGNET was used [17]. Multiple extensions to EEGNET have been proposed in recent years, with the most notable extension the addition of inception layers [18, 22, 85]. These extensions to EEGNET were not considered in the current study, as preliminary results showed lower performance when compared to EEGNET. However, this lower performance can be attributed to the higher complexity of these models, while dataset size of the current study was low. When more data is available, future studies should investigate performance of above extensions to EEGNET.

Post-experiment analysis of closed-loop experiments indicates that higher closed-loop performance could have been reached using FBCSP, contrary to results with the same subjects in the open-loop experiments. The superiority of FBCSP changed interpretation of results. Here, scores from subject X01 using GenFT (36.9% $\pm$ 4.2) and ConFT (39.0% $\pm$ 4.1) indicated that subject X01 belonged to the population which are not able to produce brain activity robust enough to be detected by the BCI system, as mentioned before [86]. However, scores observed from FBCSP (55.7% $\pm$ 4.4) and RG (54.7% $\pm$ 7.2) indicate that data of X01 is better distinguishable using other pipelines, and DTL may not be suitable for every subject. This model variability is a point of concern regarding practicability of the system, as this implies that for every subject and session, model comparison needs to be done. Ensemble learning could be a possible solution for this problem. With ensemble learning, all models are trained, and for inference during closed-loop, a majority vote is made between the predictions of all models. This technique has been implemented recently for upper-limb movement versus non-movement detection using EEG, where an ensemble model of normal SVM, a RG-TG SVM, and EEGNET, and showed superior results to using each of the techniques separately [87].

In the current study, not much is known about the underlying mechanisms, such as feature selection, of the implemented EEGNET. An extension to EEGNET was recently proposed, together with a method for visualizing topographical patterns learned by the network [85]. Due to time restrictions, visualisation of

underlying mechanisms of EEGNET in the current study was not implemented. For future work, a comparison between learned patterns of the DL model with FBCSP and RG models could improve interpretability, and acceptance, of DL models in the BCI field. Moreover, interpretability can contribute in research regarding the two-learner problem [83], as detailed feedback about predictions can help the user adapt to the model and can reduce confusion for the user when predictions are incorrect.

8.4 Extensions towards everyday usage

Although results of the BCI system in the current study are promising, there is a long way to go to everyday usage. Firstly, the current approach should be extended towards control of upper-limb exoskeletons. Moving an exoskeleton changes the usage environment from static towards dynamic, thereby introducing movement-related artifacts in the EEG signals [8, 81]. Thus, in the future the pipeline should be re-evaluated for external control. Additionally, useful control of such exoskeleton involves more degrees of freedom than the 3-class problem of left arm MI, right arm MI, and relax proposed in the current study, and MI of more complex actions should be decoded. Recent work shows promising results for decoding of reach-and-grasp tasks using EEG (e.g., [30, 31]. The question here is if decoding of more complex tasks is possible with a low-cost device with 8 electrodes as used in the current study, as in [30], performance dropped by 15% when 5 instead of 61 electrodes were used.

Another factor towards efficient everyday usage is the preparation time. The approach in the current study already reduces preparation times by reducing setup, cleanup, and calibration times when compared to studies with more electrodes and traditional ML methods. However, preparation time can be further shortened by using dry electrodes. In the current study, the small side-experiment showed that although with lower accuracy, MI decoding using dry electrodes is possible, and future work should keep investigating the potential of dry electrodes.

Lastly, experiments in the current study were done with healthy subjects. A main goal of current BCI systems is to help rehabilitation of stroke and spinal cord injury patients. It has been found that the spatial filters learned by CSP for EEG data of stroke patients change during the rehabilitation period [88]. Here, an increase of assigned importance by the CSP filters was found in the hemispheres affected by the stroke, while other areas remain unchanged. These changes in EEG data over time also underline the importance of adaptive classifiers. Directly comparing healthy subjects to stroke patients has been reported to be impractical, as often only a low number of stroke patients participate in BCI research, and experimental setups between studies vary greatly [7]. Therefore, future work should focus on adapting the current approach for assisting in rehabilitation of stroke patients, and the results should be compared to healthy subjects to learn more about the differences between these groups.

