

Alma Mater Studiorum Università di Bologna  
Archivio istituzionale della ricerca

Interpretable and lightweight convolutional neural network for EEG decoding: Application to movement execution and imagination

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

*Published Version:*

Borra D., Fantozzi S., Magosso E. (2020). Interpretable and lightweight convolutional neural network for EEG decoding: Application to movement execution and imagination. NEURAL NETWORKS, 129, 55-74 [10.1016/j.neunet.2020.05.032].

*Availability:*

This version is available at: <https://hdl.handle.net/11585/786928> since: 2024-09-18

*Published:*

DOI: <http://doi.org/10.1016/j.neunet.2020.05.032>

*Terms of use:*

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).  
When citing, please refer to the published version.

(Article begins on next page)

## **HIGHLIGHTS**

- A parsimonious and interpretable convolutional NN is proposed for EEG decoding.
- Sinc- and depthwise convolutions are used for temporal and spatial filtering.
- A gradient-based technique is designed to interpret the learned features.
- The network outperforms a traditional machine learning algorithm and other CNNs.
- The learned spectral-spatial features match well-known EEG motor-related activity.

**INTERPRETABLE AND LIGHTWEIGHT CONVOLUTIONAL  
NEURAL NETWORK FOR EEG DECODING:  
APPLICATION TO MOVEMENT EXECUTION AND  
IMAGINATION**

**Davide Borra\*, Silvia Fantozzi, Elisa Magosso**

Department of Electrical, Electronic and Information Engineering “Guglielmo  
Marconi”, University of Bologna, Cesena Campus, Cesena, Italy

\*Corresponding Author

Davide Borra

Department of Electrical, Electronic and Information Engineering

University of Bologna-Cesena Campus

Via dell'Università 50, 47522 Cesena, ITALY

phone: +39 0547339240

email: [davide.borra2@unibo.it](mailto:davide.borra2@unibo.it)

1  
2  
3  
4  
5  
6  
7  
8  
9 **INTERPRETABLE AND LIGHTWEIGHT CONVOLUTIONAL**  
10 **NEURAL NETWORK FOR EEG DECODING:**  
11 **APPLICATION TO MOVEMENT EXECUTION AND**  
12 **IMAGINATION**

13  
14  
15  
16 **Davide Borra\*, Silvia Fantozzi, Elisa Magosso**  
17

18  
19  
20 Department of Electrical, Electronic and Information Engineering “Guglielmo  
21 Marconi”, University of Bologna, Cesena Campus, Cesena, Italy  
22

23  
24  
25  
26  
27  
28 \*Corresponding Author

29 Davide Borra

30 Department of Electrical, Electronic and Information Engineering

31 University of Bologna-Cesena Campus

32 Via dell’Università 50, 47522 Cesena, ITALY

33 phone: +39 0547339240

34 email: [davide.borra2@unibo.it](mailto:davide.borra2@unibo.it)  
35  
36  
37  
38

1 **ABSTRACT (max 250)**

2 Convolutional neural networks (CNNs) are emerging as powerful tools for EEG decoding: these  
3 techniques, by automatically learning relevant features for class discrimination, improve EEG  
4 decoding performances without relying on handcrafted features. Nevertheless, the learned features  
5 are difficult to interpret and most of the existing CNNs introduce many trainable parameters. Here,  
6 we propose a lightweight and interpretable shallow CNN (Sinc-ShallowNet), by stacking a temporal  
7 sinc-convolutional layer (designed to learn band-pass filters, each having only the two cut-off  
8 frequencies as trainable parameters), a spatial depthwise convolutional layer (reducing channel  
9 connectivity and learning spatial filters tied to each band-pass filter), and a fully-connected layer  
10 finalizing the classification. This convolutional module limits the number of trainable parameters and  
11 allows direct interpretation of the learned spectral-spatial features via simple kernel visualizations.  
12 Furthermore, we designed a post-hoc gradient-based technique to enhance [interpretation by](#)  
13 identifying the more relevant and more class-specific features. Sinc-ShallowNet was evaluated on  
14 benchmark motor-execution and motor-imagery datasets and against different design choices and  
15 training strategies. Results show that (i) Sinc-ShallowNet outperformed a traditional machine  
16 learning algorithm and other CNNs for EEG decoding; (ii) The learned spectral-spatial features  
17 matched well-known EEG motor-related activity; (iii) The proposed architecture performed better  
18 with a larger number of temporal kernels still maintaining a good compromise between accuracy and  
19 parsimony, and with a trialwise rather than a cropped training strategy. In perspective, the proposed  
20 approach, with its interpretative capacity, can be exploited to investigate cognitive/motor aspects  
21 whose EEG correlates are yet scarcely known, potentially characterizing their relevant features.

22

23 **Keywords:** Electroencephalography; Convolutional neural network; Sinc-convolutional layer;  
24 Feature learning; Interpretability

25

26

## 1 1. INTRODUCTION

2 Approaches based on machine learning algorithms provide powerful tools to analyse and decode  
3 brain activity from electroencephalographic (EEG) data, both in research and application areas. In  
4 particular, machine learning techniques have been exploited in many EEG-based Brain-Computer  
5 Interfaces (BCIs). In these systems, a feature extraction stage (McFarland et al., 2006) extracts the  
6 meaningful characteristics of the pre-processed (Bashashati et al., 2007) EEG signals and a  
7 downstream classification stage (Lotte et al., 2018) makes a decision based on the extracted  
8 characteristics, to provide the appropriate feedback to the user (Mak & Wolpaw, 2009). One popular  
9 and performing feature extraction algorithm is the filter bank common spatial pattern (FBCSP) (Ang,  
10 Chin, Zhang, & Guan, 2008) that applies a bank of bandpass filters (selected a priori) and extracts  
11 features for each frequency band based on the spatial filtering method. FBCSP has been widely used  
12 as EEG feature extraction method and won several competitions, such as BCI competition IV datasets  
13 2a and 2b (Ang et al., 2012) related to EEG decoding of imagined movements.

14 However, the traditional machine learning pipeline described above performs feature extraction  
15 and classification in separate steps. Furthermore, it strongly relies on a priori knowledge in the design  
16 of the feature extraction stage (e.g. the filters' cut-off frequencies in the FBCSP) and prevents that  
17 other potentially relevant (but unknown) features are extracted and used for decoding. For this reason,  
18 this approach may also have negative impact on decoding accuracy. Recently, machine learning  
19 innovations, proposed in the computer vision field and represented by convolutional neural networks  
20 (CNNs), have been transposed to EEG decoding tasks (Roy et al., 2019), mitigating the need for  
21 manual feature extraction. CNNs automatically learn features in a hierarchical structure from the  
22 input data in an end-to-end fashion, i.e. without separating the feature extraction, selection and  
23 classification steps. Thus, in the field of EEG decoding, CNNs can be trained by feeding EEG signals  
24 as input to the neural network, obtaining as output the corresponding predicted label. Accordingly,  
25 CNNs do not need any a priori knowledge about the meaningful characteristics of the signals for the

1 specific decoding task and have the potentiality to discover the relevant features (even so-far  
2 unknown) by using all input information.

3 An efficient way to provide EEG [signals as](#) input to CNNs is to design a 2D input representation  
4 with the electrodes along one dimension and time steps along the other (Borra et al., 2020a, 2020b;  
5 Cecotti & Graser, 2011; Farahat et al., 2019; Lawhern et al., 2018; Leeuwen et al., 2019; Manor &  
6 Geva, 2015; Schirrneister et al., 2017; Shamwell et al., 2016; Tang et al., 2017; Zeng et al., 2019;  
7 Zhao et al., 2019), preserving the original EEG representation i.e. non-transformed representation.  
8 Other input representations, e.g. transformed representations [such](#) as time-frequency [decomposition](#)  
9 (Bashivan et al., 2015; Sakhavi et al., 2015; Tabar & Halici, 2016), generally increase data  
10 dimensionality requiring more training data and/or regularization to learn meaningful features. CNNs  
11 with a non-transformed representation [are typically](#) designed by stacking individual temporal and  
12 spatial convolutional layers or a single spatio-temporal convolutional layer, and eventually deeper  
13 convolutional layers that learn patterns on the spatially filtered activations. CNNs based on these  
14 architectures have been successfully applied to several EEG decoding tasks, such as P300 detection  
15 tasks (Borra et al., 2020a; Cecotti & Graser, 2011; Farahat et al., 2019; Lawhern et al., 2018; Manor  
16 & Geva, 2015; Shamwell et al., 2016), motor imagery and execution decoding tasks (Schirrneister  
17 et al., 2017; Lawhern et al., 2018; Tang et al., 2017; Zhao et al., 2019; Borra et al., 2020b), anomaly  
18 detection tasks (Leeuwen et al., 2019), emotion classification (Zeng et al., 2019), and [they](#) have been  
19 [generally](#) proved to outperform traditional machine learning approaches. Despite these effective  
20 applications of CNNs in EEG decoding, there are still a number of critical issues that require further  
21 [investigation](#). Indeed, CNNs introduce a large number of trainable parameters requiring large training  
22 datasets to obtain a good fit, have a longer training time compared to simpler models, introduce many  
23 hyper-parameters (e.g. number of kernels, kernel sizes, number of layers, type of activation functions,  
24 etc.), and the automatically learned features are difficult to be interpreted. In particular, techniques  
25 that increase the interpretability of [the learned features](#) are receiving growing interest as key  
26 ingredients to achieve more robust validation [when using CNNs](#) (Montavon et al., 2018). In the field