9 Conclusion

In this study, various BCI pipelines were investigated for upper-limb MI using a low-cost EEG device with low preparation and calibration times. Firstly, a comparison was made between DTL, FBCSP, and RG pipelines, for developing an open-loop MI-BCI for decoding relax, right arm MI, and left arm MI. It was shown that a DTL pipeline with cross-subject pre-training followed by subject specific fine-tuning, had higher average accuracy across subjects (58.7%) when compared to subject specific FBCSP (54.7%) and RG (54.1%) for low-data circumstances of open-loop MI experiments of 9 subjects. These differences were significant ($p = 0.014$, and $p = 0.020$, respectively). The second goal of this study was to further investigate performance of DTL for closed-loop experiments. The average accuracy across subjects and sessions was significantly higher ($p = 0.026$) for ConFT (63.42% $\pm$ 0.026) when compared to GenFT (60.94% $\pm$ 0.014). Contrary to the open-loop experiment results, the post-experiment analysis showed that fine-tune performance of ConFT was outperformed by FBCSP (68.5% $\pm$ 16.5), although the difference was not significant ($p = 0.09$). Results of the closed-loop trials revealed that 2 out of 4 subjects achieved high performance ($> 80\%$), and 3 out of 4 subjects reported to feel in control while playing a game using the BCI system. Results of the current study show a step forward towards a more practicable and lower cost BCI system for the general public. As a next step, ensemble learning should be investigated to overcome the model performance variability found in the current study.

References

1. W WHO. The top 10 causes of death. *World Health Organization*, 2018.
2. Jingyi Liu, Muhammad Abd-El-Barr, and John H Chi. Long-term training with a brain-machine interface-based gait protocol induces partial neurological recovery in paraplegic patients. *Neurosurgery*, 79(6):N13–N14, 2016.
3. Alexander A Frolov, Olesya Mokienko, Roman Lyukmanov, Elena Biryukova, Sergey Kotov, Lydia Turbina, Georgy Nadareyshvily, and Yulia Bushkova. Post-stroke rehabilitation training with a motor-imagery-based brain-computer interface (bci)-controlled hand exoskeleton: a randomized controlled multicenter trial. *Frontiers in neuroscience*, 11:400, 2017.
4. Marcel Van Gerven, Jason Farquhar, Rebecca Schaefer, Rutger Vlek, Jeroen Geuze, Anton Nijholt, Nick Ramsey, Pim Haselager, Louis Vuurpijl, Stan Gielen, et al. The brain–computer interface cycle. *Journal of neural engineering*, 6(4):041001, 2009.
5. Gernot R Müller-Putz, Patrick Ofner, Joana Pereira, Andreas Pinegger, Andreas Schwarz, Marcel Zube, Ute Eck, Björn Hessing, Matthias Schneiders, and Rüdiger Rupp. Applying intuitive eeg-controlled grasp neuroprostheses in individuals with spinal cord injury: Preliminary results from the moregrasp clinical feasibility study. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 5949–5955. IEEE, 2019.
6. Valeria Mondini, Reinmar J Kobler, Andreea I Sburlea, and Gernot R Müller-Putz. Continuous low-frequency eeg decoding of arm movement for closed-loop, natural control of a robotic arm. *Journal of Neural Engineering*, 17(4):046031, 2020.
7. Yongtian He, David Eguren, José M Azorín, Robert G Grossman, Trieu Phat Luu, and Jose L Contreras-Vidal. Brain–machine interfaces for controlling lower-limb powered robotic systems. *Journal of neural engineering*, 15(2):021004, 2018.
8. Laura Ferrero, Vicente Quiles, Mario Ortiz, Eduardo Iáñez, and José M Azorín. A bmi based on motor imagery and attention for commanding a lower-limb robotic exoskeleton: A case study. *Applied Sciences*, 11(9):4106, 2021.
9. Gelu Onose, Cristian Grozea, Aurelian Anghelescu, Cristina Daia, Crina Julieta Sinescu, Alexandru Vlad Ciurea, Tiberiu Spircu, A Mirea, I Andone, A Spânu, et al. On the feasibility of using motor imagery eeg-based brain–computer interface in chronic tetraplegics for assistive robotic arm control: a clinical test and long-term post-trial follow-up. *Spinal cord*, 50(8):599–608, 2012.
10. Han Yuan and Bin He. Brain–computer interfaces using sensorimotor rhythms: current state and future perspectives. *IEEE Transactions on Biomedical Engineering*, 61(5):1425–1435, 2014.
11. Marc Jeannerod. Mental imagery in the motor context. *Neuropsychologia*, 33(11):1419–1432, 1995.
12. Alyssa M Batula, Jesse A Mark, Youngmoo E Kim, and Hasan Ayaz. Comparison of brain activation during motor imagery and motor movement using fnirs. *Computational intelligence and neuroscience*, 2017, 2017.
13. Christoph Stippich, Henrik Ochmann, and Klaus Sartor. Somatotopic mapping of the human primary sensorimotor cortex during motor imagery and motor execution by functional magnetic resonance imaging. *Neuroscience letters*, 331(1):50–54, 2002.
14. Mamunur Rashid, Norizam Sulaiman, Anwar PP Abdul Majeed, Rabiu Muazu Musa, Bifta Sama Bari, Sabira Khatun, et al. Current status, challenges, and possible solutions of eeg-based brain-computer interface: a comprehensive review. *Frontiers in neurorobotics*, page 25, 2020.