1 of CNN-based EEG decoding, **increasing the** interpretability may be particularly relevant for  
2 neuroscientists as to the following aspects: (i) check the correct learning by verifying that the models  
3 do not rely on artefactual sources but on neurophysiological features; (ii) enable the understanding of  
4 which EEG features better discriminate the investigated classes; (iii) potentially characterize new  
5 features exploited by the network for the classification, and thus increase the insight into the neural  
6 correlates **underlying the classified behaviours**.

7 Several efforts have been made to increase CNN interpretability via post-hoc interpretation  
8 techniques (i.e. techniques that analyse the **trained** model). These techniques include temporal and  
9 spatial kernel visualizations (Cecotti & Graser, 2011; Lawhern et al., 2018), kernel ablation tests (i.e.  
10 selective removal of single kernels) (Lawhern et al., 2018), saliency maps (i.e. maps showing the  
11 gradient of CNN prediction with respect to its input example) (Farahat et al., 2019), gradient-  
12 weighted class activation mapping (Jonas et al., 2019), correlation maps between input features and  
13 outputs of given layers (Schirrneister et al., 2017). Some of these works face the interpretability issue  
14 together with other key issues previously cited, such as **model** complexity (in terms of number of  
15 layers and numbers of trainable parameters) and the **size of the** training **dataset**. Schirrneister et al.  
16 (2017) tested both a deeper CNN (DeepConvNet, with 5 convolutional layers and one fully-connected  
17 layer, ) and a shallower CNN (ShallowConvNet, with 2 convolutional layers and one fully-connected  
18 layer) for decoding movement execution and motor imagery, **analysed** the effect of increasing the  
19 amount of training examples (via cropped training), and used correlation maps to interpret the CNN  
20 learned features. Lawhern et al. (2018) designed a shallow and lightweight CNN (EEGNet, with 3  
21 convolutional layers and one fully-connected layer) by introducing depthwise and separable  
22 **convolutions** that reduced the number of parameters to fit, tested a range of EEG decoding tasks with  
23 various training sizes, and interpreted the learned features via kernel visualization and ablation.

24 Besides post-hoc techniques, network interpretability may be increased by introducing directly  
25 interpretable layers within the network architecture; importantly, these layers may intrinsically reduce  
26 the number of trainable parameters too, **promoting** more interpretable and, at the same time,

1 lightweight CNNs. [Very recently](#), few studies have explored this approach in CNNs for EEG  
2 decoding. Zhao et al. (2019) introduced a time-frequency convolutional layer in an architecture  
3 inspired by ShallowConvNet (Schirrmester et al., 2017) to learn time-frequency filters designed by  
4 real-valued Morlet wavelets. In a previous preliminary work (Borra et al., 2020b), for the first time  
5 we used a temporal sinc-convolutional layer (Ravanelli & Bengio, 2018) for EEG decoding, [included](#)  
6 [in](#) an architecture based on DeepConvNet (Schirrmester et al., 2017), to learn [temporal](#) filters [defined](#)  
7 [by](#) parametrized [sinc-functions that implement band-pass filters](#). Instead of learning all the kernel  
8 values as in a traditional convolutional layer, both in the wavelet- and sinc-convolutional layer only  
9 2 parameters for each kernel need to be learned and they are directly interpretable: the bandwidth of  
10 the Gaussian and the wavelet central frequency in one case (Zhao et al., 2019), and the two cutoff  
11 frequencies of the band-pass filters in the other case (Borra et al., 2020b). While this approach appears  
12 promising, its use in EEG decoding is still limited and the so-far proposed CNNs (Borra et al., 2020b;  
13 Zhao et al., 2019) have some limitations. Indeed, except for a single directly interpretable  
14 convolutional layer, the rest of these CNNs uses traditional less interpretable convolutional layers.  
15 This aspect, not only may hinder the overall interpretability of the learned features, but also requires  
16 a large number of trainable parameters leading to models more prone to overfitting and this is  
17 especially true in case of the deep CNN we previously proposed (Borra et al., 2020b). Furthermore,  
18 each of these CNNs has been tested only on a single decoding task (movement imagination (Zhao et  
19 al., 2019), and movement execution (Borra et al., 2020b)), and the ability of each network to  
20 generalize across motor paradigms has not been verified.

21 The purpose of this work is to contribute to the recent developments of CNN-based EEG  
22 decoding by [designing](#) and [analysing a novel](#) CNN that includes interpretable and optimized layers,  
23 able to increase the overall interpretability of the network, reduce the number of trainable parameters  
24 and, at the same time, ensure good performances compared [to existing](#) state-of-the art (SOA)  
25 algorithms. [The CNN proposed here is a](#) lightweight shallow CNN, named Sinc-ShallowNet,  
26 obtained by stacking two convolutional layers [that extract spectral and spatial EEG features](#)

1 respectively, followed by a fully-connected layer finalizing the classification. The two convolutional  
2 layers are specifically devised to increase interpretability and decrease the number of trainable  
3 parameters and consist of a temporal sinc-convolutional layer and a spatial depthwise convolutional  
4 layer. The spatial depthwise convolutional layer ties spatial filters to each particular band-pass filter  
5 learned by the temporal sinc-convolutional layer, enabling the learning of spatial features related to  
6 specific frequency ranges. The proposed architecture was applied to decode sensorimotor rhythms  
7 both during motor execution (ME) and motor imagery (MI) using public benchmark datasets.  
8 Moreover, an extensive analysis of Sinc-ShallowNet was performed including the following aspects:

- 9 i. Comparison of the decoding performance of Sinc-ShallowNet with SOA decoding algorithms,  
10 including one traditional machine learning pipeline based on FBCSP coupled with regularized  
11 Linear Discriminant Analysis (rLDA) and other three CNNs (ShallowConvNet and  
12 DeepConvNet (Schirrneister et al., 2017), EEGNet (Lawhern et al., 2018)).
- 13 ii. Assessment of some design choices on Sinc-ShallowNet performance in a post-hoc hyper-  
14 parameter evaluation procedure inspired by Schirrneister et al. (2017). The evaluated design  
15 choices concern: the number of the temporal band-pass filters, the number of spatial filters for  
16 each temporal filter, the introduction of an optional recombination of the spatial activations, and  
17 the size of activation aggregation (average pooling) before the fully-connected layer.
- 18 iii. Evaluation of the effect of increasing the training data size via cropped training compared to  
19 trialwise training. Indeed, the effect of cropped training on different CNN architectures is still  
20 unclear. Schirrneister et al. (2017) found that cropped training significantly increased the  
21 performance of deep architectures (DeepConvNet), while no significant effect was obtained with  
22 shallow architectures (ShallowConvNet). Despite this, other shallow architectures (Zhao et al.,  
23 2019) were trained with a cropped strategy. Therefore, we evaluated the effect of the training  
24 strategy on the performance of Sinc-ShallowNet and of the re-implemented SOA CNNs.
- 25 iv. Feature interpretation. Since the trainable parameters of the temporal sinc-convolutional layer  
26 are the cutoff frequencies of the learned band-pass filters, the learned spectral features can be

1 directly visualized and interpreted once the training ends. Furthermore, inspired from the saliency  
 2 maps (Simonyan et al., 2013), we designed a post-hoc interpretation technique named “temporal  
 3 sensitivity analysis” (as it acts on the kernels of the temporal sinc-convolutional layer). This  
 4 technique enables the identification of the more relevant and more class-specific band-pass filters  
 5 and the spatial features (as learned in the depthwise convolutional layer) related to these band-  
 6 pass filters can be visualized.