15. Yalda Shahriari, Theresa M Vaughan, LM McCane, Brendan Z Allison, Jonathan R Wolpaw, and Dean J Krusienski. An exploration of bci performance variations in people with amyotrophic lateral sclerosis using longitudinal eeg data. *Journal of neural engineering*, 16(5):056031, 2019.
16. Robin Tibor Schirrmeister, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and Tonio Ball. Deep learning with convolutional neural networks for eeg decoding and visualization. *Human brain mapping*, 38(11):5391–5420, 2017.
17. Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. *Journal of neural engineering*, 15(5):056013, 2018.
18. Mouad Riyad, Mohammed Khalil, and Abdellah Adib. Mi-eegnet: A novel convolutional neural network for motor imagery classification. *Journal of Neuroscience Methods*, 353:109037, 2021.
19. Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.
20. Fatemeh Fahimi, Zhuo Zhang, Wooi Boon Goh, Tih-Shi Lee, Kai Keng Ang, and Cuntai Guan. Inter-subject transfer learning with an end-to-end deep convolutional neural network for eeg-based bci. *Journal of neural engineering*, 16(2):026007, 2019.
21. Chuanqi Tan, Fuchun Sun, and Wenchang Zhang. Deep transfer learning for eeg-based brain computer interface. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 916–920. IEEE, 2018.
22. Ce Zhang, Young-Keun Kim, and Azim Eskandarian. Eeg-inception: an accurate and robust end-to-end neural network for eeg-based motor imagery classification. *Journal of Neural Engineering*, 18(4):046014, 2021.
23. Ruilong Zhang, Qun Zong, Liqian Dou, Xinyi Zhao, Yifan Tang, and Zhiyu Li. Hybrid deep neural network using transfer learning for eeg motor imagery decoding. *Biomedical Signal Processing and Control*, 63:102144, 2021.
24. Andrea Kübler, Nicola Neumann, Jochen Kaiser, Boris Kotchoubey, Thilo Hinterberger, and Niels P Birbaumer. Brain-computer communication: self-regulation of slow cortical potentials for verbal communication. *Archives of physical medicine and rehabilitation*, 82(11):1533–1539, 2001.
25. Fabien Lotte, Laurent Bougrain, Andrzej Cichocki, Maureen Clerc, Marco Congedo, Alain Rakotomamonjy, and Florian Yger. A review of classification algorithms for eeg-based brain–computer interfaces: a 10 year update. *Journal of neural engineering*, 15(3):031005, 2018.
26. Gernot R Müller-Putz, Ursula Tunkowitsch, Randall K Minas, Alan R Dennis, and René Riedl. On electrode layout in eeg studies: A limitation of consumer-grade eeg instruments. In *NeuroIS Retreat*, pages 90–95. Springer, 2021.
27. Piotr Szczuko. Real and imaginary motion classification based on rough set analysis of eeg signals for multimedia applications. *Multimedia Tools and Applications*, 76(24):25697–25711, 2017.
28. Alexander Craik, Yongtian He, and Jose L Contreras-Vidal. Deep learning for electroencephalogram (eeg) classification tasks: a review. *Journal of neural engineering*, 16(3):031001, 2019.
29. Eduardo López-Larraz, Fernando Trincado-Alonso, Vijaykumar Rajasekaran, Soraya Pérez-Nombela, Antonio J Del-Ama, Joan Aranda, Javier Minguez, Angel Gil-Agudo, and Luis Montesano. Control of an ambulatory exoskeleton with a