## 7 2. METHODS

8 This section is devoted to the description of the proposed CNN for EEG motor decoding. At first,  
 9 we define the problem of EEG decoding into the framework of supervised classification learning via  
 10 CNNs and provide notations useful for the following description. Subsequently, we illustrate the  
 11 benchmark datasets used to train and test the CNNs (the proposed one and the SOA CNNs), the  
 12 architecture of the proposed CNN, the training procedure, and finally the post-hoc interpretation  
 13 technique. The CNNs were developed in PyTorch (Paszke et al., 2017) and trained from scratch using  
 14 a workstation equipped with an AMD Threadripper 1900X, NVIDIA TITAN V and 32 GB of RAM.

### 15 2.1. Problem definition and notations

16 Let us assume to have an EEG dataset collected from each subject. Each dataset consists of  
 17 separated trials (e.g. obtained by epoching the original continuous EEG recording), with each trial  
 18 belonging to one of several classes (let’s say  $N_c$  classes). By indicating with  $M^{(s)}$  the total number of  
 19 trials for s-th subject, the corresponding dataset can be denoted by  $D^{(s)} =$   
 20  $\left\{ \left( X_0^{(s)}, y_0^{(s)} \right), \left( X_2^{(s)}, y_2^{(s)} \right), \dots, \left( X_{M^{(s)}-1}^{(s)}, y_{M^{(s)}-1}^{(s)} \right) \right\}$ .  $X_i^{(s)} \in \mathbb{R}^{C \times T}$  contains the pre-processed EEG  
 21 signals of the i-th trial ( $0 \leq i \leq M^{(s)} - 1$ ), collected at  $C$  electrodes and  $T$  time samples;  $y_i^{(s)}$  is the  
 22 class label of the i-th trial and assumes one value among the  $N_c$  possible values, i.e.  $\forall i, y_i^{(s)} \in L =$   
 23  $\{l_0, l_2, \dots, l_{N_c-1}\}$ . The two public EEG datasets used here were EEG signals collected while the  
 24 subjects executed (High-Gamma dataset, see Section 2.2.1) or imagined (BCI-IV2a dataset, see

1 [Section 2.2.2](#)) **movements** of different body parts. Thus, the classes discriminate among the specific  
2 body part moved (or imagined to be moved) during each trial (e.g.  $l_0 = \text{“Right Hand”}$ ,  $l_1 = \text{“Left$   
3  $\text{Hand”}$  etc.).

4 The problem at hand is to train a classifier  $f$  so that it learns, from a training set of [EEG trials](#), to  
5 assign the correct label to [previously unseen EEG trials](#). Specifically, the parametric classifier is  
6  $f(X_i^{(s)}; \theta^{(s)}) : \mathbb{R}^{C \times T} \rightarrow L$ , parametrized by parameters  $\theta^{(s)}$ , which assigns a label  $y_i^{(s)}$  to the trial  
7  $X_i^{(s)}$ , i.e.  $y_i^{(s)} = f(X_i^{(s)}; \theta^{(s)})$ . The classifier  $f(X_i^{(s)}; \theta^{(s)})$  can be formally interpreted as the  
8 composition of two functions: (i) a first function  $\phi$  that extracts a (vector-valued) feature  
9 representation  $\phi(X_i^{(s)}; \theta_\phi^{(s)}) : \mathbb{R}^{C \times T} \rightarrow \mathbb{R}^{N_\phi}$  ( $N_\phi$  denoting the number of extracted features) having  
10 parameters  $\theta_\phi^{(s)}$ ; (ii) a second function  $g(\phi^{(s)}; \theta_g^{(s)}) : \mathbb{R}^{N_\phi} \rightarrow L$  with parameters  $\theta_g^{(s)}$  that exploits the  
11 extracted features to finalize the classification, that is  $f(X_i^{(s)}; \theta^{(s)}) = g(\phi(X_i^{(s)}; \theta_\phi^{(s)}); \theta_g^{(s)})$ . [When](#)  
12 [the decoder  \$f\$  is implemented by a CNN](#), the two stages (feature extraction and final classification)  
13 are learned jointly with all parameters  $\theta^{(s)}$  optimized simultaneously. By keeping superscript  $s$  in the  
14 classifier parameters, we emphasize that the parameters are optimized separately per subject, as here  
15 a within-subject training procedure ([see Section 2.4](#)) was adopted. The overall set of trials  $D^{(s)}$  of  
16 each subject is divided into a training set, used to optimize the parameters  $\theta^{(s)}$  for the specific subject,  
17 and a test set used to evaluate the performance of the learned subject-specific decoder.

18 Of course, besides the trainable parameters  $\theta^{(s)}$ , the network hyper-parameters (i.e. [parameters](#)  
19 [that define the functional form of decoder  \$f\$  not adapted by the learning itself, such as the number of](#)  
20 [layers, number and size of convolutional kernels, type of activation function, etc.](#)) may affect the  
21 decoding accuracy.

22 In the following, we assume that the generic trial  $X_i^{(s)} \in \mathbb{R}^{C \times T}$  is given in input to the CNNs as a  
23 2D matrix of shape  $(C, T)$ , having the time steps along the width and the electrodes along the height.

## 24 **2.2. Datasets**

1 The datasets used in this study are two common benchmark MI- and ME-EEG datasets for  
2 [sensorimotor](#) rhythm decoding. It is known that the  $\alpha$ ,  $\beta$  and  $\gamma$  bands are associated with movement-  
3 related spectral power modulations and thus provide class-discriminative information (Ball et al.,  
4 2008; Crone et al., 1998; G. Pfurtscheller, 1981; G. Pfurtscheller et al., 2006; G. Pfurtscheller &  
5 Aranibar, 1977; G. Pfurtscheller & Berghold, 1989; G. Pfurtscheller & Silva, 1999; Gert Pfurtscheller  
6 et al., 1994). In the following, these datasets are described together with the light pre-processing  
7 applied to obtain the trials  $X_i^{(s)}$  used to train and test the [CNNs](#).

### 8 **2.2.1. Motor execution: High-Gamma dataset**

9 High-Gamma dataset is a 128-channel ME-EEG dataset acquired from 14 healthy subjects (age  
10  $27.2 \pm 3.6$ , 6 female, 2 left-handed) by Schirrneister et al. (2017) and made freely available  
11 (<https://web.gin.g-node.org/robintibor/high-gamma-dataset>). Each subject performed roughly 1000  
12 ( $963.1 \pm 150.9$  mean  $\pm$  [standard deviation \(std\)](#) across participants) four-second trials of movement  
13 execution (three different movements) and of rest. The three movements were repetitive right- and  
14 left-hand sequential finger tapping, and repetitive toes clenching. Therefore, the decoding problem is  
15 a 4-way classification task. This dataset is well-suited for extracting information from the high  $\gamma$  band  
16 ( $> 50\text{Hz}$ ) since it was acquired in a laboratory optimized for the recording of high-frequency EEG  
17 components (Schirrneister et al., 2017).

18 EEG signals were downsampled from 500 to 250 Hz, the same sample frequency as the other  
19 [analysed](#) dataset (see [Section 2.2.2](#)), so that the CNN hyper-parameters related to the temporal  
20 dimension (i.e. temporal kernel and pooling sizes) were kept the same. The 44 signals relative to the  
21 electrodes covering the motor cortex were selected (Figure 1a) as done in (Schirrneister et al., 2017)  
22 and a high-pass 3<sup>rd</sup> order Butterworth filter was applied with a cutoff frequency of 4 Hz. Then, each  
23 electrode signal was standardized by applying an exponential moving average window with a decay  
24 factor of 0.999 as done in (Schirrneister et al., 2017). Each signal was epoched between -0.5 and 4.0  
25 s relative to the movement onset, so that each trial contains EEG values at  $C = 44$  electrodes and at

1  $T = 1125$  time samples organized in a single input feature map ( $K = 1$ , denoting with  $K$  the number  
2 of the input feature maps). Finally, the resulting trials were cleaned from high-amplitude artefacts by  
3 removing those with at least one electrode signal with absolute value  $> 800\mu V$ . Based on the previous  
4 description, the CNN input (corresponding to a single trial) had shape  $(K, C, T) = (1, 44, 1125)$  in  
5 this case.

6 For the sake of reproducibility of the results, the trial set  $D^{(s)}$  of the  $s$ -th subject was split as in  
7 the original paper (Schirrneister et al., 2017) for training and testing: for each subject, 160 trials (40  
8 for each class) were used as test set and the remaining as training set. In addition, the training set was  
9 further split into a validation set (20% of the training set) in order to perform early stopping during  
10 the first step of the optimization process (see Section 2.4).

### 11 **2.2.2. Motor imagery: BCI-IV2a dataset**

12 BCI-IV2a dataset is a 22-channel MI-EEG dataset collected for the BCI Competition IV  
13 (Tangermann et al., 2012). This set comprises four classes of imagined movements of left and right  
14 hands, feet and tongue, acquired from 9 participants and made freely available  
15 (<http://www.bbci.de/competition/iv/>). Therefore, the decoding problem is a 4-way classification task.  
16 The organizers of the challenge provided the dataset sampled at 250 Hz and band-pass filtered  
17 between 0.5 and 100 Hz. All 22 signals were used, and the montage is shown in Figure 1b.

18 The EEG signals were band-pass filtered between 4 and 38 Hz with a 3<sup>rd</sup> order Butterworth filter  
19 and each electrode signal was standardized by applying an exponential moving average window with  
20 a decay factor of 0.999 (Schirrneister et al., 2017). Then, the signals were epoched between 0.5 and  
21 2.5 s relative to the movement onset of movement imagination, as done in previous studies (Lawhern  
22 et al., 2018; Lotte, 2015; Sakhavi et al., 2015). In this case, the CNN input (i.e. the single trial) had  
23 shape (1, 22, 500).

24 Here we used the same training set (288 trials per subject, balanced between the classes) and test  
25 set (288 trials per subject, balanced between the classes) provided by the organizers of the

1 competition. The training set was further split into a validation set (20% of the training set) in order  
2 to perform early stopping during the first step of the optimization process (see Section 2.4).

3 [Figure 1 about here.]

### 4 **2.3. Sinc-ShallowNet**

5 The proposed architecture is designed with three fundamental blocks, each of them composed by  
6 a few layers. The blocks of the proposed architecture and their fundamental layers are shown in Figure  
7 2; a detailed description of the architecture (including the name, output shape and number of trainable  
8 parameters of each layer) is reported in Table 1. Block 1 has the function to extract spectral and spatial  
9 features from the input data, via temporal and spatial convolutional layers, respectively. The  
10 performed convolutions are designed to reduce the number of trainable parameters while increasing  
11 their interpretability. As to the temporal convolution, this is achieved via a sinc-convolutional layer  
12 (see Section 2.3.1), while for the spatial convolution, this is achieved via a depthwise convolutional  
13 layer (Chollet, 2016). Block 2 is devoted to perform a temporal aggregation (via a pooling layer) of  
14 the first block feature maps. Block 3 is designed to finalize the classification including a single fully-  
15 connected layer. The term “sinc” of Sinc-ShallowNet is related to the inclusion of the temporal sinc-  
16 convolutional layer within the first block; the term “shallownet” refers to the relative low number of  
17 trainable layers.

18 [Figure 2 about here.]

19 [Table 1 about here.]

20 In the first two blocks, the output of each layer can be interpreted as a collection of spatio-  
21 temporal feature maps. Thus, its shape can be described by a tuple of 3 integers, with the first integer  
22 indicating the number of feature maps provided by the layer, the second and third integers the number  
23 of spatial and temporal samples within each map, respectively. Each convolutional layer in these  
24 blocks is characterized by the number of convolutional kernels ( $K$ ), kernel size ( $F$ ), stride size ( $S$ ),  
25 and padding size ( $P$ ). In addition, depthwise convolution introduces also a depth multiplier ( $D$ ), that

1 specifies the number of kernels to learn for each input feature map. Since Sinc-ShallowNet has two  
 2 convolutional layers, the previous symbols are provided with subscript (“1”, “2”). The pooling layer  
 3 in block 2 is described by the pool size ( $F_p$ ) and pool stride ( $S_p$ ). **Since the adopted convolutions and**  
 4 **pooling are 2D**, the hyper-parameters  $F_i, S_i, P_i$  ( $i = 1, 2$ ),  $F_p, S_p$  are tuples of two elements: the first  
 5 element refers to the spatial dimension, while the second to the temporal dimension.

6 Block 1 and block 2 stacked together can be seen as implementing the function  
 7  $\phi\left(X_i^{(s)}; \theta_\phi^{(s)}\right): \mathbb{R}^{C \times T} \rightarrow \mathbb{R}^{N_\phi}$  (described in Section 2.1), where  $N_\phi$  is the overall number of units  
 8 provided as output by block 2. Block 3 receives this flattened feature map and finalizes the  
 9 classification, implementing a dense softmax classification. Thus, this block realizes the function  
 10  $g\left(\phi^{(s)}; \theta_g^{(s)}\right): \mathbb{R}^{N_\phi} \rightarrow L$  (described in Section 2.1). Of course, all parameters of the three blocks are  
 11 optimized simultaneously during the training, without any separation between the feature extraction  
 12 and classification stages.

13 In the following, we will first describe the mathematical aspects of the temporal sinc-  
 14 convolutional layer and the motivation for its inclusion. Then, the structure and function of each block  
 15 will be detailed.

### 16 2.3.1. Sinc-convolutional layer

17 Recently, Ravanelli and Bengio (2018) designed a CNN for speaker recognition (SincNet)  
 18 including a “sinc-convolutional layer”, that forces each kernel to describe a band-pass filter. In a  
 19 traditional convolutional layer, each value of a kernel is learned during the optimization. In a sinc-  
 20 convolutional layer, each value of a kernel is defined by a parametrized function, forcing the kernel  
 21 description to belong to a specific subset of temporal filters (here only band-pass filters) and at the  
 22 same time reducing the number of trainable parameters. **This** implementation promotes the learning  
 23 of more meaningful and well-defined temporal filters.

24 Considering the  $i$ -th electrode signal  $x_i$  (here, for simplicity the superscript  $s$  referring to a specific  
 25 subject is omitted), the 1D convolution between this signal and the  $j$ -th kernel  $k_j$  is (Equation 1):

$$1 \quad o_{i,j}[n] = x_i[n] * k_j[n] = \sum_{l=0}^{F-1} x_i[n-l] \cdot k_j[l], \quad (1)$$

2 where  $i \in [0, C - 1]$  with  $C$  representing the number of EEG electrodes,  $j \in [0, K - 1]$  with  $K$   
3 representing the number of temporal kernels, and  $F$  is the kernel size. Since, for brevity, we are  
4 describing a 1D convolution, here  $F$  is 1D (i.e.  $F$  represents the length of the filter along the temporal  
5 dimension). For instance, let's say  $F = 65$  for capturing frequencies at  $\sim 4$  Hz and above in case of  
6 data at 250 Hz sampling rate (Lawhern et al., 2018).

7 The kernel values of a sinc-convolutional layer can be obtained by evaluating the parametrized  
8 function  $k_j'[n; \theta_j]$  with a specific set of trainable parameters  $\theta_j$  defining the  $j$ -th band-pass filter. To  
9 describe band-pass filters in the frequency domain, the amplitude  $K_j'$  can be expressed as (Equation  
10 2):

$$11 \quad K_j'[f; f_{1,j}, f_{2,j}] = \text{rect}\left(\frac{f}{2f_{2,j}}\right) - \text{rect}\left(\frac{f}{2f_{1,j}}\right), \quad (2)$$

12 where  $\theta_j = \{f_{1,j}, f_{2,j}\}$  is the set of the trainable parameters of the  $j$ -th kernel. This set includes only  
13 the inferior ( $f_{1,j}$ ) and the superior ( $f_{2,j}$ ) cutoff frequencies of the  $j$ -th band-pass filter, reducing the  
14 number of trainable parameters of the temporal convolutional layer from  $F = 65$  to 2 for each  
15 temporal kernel. In the temporal domain,  $k_j'$  can be expressed as (Equation 3):

$$16 \quad k_j'[n; f_{1,j}, f_{2,j}] = 2f_{2,j} \text{sinc}(2\pi f_{2,j}n) - 2f_{1,j} \text{sinc}(2\pi f_{1,j}n). \quad (3)$$

17 To alleviate the effects of the inevitable truncation of  $k_j'$  on the characteristics of the filters (e.g.  
18 passband ripples, reduced stopband attenuation), the function is multiplied by a Hamming window  
19 (Equation 4) (Ravanelli & Bengio, 2018):

$$20 \quad \begin{cases} k_{w,j}'[n; f_{1,j}, f_{2,j}] = k_j'[n; f_{1,j}, f_{2,j}] \cdot w[n] \\ w[n] = 0.54 - 0.46 \cos\left(\frac{2\pi n}{F-1}\right) \end{cases} \quad (4)$$

21 The [so-defined](#) convolutional layer can be integrated into a traditional CNN to learn band-pass  
22 filters in the first layer, with only the two cutoff frequencies as trainable parameters. In [this](#) study,  
23 these frequencies were randomly initialized from a uniform distribution in the frequency range of

1 interest: (4,125] Hz and (4,38] Hz for ME- and MI-EEG signals, respectively. During the  
2 optimization, these frequencies were updated in the range of interest by keeping  $f_{2,j} > f_{1,j}$ .