brain–machine interface for spinal cord injury gait rehabilitation. *Frontiers in neuroscience*, 10:359, 2016.
30. Andreas Schwarz, Patrick Ofner, Joana Pereira, Andreea Ioana Sburlea, and Gernot R Müller-Putz. Decoding natural reach-and-grasp actions from human eeg. *Journal of neural engineering*, 15(1):016005, 2017.
31. Andreas Schwarz, Joana Pereira, Reinmar Kobler, and Gernot R Müller-Putz. Unimanual and bimanual reach-and-grasp actions can be decoded from human eeg. *IEEE transactions on biomedical engineering*, 67(6):1684–1695, 2019.
32. Giulia Bressan, Giulia Cisotto, Gernot R Müller-Putz, and Selina Christin Wriessnegger. Deep learning-based classification of fine hand movements from low frequency eeg. *Future Internet*, 13(5):103, 2021.
33. Xiao Jiang, Gui-Bin Bian, and Zean Tian. Removal of artifacts from eeg signals: a review. *Sensors*, 19(5):987, 2019.
34. Kip A Ludwig, Rachel M Miriani, Nicholas B Langhals, Michael D Joseph, David J Anderson, and Daryl R Kipke. Using a common average reference to improve cortical neuron recordings from microelectrode arrays. *Journal of neurophysiology*, 101(3):1679–1689, 2009.
35. Bernhard Graimann, Brendan Allison, and Gert Pfurtscheller. Brain–computer interfaces: A gentle introduction. In *Brain-computer interfaces*, pages 1–27. Springer, 2009.
36. Gert Pfurtscheller and FH Lopes Da Silva. Event-related eeg/meg synchronization and desynchronization: basic principles. *Clinical neurophysiology*, 110(11):1842–1857, 1999.
37. Benjamin Blankertz, Ryota Tomioka, Steven Lemm, Motoaki Kawanabe, and Klaus-Robert Muller. Optimizing spatial filters for robust eeg single-trial analysis. *IEEE Signal processing magazine*, 25(1):41–56, 2007.
38. Benjamin Blankertz, Steven Lemm, Matthias Treder, Stefan Haufe, and Klaus-Robert Müller. Single-trial analysis and classification of erp components—a tutorial. *NeuroImage*, 56(2):814–825, 2011.
39. Fabien Lotte. A tutorial on eeg signal-processing techniques for mental-state recognition in brain–computer interfaces. *Guide to brain-computer music interfacing*, pages 133–161, 2014.
40. Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. Multiclass brain–computer interface classification by riemannian geometry. *IEEE Transactions on Biomedical Engineering*, 59(4):920–928, 2012.
41. 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. *Frontiers in neuroscience*, 6:39, 2012.
42. 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*, pages 629–636. Springer, 2010.
43. Younghak Shin, Seungchan Lee, Junho Lee, and Heung-No Lee. Sparse representation-based classification scheme for motor imagery-based brain–computer interface systems. *Journal of neural engineering*, 9(5):056002, 2012.
44. Moritz Grosse-Wentrup and Martin Buss. Multiclass common spatial patterns and information theoretic feature extraction. *IEEE transactions on Biomedical Engineering*, 55(8):1991–2000, 2008.
45. Marco Congedo. How riemannian geometry transformed bci? `https://github.com/lkorczowski/BCI-2021-Riemannian-Geometry-workshop/blob/master/slides/2021-BCI_Society-Riemannian_Geometry_History-Marco_Congedo.pdf`, 2021. Accessed: 21-07-2022.