### 3 **2.3.2. Block 1: Spectral and spatial feature extraction**

4 The first block (see Figure 2 and Table 1) performed a separate spectral and spatial feature  
5 learning. The first layer of this block was a 2D temporal sinc-convolutional layer that learned  $K_1 =$   
6 32 band-pass temporal filters with a low number of learnable parameters. The filter size  $F_1$  was set  
7 to (1,65) to extract information at 4 Hz and above, since the CNN input data were high-pass filtered  
8 at 4 Hz in the pre-processing stage (see Section 2.2). Following this layer, batch normalization (see  
9 Section 2.5) (Ioffe & Szegedy, 2015) was introduced along the feature map dimension. Then, a 2D  
10 spatial depthwise convolutional layer was introduced to learn  $D_2 = 2$  spatial filters of size  $(C, 1)$  for  
11 each temporal feature map, with a total number of  $K_2 = K_1 \cdot D_2$  spatial filters. The depthwise  
12 convolution is not fully-connected with the previous temporal feature maps (see Figure 2), reducing  
13 the number of trainable parameters. Moreover, it allows a straightforward extraction of the spatial  
14 distribution of each band-pass filter, making the interpretation of the learned CNN features easier. In  
15 this layer, kernel maximum norm constraint was used.

### 16 **2.3.3. Block 2: Aggregation**

17 The second block (see Figure 2 and Table 1) was designed to perform a temporal aggregation of  
18 the first block output. First, batch normalization (see Section 2.5) (Ioffe & Szegedy, 2015) along the  
19 feature map dimension was applied to the neurons of the spatial depthwise convolutional layer,  
20 followed by a non-linear activation function. In this study, Exponential Linear Units (ELUs) were  
21 adopted with activation function  $f(x) = x, x > 0$  and  $f(x) = \alpha \cdot (\exp(x) - 1), x \leq 0$ , as this non-  
22 linearity allows faster and more noise-robust learning than other non-linearities (Clevert et al., 2015).  
23 Furthermore, Schirrneister et al. (2017) reported better performance for shallow and deep CNNs  
24 applied to EEG motor decoding when using ELUs compared to other activation functions. The  $\alpha$   
25 parameter is the ELU hyper-parameter that controls the saturation value for negative inputs and  $\alpha =$

1 1 was set for the proposed architecture. Then, an average pooling layer was introduced to reduce the  
 2 number of trainable parameters in the transition from the block 2 and the subsequent fully-connected  
 3 layer in block 3, i.e. the convolutional-to-dense connections. A pool size of  $F_p = (1, 109)$  and pool  
 4 stride of  $S_p = (1, 23)$  were used. These hyper-parameters allow the extraction of averaged spatial  
 5 activations of  $\sim 500$  ms with a stride of  $\sim 100$  ms. Lastly, a dropout layer (Srivastava et al., 2014)  
 6 was introduced (see Section 2.5).

### 7 2.3.4. Block 3: Classification

8 After the second block, a flatten layer was introduced to unroll the second block output values,  
 9 resulting in a 1D array of features extracted by the previous layers. These values are densely  
 10 connected with a single fully-connected layer containing  $N_c = 4$  neurons.

11 Accordingly, the entire CNN maps the input data of the  $i$ -th trial  $X_i^{(s)}$  to one real number per  
 12 class, i.e.  $h(X_i^{(s)}; \theta^{(s)}): \mathbb{R}^{C \times T} \rightarrow \mathbb{R}^{N_c}$ . These  $N_c$  outputs are transformed via a softmax activation  
 13 function to obtain the conditional probabilities of the labels  $l_k \forall k \in L = \{l_0, l_1, \dots, l_{N_c-1}\}$ :

14  $p(l_k | X_i^{(s)}, \theta^{(s)}) = \frac{\exp h_k(X_i^{(s)}; \theta^{(s)})}{\sum_{j=0}^{N_c-1} \exp h_j(X_i^{(s)}; \theta^{(s)})}$ . Since the training strategy adopted was a within-subject training  
 15 (see Section 2.4), the softmax provides subject-specific conditional distribution over the  $N_c$  classes  
 16 for each example. The final classification is performed by assigning the label with the maximum  
 17 conditional probability to the trial  $X_i^{(s)}$ , i.e.  $y_i^{(s)} = f(X_i^{(s)}; \theta^{(s)}) = \arg \max_{l_k} p(l_k | X_i^{(s)}, \theta^{(s)})$ .

18 Based on the number of trainable parameters within each layer (see Table 1), Sinc-ShallowNet  
 19 introduced a total number of trainable parameters of 13828 and 5508, for ME- and MI-EEG signals,  
 20 respectively.

### 21 2.3.5. Design choices

22 In the following, the Sinc-ShallowNet described as in the previous sections (with the  
 23 corresponding hyper-parameters, see Table 1) will be denoted as the “basal” Sinc-ShallowNet. In

1 order to evaluate the influence of specific hyper-parameters of interest on the performance metric, a  
2 post-hoc hyper-parameter evaluation was performed by testing some variants compared to the basal  
3 architecture. The investigated hyper-parameters were: (i) the number of the temporal filters in block  
4 1 ( $K_1$ ); (ii) the number of the spatial filters per temporal filter in block 1 ( $D_2$ ); (iii) the pooling size  
5  $F_p$  and stride  $S_p$  of the average pooling in block 2; (iv) the recombination of the spatial activations.  
6 **In the condition** (iv), a pointwise convolution was included as the first layer in block 2 (fed by the  
7 outputs of the spatial depthwise convolution), followed by the other layers of block 2 (i.e. batch  
8 normalization, non-linear activation, average pooling, dropout).

9 These alternative design choices were evaluated through an extensive experimentation as  
10 described and motivated **in Table 2**.

11 [Table 2 about here.]

12 From the specifications reported **in Table 2**, five variants of Sinc-ShallowNet were designed by  
13 changing **one** specific hyper-parameter at a time while keeping all other hyper-parameters fixed, as  
14 previously done in Schirrmeister et al. (2017) and Farahat et al. (2019), and were **trained as** specified  
15 **in Section 2.4.1**.

## 16 **2.4. Training**

### 17 **2.4.1. Trialwise training strategy**

18 For each subject, the entire trial was used as input and the corresponding trial label as target to  
19 optimize one CNN per subject (within-subject training). **Weights were** randomly initialized adopting  
20 a Xavier uniform initialization scheme (Glorot & Bengio, 2010) and biases were initialized to zero.  
21 **The cutoff frequencies of the temporal sinc-convolutional layer were initialized as described**  
22 **previously (see Section 2.3.1)**. The trainable parameters  $\theta^{(s)}$  were optimized such that the parametric  
23 classifier assigned high probabilities to the correct labels by minimizing the sum of the per-example  
24 losses computed on the  $N$  training examples, converging to an optimal trainable parameter set  $\theta^{(s)*}$   
25 (Equation 5):

$$1 \quad \theta^{(s)*} = \arg \min_{\theta^{(s)}} \sum_{i=0}^{N-1} \text{loss} \left( y_i^{(s)}, p(l_k | X_i^{(s)}, \theta^{(s)}) \right), \quad (5)$$

2 where

$$3 \quad \text{loss} \left( y_i^{(s)}, p(l_k | X_i^{(s)}, \theta^{(s)}) \right) = \sum_{k=0}^{N_c-1} -\log \left( p(l_k | X_i^{(s)}, \theta^{(s)}) \right) \cdot \delta(y_i = l_k) \quad (6)$$

4 is the negative log likelihood of the labels. The minimization of the negative log likelihood is  
 5 equivalent to the minimization of the cross entropy between the empirical probability distribution  
 6 defined by the training labels and the probability distribution defined by the model. The parameters  
 7 were optimized via mini-batch stochastic gradient descent, using gradients computed via  
 8 backpropagation. Adaptive moment estimation (Adam) (Kingma & Ba, 2014), a commonly used  
 9 adaptive learning rate optimization algorithm, was used as optimizer with a learning rate of 1e-3 and  
 10 a mini-batch size of 64 trials.

11 The training phase was divided into two steps (Goodfellow et al., 2013). During the first training  
 12 step (800 maximum number of epochs), the CNN was trained until the validation loss reached its  
 13 minimum, performing early stopping. The training loss recorded at the first run minimum became the  
 14 target threshold for the second run. During the second training step (800 maximum number of  
 15 epochs), the validation set was included in the training set and the optimization continued until the  
 16 validation loss reached the threshold loss.

17 This trialwise training strategy was applied to the basal Sinc-ShallowNet (Table 1), and all its  
 18 variants (Table 2) on both ME- and MI-EEG dataset, to test the effect of different design choices on  
 19 the decoding accuracy (see Section 2.3.5). Moreover, this strategy was applied to the three re-  
 20 implemented SOA CNNs on both datasets, for a comparison with Sinc-ShallowNet performance (see  
 21 Section 2.6), as well as to evaluate how the two training strategies affect different CNN architectures  
 22 (see Sections 2.4.2 and 2.6).

### 23 **2.4.2. Cropped training strategy**

24 Schirrneister et al. (2017) introduced a cropped training strategy for EEG decoding: they used  
 25 crops of trials (i.e. sliding time windows within the trial) as input for the CNNs instead of the entire

1 trial and set the target label of each crop equal to the label of the trial the crop belonged to. This leads  
 2 to an augmented dataset that could increase the performance on the test set (i.e. additional regularizer  
 3 effect). Actually, Schirrneister et al. (2017) reported a statistically significant improvement of  
 4 cropped training only for deep architectures. Here, cropped training was applied to Sinc-ShallowNet  
 5 (in its basal version), as well as to the re-implemented SOA CNNs, to compare trialwise training with  
 6 cropped training for each network, in order to further study the effect of cropped training depending  
 7 on the CNN architecture. To perform cropped training and allow a strict comparison with results of  
 8 Schirrneister et al. (2017), the pre-processing of the MI dataset had to be modified by epoching  
 9 signals between 0.5-4.0 s to keep the same epoching procedure as in (Schirrneister et al., 2017) (i.e.  
 10 an epoching procedure that allows the extraction of a few overlapped crops of 2 s), resulting in EEG  
 11 patterns of shape (1,22,875) as input. This is at variance with the 0.5-2.5 s epoching of the MI dataset  
 12 adopted here for the other analyses (since such epoching was in agreement with other studies  
 13 (Lawhern et al., 2018; Lotte, 2015; Sakhavi et al., 2015), see also Section 2.2.2). Therefore, for each  
 14 CNN, the trialwise training on the MI dataset had to be performed also with the 0.5-4 s epoching to  
 15 evaluate the effect of cropped training against trialwise training. Crops of 2 s (corresponding to 500  
 16 time samples) with a stride of 0.5 s (corresponding to 125 time samples) were extracted for each trial  
 17 and these crops represented the CNN inputs. For each subject, this cropping procedure resulted in 6  
 18 crops (1,44,500) per trial for the ME-EEG signals and 4 crops (1,22,500) per trial for the MI-EEG  
 19 signals, augmenting the available dataset. Adopting this training strategy, the CNNs output one  
 20 prediction for each crop and thus several crop predictions belong to the same trial. To further  
 21 regularize CNNs trained with cropped training, the same loss function designed by Schirrneister et  
 22 al. (2017), named “tied sample loss function” (Equation 7) was employed. In particular, the cross-  
 23 entropy of two neighbouring crop predictions is added to the usual negative log likelihood of the  
 24 labels to drive the optimization towards more stable features across crops. Let us denote with  $t_c$  the  
 25 start frame of the  $c$ -th crop, with  $T$  the crop size (i.e. number of crop temporal samples) and with  
 26  $X_{i,c}^{(s)} = X_i^{(s)}[:, :, t_c : t_c + T]$  the  $c$ -th crop ( $0 \leq c \leq 5$  and  $0 \leq c \leq 3$  for the ME- and MI-EEG signals,

1 respectively) belonging to the  $i$ -th trial of the  $s$ -th subject. Hence, the loss was modified to depend  
 2 also on the prediction for the next crop  $c + 1$ :

$$\begin{aligned}
 3 \quad \text{loss} \left( y_i^{(s)}, p(l_k | X_{i,c}^{(s)}, \theta^{(s)}) \right) &= \sum_{k=0}^{N_c-1} -\log \left( p(l_k | X_{i,c}^{(s)}, \theta^{(s)}) \right) \cdot \delta(y_i = l_k) + \\
 4 \quad &\quad \sum_{k=0}^{N_c-1} -\log \left( p(l_k | X_{i,c}^{(s)}, \theta^{(s)}) \right) \cdot p(l_k | X_{i,c+1}^{(s)}, \theta^{(s)}). \quad (7)
 \end{aligned}$$

5 Except for the loss function, cropped training follows the setting adopted for the trialwise  
 6 training, sharing the same hyper-parameters (e.g. same optimizer, regularizers, learning rate, mini-  
 7 batch size, etc.) and the same two-runs training procedure. Cropped training was applied to Sinc-  
 8 ShallowNet (its basal version, see Table 1) and to the other three re-implemented CNNs.

## 9 2.5. Regularization

10 In addition to early stopping and cropped training which act as regularizers, other regularizing  
 11 techniques were used and implicitly integrated in Sinc-ShallowNet, as specified in its description (see  
 12 Sections 2.3.2, 2.3.3, 2.3.4). These are highlighted here:

- 13 i. Dropout (Srivastava et al., 2014). This technique randomly sets the outputs of the previous layer  
 14 to zero with a probability  $p$ , during each training update. This helps to prevent co-adaptation (i.e.  
 15 that some neurons are highly dependent to others) which could lead to overfitting. In the proposed  
 16 network, dropout with  $p = 0.5$  was introduced in block 2 immediately after the average pooling  
 17 layer.
- 18 ii. Batch normalization (Ioffe & Szegedy, 2015). This technique mitigates a phenomenon named  
 19 “internal covariate shift”, i.e. the change in the distribution of the layers’ activation due to the  
 20 change of the trainable parameters during training (Ioffe & Szegedy, 2015). This phenomenon  
 21 hinders the learning since the layers continuously need to adapt to the changed distribution while  
 22 training and is particularly severe in deep neural networks. Batch normalization reduces the  
 23 internal covariate shift, and consequently accelerates the training, by normalizing the output  
 24 feature maps of intermediate layers to zero mean and unit variance across each training mini-

1 batch. This technique introduces two trainable parameters since the normalization is followed by  
2 a channelwise affine transformation (that serves to maintain the expressive power of the  
3 network), whose parameters of scaling and shift are learned during training. Batch normalization  
4 enables higher learning rates without the risk of divergence, reduces the influence of a specific  
5 initialization scheme on the training, and also regularizes the model (Ioffe & Szegedy, 2015).  
6 While this technique is commonly used in deep neural networks, also shallow neural networks  
7 adopting batch normalization have been proposed in the literature. In particular, shallow CNNs  
8 including batch normalization have been recently applied to EEG signals for ME and MI  
9 decoding tasks (Lawhern et al., 2018; Schirrmeister et al., 2017), and for P300 detection (Liu et  
10 al., 2018). Importantly, Schirrmeister et al. (2017) reported an improved performance both in  
11 their shallow and deeper neural networks when using batch normalization compared to omitting  
12 it. Motivated by these previous results, we adopted this technique in our shallow CNN (blocks  
13 1, 2) too, by applying it to the output of the convolutional layer immediately before the non-  
14 linearity, as recommended in the original paper (Ioffe & Szegedy, 2015), with a momentum term  
15 of  $m = 0.99$  and with  $\varepsilon = 1e - 3$  for numerical stability.