46. Alan Julian Izenman. Linear discriminant analysis. In *Modern multivariate statistical techniques*, pages 237–280. Springer, 2013.
47. Shan Suthaharan. Support vector machine. In *Machine learning models and algorithms for big data classification*, pages 207–235. Springer, 2016.
48. Steven J Rigatti. Random forest. *Journal of Insurance Medicine*, 47(1):31–39, 2017.
49. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
50. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
51. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.
52. Zitong Wan, Rui Yang, Mengjie Huang, Nianyin Zeng, and Xiaohui Liu. A review on transfer learning in eeg signal analysis. *Neurocomputing*, 421:1–14, 2021.
53. Clemens Brunner, Robert Leeb, Gernot Müller-Putz, Alois Schlögl, and Gert Pfurtscheller. Bci competition 2008–graz data set a. *Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology*, 16:1–6, 2008.
54. Pedro Luiz Coelho Rodrigues, Christian Jutten, and Marco Congedo. Riemannian procrustes analysis: transfer learning for brain–computer interfaces. *IEEE Transactions on Biomedical Engineering*, 66(8):2390–2401, 2018.
55. He Zhao, Qingqing Zheng, Kai Ma, Huiqi Li, and Yefeng Zheng. Deep representation-based domain adaptation for nonstationary eeg classification. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2):535–545, 2020.
56. Praveen K Parashiva and A Prasad Vinod. Online hand motor imagery direction decoding using brain computer interface. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3182–3187. IEEE, 2021.
57. Steven J Luck. *An introduction to the event-related potential technique*. MIT press, 2014.
58. Kais Belwafi, Ridha Djemal, Fakhreddine Ghaffari, Olivier Romain, Bouraoui Ouni, and Sofien Gannouni. Online adaptive filters to classify left and right hand motor imagery. In *BIOSIGNALS*, pages 335–339, 2016.
59. Camille Benaroch, Khadijeh Sadatnejad, Aline Roc, Aurélien Appriou, Thibaut Monseigne, Smeety Pramij, Jelena Mladenovic, Léa Pillette, Camille Jeunet, and Fabien Lotte. Long-term bci training of a tetraplegic user: Adaptive riemannian classifiers and user training. *Frontiers in Human Neuroscience*, 15:635653, 2021.
60. Jianjun Meng, John H Mundahl, Taylor D Streitz, Kaitlin Maile, Nicholas S Gulachek, Jeffrey He, and Bin He. Effects of soft drinks on resting state eeg and brain–computer interface performance. *IEEE Access*, 5:18756–18764, 2017.
61. Herbert Ramoser, Johannes Muller-Gerking, and Gert Pfurtscheller. Optimal spatial filtering of single trial eeg during imagined hand movement. *IEEE transactions on rehabilitation engineering*, 8(4):441–446, 2000.
62. Min-Ho Lee, O-Yeon Kwon, Yong-Jeong Kim, Hong-Kyung Kim, Young-Eun Lee, John Williamson, Siamac Fazli, and Seong-Whan Lee. Eeg dataset and openbmi toolbox for three bci paradigms: an investigation into bci illiteracy. *GigaScience*, 8(5):giz002, 2019.

63. Christa Neuper, Reinhold Scherer, Miriam Reiner, and Gert Pfurtscheller. Imagery of motor actions: Differential effects of kinesthetic and visual–motor mode of imagery in single-trial eeg. *Cognitive brain research*, 25(3):668–677, 2005.
64. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
65. Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A. Engemann, Daniel Strohmeier, Christian Brodbeck, Roman Goj, Mainak Jas, Teon Brooks, Lauri Parkkonen, and Matti S. Hämäläinen. MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience*, 7(267):1–13, 2013.
66. Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
67. Jonathan Peirce, Jeremy R Gray, Sol Simpson, Michael MacAskill, Richard Höchenberger, Hiroyuki Sogo, Erik Kastman, and Jonas Kristoffer Lindeløv. Psychopy2: Experiments in behavior made easy. *Behavior research methods*, 51(1):195–203, 2019.
68. Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.
69. Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.
70. Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
71. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
72. Kyuhwa Lee, Dong Liu, Laetitia Perroud, Ricardo Chavarriaga, and José del R Millán. A brain-controlled exoskeleton with cascaded event-related desynchronization classifiers. *Robotics and Autonomous Systems*, 90:15–23, 2017.
73. Laura Ferrero, Mario Ortiz, Vicente Quiles, Eduardo Iáñez, José A Flores, and José M Azorín. Brain symmetry analysis during the use of a bci based on motor imagery for the control of a lower-limb exoskeleton. *Symmetry*, 13(9):1746, 2021.
74. Kai Keng Ang, Zheng Yang Chin, Haihong Zhang, and Cuntai Guan. 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, 2008.
75. Stephanie Brandl, Klaus-Robert Müller, and Wojciech Samek. Alternative csp approaches for multimodal distributed bci data. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 003742–003747. IEEE, 2016.
76. Jian-Guo Wang, Zeng Chen, and Yuan Yao. Personalized eeg feature extraction method based on filter bank and elastic network. In *International Conference*