16 iii. Kernel max-norm regularization. This technique constraints the norm of the trainable parameters  
17 to be upper bounded by a fixed constant  $c$ . Typically, it improves the performance of mini-batch  
18 stochastic gradient descent training and it was found to be especially useful with dropout  
19 (Srivastava et al., 2014). This technique was applied to the spatial depthwise convolutional (block  
20 1) and to the fully-connected (block 3) layers similarly to (Lawhern et al., 2018), using  $c = 1$   
21 and  $c = 0.5$ , respectively.

22 These regularization techniques were also used in the other re-implemented CNNs, as proposed in  
23 their original formulation.

## 24 2.6. Classification performance and comparison with state-of-the-art approaches

1 The performance of Sinc-ShallowNet in its basal form (Table 1) was compared to the five  
2 variants (Table 2) and to the re-implemented SOA algorithms. The latter comprise three CNNs  
3 (EEGNet (Lawhern et al., 2018), DeepConvNet and ShallowNet (Schirrmeyer et al., 2017)) and one  
4 traditional machine learning approach (FBCSP (Ang et al., 2008)+rLDA).

5 The three SOA CNNs (more details in Appendix A) include different convolutional modules,  
6 while keeping a single fully-connected layer in the classification module. EEGNet consists of three  
7 convolutional layers (one of them depthwise and one separable), DeepConvNet of five convolutional  
8 layers, and ShallowConvNet of two convolutional layers. The first two CNNs are general-purpose  
9 architectures; the last CNN is designed specifically for oscillatory signal classification, learning  
10 features related to log band-power by the introduction of a squaring nonlinearity, an average pooling  
11 layer and a log nonlinearity after the convolutional module. As EEGNet was designed for 128 Hz  
12 EEG signals (Lawhern et al., 2018), we multiplied the lengths of its temporal kernels and pooling  
13 sizes by a scaling factor of 2 to learn features coherently with the sampling frequency used here (a  
14 similar procedure was adopted in (Lawhern et al., 2018) when previous CNNs were re-implemented  
15 for comparison with EEGNet). Then, as explained in Sections 2.4.1 and 2.4.2, these CNNs were  
16 trained as Sinc-ShallowNet, with trialwise and cropped training strategies. Compared to Sinc-  
17 ShallowNet (in its basal form having 13828 and 5508 trainable parameters in case of ME- and MI-  
18 EEG signals, respectively), the other three CNNs (EEGNet, ShallowConvNet and DeepConvNet)  
19 have a total number of trainable parameters of 2604, 82564, 298229 in case of ME-EEG signals, and  
20 of 1932, 40644, 278079 in case of MI-EEG signals, respectively. EEGNet and ShallowConvNet are  
21 both shallow architectures, the first one having an extremely low number of trainable parameters due  
22 to the low number of temporal kernels adopted in the first layer ( $K_1 = 8$ ) and the use of depthwise  
23 and separable convolutions. These two architectures were chosen as reference shallow architectures  
24 (both general-purpose and specific for sensorimotor rhythm classification) to be compared with Sinc-  
25 ShallowNet. DeepConvNet was chosen as reference deep architecture (general-purpose) to be  
26 compared with Sinc-ShallowNet.

1 The traditional decoding pipeline adopted included FBCSP – a commonly used algorithm in EEG  
2 decoding and the winner of the BCI competition IV datasets 2a and 2b – coupled with rLDA. More  
3 details about the implementation of FBCSP+rLDA can be found in Appendix B. This algorithm was  
4 used as the best-performing traditional approach in movement-related EEG decoding to be compared  
5 with Sinc-ShallowNet.

6 We adopted the decoding accuracy as performance metric of the classifiers; furthermore, for  
7 completeness, the confusion matrices of basal Sinc-ShallowNet and the benchmark traditional  
8 approach FBCSP+rLDA were computed. Wilcoxon signed-rank test was used to check for a  
9 statistically significant difference between the contrasted conditions. To correct for multiple tests, a  
10 false discovery rate correction at  $\alpha = 0.05$  using the Benjamini-Hochberg procedure (Benjamini &  
11 Hochberg, 1995) was applied.

## 12 2.7. Interpretation

13 Post-hoc interpretation techniques were applied to Sinc-ShallowNet (in its basal version) at the  
14 end of the optimization. These include temporal and spatial kernel visualizations and an additional  
15 gradient-based technique, denoted as “temporal sensitivity analysis” (since it is applied to the features  
16 learned by the temporal sinc-convolutional layer).

### 17 2.7.1. Temporal and spatial kernels visualization

18 The visualization of the learned kernels of the first block allows the interpretation of the temporal  
19 and spatial convolutional layers. The temporal sinc-convolutional layer introduced in the Sinc-  
20 ShallowNet architecture allows a direct interpretation of the learned parameters, which are the lower  
21 and upper cutoff frequencies  $f_{1,j}$  and  $f_{2,j}$  of the  $K_1$  band-pass filters. Hence, for each subject, the  
22 distribution of the learned temporal kernels can be visualized by displaying how their passbands are  
23 distributed within the frequency range of the input signals (i.e. (4,125] Hz for ME- and (4,38] Hz for  
24 MI-EEG signals), and the preferred EEG rhythm (e.g.  $\alpha$ ,  $\beta$ , etc.) can be immediately derived. In  
25 particular, the following EEG bands  $b$  were considered:  $\theta = (4,8]$  Hz,  $\alpha = (8,12]$  Hz,  $\beta = (12,30]$  Hz,

1 low  $\gamma = (30,50]$  Hz, high  $\gamma = (50, 125]$  Hz. A temporal filter was considered belonging to a specific  
2 band  $b$  if its central frequency fell within that band (actually, in most cases the band-pass filters had  
3 narrow passbands totally falling within a specific band range, see also Section 3.3 in Results).

4 Moreover, since the spatial depthwise convolution applies separate spatial kernels to each  
5 temporally-filtered version of the input, the learned spatial kernels can be interpreted as the spatial  
6 features associated to a specific band-pass filter and can be visualized as scalp maps. Since we were  
7 interested in the evaluation of the discriminant power at the level of single electrode, here the absolute  
8 spatial kernel values were considered, as done by (Cecotti & Graser, 2011). This visualization was  
9 limited to the spatial filters related to the more relevant and more class-specific band-pass filters  
10 (selected as described in Section 2.7.2).

### 11 2.7.2. Temporal sensitivity analysis

12 The visualization of the learned band-pass filters (see Section 2.7.1) provides information about  
13 their frequency-range preference but does not provide any information about their importance for the  
14 classification task. Hence, in order to quantify the relevance of the band-pass filters for the  
15 classification task, we designed the temporal sensitivity analysis inspired by the saliency maps  
16 (Simonyan et al., 2013). This analysis allows the quantification of the importance of the different  
17 temporal kernels based on the gradient values, as described in the following (for simplicity, here we  
18 omit the superscript  $s$  referring to the specific subject).

19 *1. Gradient computation.* Given a class  $k$  of interest and the  $i$ -th test trial of the  $s$ -th subject  $X_i \in \mathbb{R}^{C \times T}$   
20 as input, let  $Y_j \in \mathbb{R}^{C \times T_1}$  ( $Y_{i,j}$  when  $X_i$  is fed as input) be the output of the  $j$ -th temporal kernel (i.e. the  
21  $j$ -th feature map) of the sinc-convolutional layer and  $z_k = h_k(X; \theta) \in \mathbb{R}^{N_c}$  ( $z_{i,k}$  when  $X_i$  is fed as  
22 input) be the class score (i.e. output of the block 3 fully-connected layer, immediately before the  
23 softmax activation function). The class score  $z_k$  is a highly non-linear function of  $Y_j$ ; given the input  
24 test trial  $X_i$ , this function can be approximated by a linear function in the neighbourhood of  $Y_{i,j}$  by  
25 computing the first-order Taylor expansion (Simonyan et al., 2013) (Equation 8):

$$1 \quad \begin{cases} z_k = z_k(Y_j) \approx G_{i,j,k}^{*T} \cdot Y_j^* + b_{i,j,k} \\ G_{i,j,k}^* = \left. \frac{\partial z_k}{\partial Y_j} \right|_{Y_{i,j}^*} \end{cases} \quad (8)$$

2        In the Equation 8, the superscript \* denotes a vectorized form (column vector), superscript  $T$   
3 represents the transposition of the vector, and  $b_{i,j,k}$  a bias term. In this linearized expression, the  
4 magnitude of each element of  $G_{i,j,k}^*$  quantifies how much the corresponding spatio-temporal sample  
5 within the  $j$ -th feature map (i.e. the  $j$ -th temporally filtered version of the input trial) affects the score  
6 for the  $k$ -th class  $z_k$  when presenting the input  $X_i$ . In other words, this quantifies how the value of an  
7 output category (e.g. output of the neuron related to class “Right Hand”) changes with respect to a  
8 small change in the temporally filtered EEG signals.