on Bio-inspired Information and Communication Technologies, pages 116–129. Springer, 2020.
77. Maryam Alimardani, Shuichi Nishio, and Hiroshi Ishiguro. Effect of biased feedback on motor imagery learning in bci-teleoperation system. *Frontiers in systems neuroscience*, 8:52, 2014.
78. Eduardo Santamaria-Vazquez, Victor Martinez-Cagigal, Fernando Vaquerizo-Villar, and Roberto Hornero. Eeg-inception: A novel deep convolutional neural network for assistive erp-based brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 28(12):2773–2782, 2020.
79. Andreas Schwarz, Carlos Escolano, Luis Montesano, and Gernot R Müller-Putz. Analyzing and decoding natural reach-and-grasp actions using gel, water and dry eeg systems. *Frontiers in neuroscience*, 14:849, 2020.
80. Carmen Vidaurre, A Schlogl, Rafael Cabeza, Reinhold Scherer, and Gert Pfurtscheller. Study of on-line adaptive discriminant analysis for eeg-based brain computer interfaces. *IEEE transactions on biomedical engineering*, 54(3):550–556, 2007.
81. Laura Ferrero, Mario Ortíz, Vicente Quiles, Eduardo Iáñez, and José Maria Azorín. Improving motor imagery of gait on a brain–computer interface by means of virtual reality: A case of study. *IEEE Access*, 9:49121–49130, 2021.
82. Jianjun Meng and Bin He. Exploring training effect in 42 human subjects using a non-invasive sensorimotor rhythm based online bci. *Frontiers in human neuroscience*, 13:128, 2019.
83. Jan Saputra Müller, Carmen Vidaurre, Martijn Schreuder, Frank C Meinecke, Paul Von Bünau, and Klaus-Robert Müller. A mathematical model for the two-learners problem. *Journal of neural engineering*, 14(3):036005, 2017.
84. Xiu An, Deping Kuang, Xiaojiao Guo, Yilu Zhao, and Lianghua He. A deep learning method for classification of eeg data based on motor imagery. In *International Conference on Intelligent Computing*, pages 203–210. Springer, 2014.
85. Abbas Salami, Javier Andreu-Perez, and Helge Gillmeister. Eeg-itnet: An explainable inception temporal convolutional network for motor imagery classification. *IEEE Access*, 2022.
86. Benjamin Blankertz, Claudia Sanelli, Sebastian Halder, E Hammer, Andrea Kübler, Klaus-Robert Müller, Gabriel Curio, and Thorsten Dickhaus. Predicting bci performance to study bci illiteracy. *BMC Neurosci*, 10(Suppl 1):P84, 2009.
87. Jiansheng Niu and Ning Jiang. Pseudo-online detection and classification for upper-limb movements. *Journal of Neural Engineering*, 2022.
88. Dawei Cheng, Ye Liu, and Liqing Zhang. Exploring motor imagery eeg patterns for stroke patients with deep neural networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2561–2565. IEEE, 2018.

10 Appendix

10.1 Pipeline experiments

In the following sections, experimental setup of each pipeline experiment is described.

Window size Experiments for window size were run with a TG-LDA pipeline, with a window size of 250, 375, or 500 datapoints (sample frequency 250Hz), together with a filter order of 2 and a bandpass filter of 8-35Hz.

Common Average Referencing Experiments were run with a FBCSP with LDA pipeline, a TG with LDA pipeline for RG, and EEGNET for DL, with a window size of 500 datapoints, and a 2nd order bandpass filter of 8-35Hz for both pipelines.

Filter type Experiments were run with a TG-LDA pipeline for RG, with a window size of 500 datapoints, and a 2nd order bandpass filter of 8-35Hz, either filtered with filtfilt or SP.

Filter orders Experiments for FBCSP were run with a FBCSP-LDA a window size of 500 datapoints (2 seconds), and a FB of 10Hz to 35Hz in steps of 5. Experiments for RG were run with a TG-LDA pipeline, with a window size of 500 datapoints, and a bandpass filter of 8-35Hz. No experiments were done for DL as it was assumed that DL results would follow the same trend as RG.

FB bands Experiments for FBCSP were run with FBCSP-LDA, with a window size of 500 datapoints, and filters of 2nd order were applied, filtered using SP.

Filter bands RG experiments were run with TG-LDA, with a window size of 500 datapoints, and filters of 2nd order were applied, filtered using SP. EEGNET was used for DL experiments. Next to this, the DL pipeline consisted of a window size of 500 datapoints, and filters of 2nd order were applied, filtered using SP.

Machine learning models Experiments for FBCSP were run with a window size of 500 datapoints, filters of 2nd order were applied with FB of 5-25Hz, filtered using SP. For RG, experiments were run with a window size of 500 datapoints, filters of 2nd order were applied with a bandpass filter of 8-35Hz, filtered using SP.

10.2 Deep learning experiments

Hyperparameter search In Table 8, results of the parameter search for EEGNET in terms of final validation accuracy are presented. Here, pre-trained models were fine-tuned on first 4 trials of the validation subject, with 1 trial as validation trial and remaining 5 trials as test data. Results are presented in the form of final accuracy for the test trials. For lr, 0.001 was chosen. For $pdrop$: 0.4, and for D : 2. For fz, no best performing value was found, and the default fz : 8 from [17] was chosen.

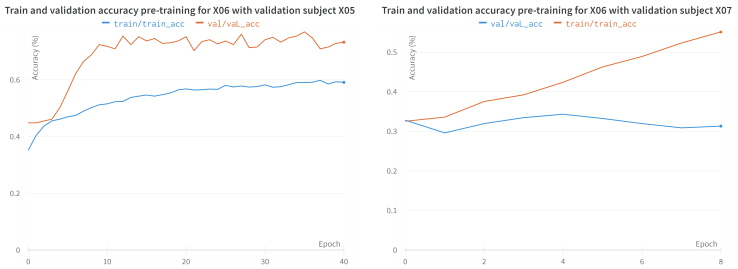
Table 8. Validation accuracies for hyperparameter search for validation subjects X03, X05, and X08. Best results of values of parameters are highlighted in bold.

	Learning rate		**Dropout rate**			**Filter size**			**D**		
Subject	0.001	0.005	0.1	0.25	0.4	4	8	16	1	2	3
X03	**42.6**	41.8	41.8	**43.3**	42.8	**44.7**	42.3	40.7	41.8	42.8	**43.2**
X05	**66.0**	62.4	63.7	63.9	**65.1**	61.0	**66.0**	65.7	60.7	**66.5**	65.5
X08	39.7	**41.1**	40.6	41.8	**42.1**	40.2	40.4	**40.7**	40.3	**40.8**	40.1

Subject	**All**	**C3, Cz, C4**	**Fz, Pz, Oz, PO7, PO8**
1	83.6%	84.2%	76.8%
2	73.8%	73.5%	57.7%

Pre-training Pre-training process was smoother when having high-performing subjects (X02, X04, X05) as validation subject. In Figure 18a, training and validation accuracy with X05 as validation subjects is plotted, in which a smooth curve is visible, where validation accuracy smoothly increases until around 70%, after which it seems unable to increase further. For validation subjects with lower final pre-train accuracy score, like X07, visible in Figure 18b, the training process shows almost no improvement in validation accuracy over epochs, resulting in final validation accuracy around 33%, which is equal to chance level.

Ablation study Different methods of fine-tuning were tested for test accuracy for subject X01 and X02. Comparison was made between fine-tuning only the last layer (linear layer), the last two layers (separable convolution and linear layer), and weights of entire network. The first two methods were executed either with re-initializing the weights or starting from pre-trained weights. Results are visible in Table 9. Fine-tuning all weights starting from the pre-trained weights outperformed other methods.



(a) (b)

Fig. 18. Training and validation accuracy of pre-training with validation subject **a)** X05 and **b)** X07.

Table 9. Ablation study for fine-tuning. Results for best fine-tune method are highlighted in bold.

	Last layer		**Last two layers**		**All**
Subject	Re-init	Pre-trained	Re-init	Pre-trained	Pre-trained
X01	34.6	37.9	33.5	38.4	**39.0**
X02	67.8	70.7	54.1	73.9	**76.4**