## 9    2. Gradient processing

- 10        a) For each  $G_{i,j,k}$  (i.e.  $\forall i, j, k$ ), the absolute value  $|G_{i,j,k}|$  was computed and averaged across  
11 the spatial and temporal dimension to obtain a scalar value  $\overline{|G_{i,j,k}|}$ .
- 12        b) Quantities  $\overline{|G_{i,j,k}|}$  related to trials belonging to each specific class were averaged together,  
13 resulting in the absolute gradient value  $g_{j,k}$  (scalar value):

$$14 \quad g_{j,k} = \frac{1}{M_k} \sum_i \overline{|G_{i,j,k}|} \quad (9)$$

15        In Equation 9, the **sum runs over the**  $M_k$  trials belonging to the class  $k$ , i.e.  $\{i : y_i = k\}$ .  
16 Hence,  $g_{j,k}$  quantifies how much, on average, the  $j$ -th temporal filter affects the score of the  
17 class  $k$ .

- 18        c) The gradients  $g_{j,k}$  (Equation 9) were normalized dividing by the maximum across the classes  
19 and kernels (Equation 10):

$$20 \quad \hat{g}_{j,k} = \frac{g_{j,k}}{\max_{j,k} g_{j,k}} \quad (10)$$

21        This was done in order to facilitate the comparison across kernels and classes.

1 Then, the normalized gradients  $\hat{g}_{j,k}$  from Equation 10 were further processed in two ways  
 2 for different purposes (d.1 and d.2).

3 d.1) *Temporal sensitivity analysis at the level of EEG bands* – For each considered EEG band  $b$ ,  
 4  $\hat{g}_{j,k}$  were averaged across the band-pass filters belonging to a specific EEG band  $b$  (see  
 5 Section 2.7.1). The resulting score  $\hat{g}_{b,k}$  (Equation 11) quantifies the overall importance of  
 6 the specific band  $b$  for the classification of the specific class  $k$ :

$$7 \quad \hat{g}_{b,k} = \frac{1}{K_{1,b}} \sum_j \hat{g}_{j,k}. \quad (11)$$

8 In Equation 11, the sum runs over the  $K_{1,b}$  band-pass filters belonging to the  $b$  band, i.e.

$$9 \quad \left\{ j : f_{c,j} = \frac{f_{1,j} + f_{2,j}}{2} \in (f_{1,b}, f_{2,b}] \right\}, \text{ where } (f_{1,b}, f_{2,b}] \text{ denotes the frequency range of the band.}$$

10 d.2) *Temporal sensitivity analysis at the level of single band-pass filter* – This step was  
 11 introduced to select the more relevant and more class-specific band-pass filters (i.e. the  
 12 filters that are relatively more discriminative for a specific class than for the other classes)  
 13 and to limit the visualizations of the learned spatial features to these selected temporal filters.  
 14 Indeed, the normalized gradients  $\hat{g}_{j,k}$  (Equation 10) corresponding to a specific temporal  
 15 filter, can assume large values across all classes, indicating a large importance in the use of  
 16 that temporal filter shared across the classes. To emphasize the specificity of each filter for  
 17 a single class or a subset of classes, the gradient  $\hat{g}_{j,k}$  was rescaled. The rescaling (Equation  
 18 12) was designed so that a gradient resulting higher (or lower) for a specific class than for  
 19 the other classes on average, was scaled more (or less). This way, the differences of the filter  
 20 relevance across the classes were emphasized:

$$21 \quad \begin{cases} \hat{g}'_{j,k} = \gamma_{j,k} \cdot \hat{g}_{j,k}, \\ \gamma_{j,k} = \frac{3 \cdot \hat{g}_{j,k}}{\sum_{m=0, m \neq k}^3 \hat{g}_{j,m}}. \end{cases} \quad (12)$$

22 Based on this scaling, the quantity  $\hat{g}'_{j,k}$  assumes larger values ( $\gamma_{j,k} > 1$ ) when the impact of  
 23  $j$ -th temporal filter on the score of the specific class  $k$  is higher than its average impact on

1 the other three classes; vice versa it assumes lower values ( $\gamma_{j,k} < 1$ ) when the j-th temporal  
2 filter impacts on average more on the other three classes than on the considered  $k$  class.  
3 Therefore, given a class  $k$ , filters having  $\hat{g}'_{j,k} > \hat{g}_{j,k}$  (i.e. with  $\gamma_{j,k} > 1$ ) represent the filters  
4 having a discriminative power relatively heavier for that class than for the other classes on  
5 average. Thus, considering a class  $k$ , the more relevant and more class-specific temporal  
6 band-pass filters can be identified as the filters with  $\gamma_{j,k} > 1$  and that scored higher  $\hat{g}'_{j,k}$   
7 values. Lastly, the spatial kernels associated with the so selected band-pass filters can be  
8 visualized as described in Section 2.7.1.

### 9 3. RESULTS

#### 10 3.1. Classification performance and comparison with state-of-the-art approaches

11 In this section, the performances of the basal Sinc-ShallowNet (trained via trialwise strategy) are  
12 compared with the traditional machine learning algorithm and with the three re-implemented CNNs  
13 (trained via trialwise strategy).

14 Figure 3 reports the confusion matrices obtained with the proposed architecture and with the  
15 machine learning algorithm FBCSP+rLDA, with ME- and MI-EEG signals. Each of these matrices  
16 represents the confusion matrix across the subject-specific classifiers. Denoting with  $i$  and  $j$  the  $i$ -th  
17 row and  $j$ -th column, the entry in the  $(i,j)$  location represents the total number of test trials across  
18 subjects predicted as class  $i$  when the true class is  $j$  (together with the % ratio between this number  
19 and the total number of trials for each class  $j$ ). For each  $(i,j)$  location (16 in total), a Wilcoxon signed-  
20 rank test was performed between the entries of the subject-specific confusion matrices obtained with  
21 FBCSP+rLDA and with Sinc-ShallowNet, separately for the two datasets; that is, for each  $(i,j)$   
22 location, we compared two samples of 14 values in case of the ME dataset and two samples of 9  
23 values in case of the MI dataset. In order to correct for multiple comparisons (16 in total within each  
24 dataset), the Benjamini-Hochberg procedure was applied. The corrected p-value resulting from each

1 comparison is displayed inside the corresponding cell of the matrices reporting Sinc-ShallowNet  
2 results (matrices on the right in Figure 3).

3 [Figure 3 about here.]

4 The confusion matrices were similar between the approaches, with only 4 entries significantly  
5 different ( $P < 0.05$ ) in case of ME-EEG signals. In particular, Sinc-ShallowNet classified  
6 significantly better “Left Hand” and “Feet” classes ( $P = 0.036$ ) and produced a significantly lower  
7 number of misclassifications between “Right Hand” and “Rest” classes. In both algorithms, the  
8 majority of the misclassifications were associated with a wrong discrimination between “Right  
9 Hand”-“Left Hand” classes (110 misclassified trials for FBCSP+rLDA and 90 for Sinc-ShallowNet)  
10 in case of ME-EEG signals, and between “Right Hand”-“Left Hand” classes (196 misclassified trials  
11 for FBCSP+rLDA and 179 for Sinc-ShallowNet) and “Feet”-“Tongue” classes (181 misclassified  
12 trials for FBCSP+rLDA and 160 for Sinc-ShallowNet) in case of MI-EEG signals.

13 Tables 3 and 4 show the accuracies obtained with Sinc-ShallowNet, the three SOA CNNs, and  
14 the algorithm FBCSP+rLDA on ME- and MI-EEG signals, respectively. Results of the statistical  
15 analyses are reported too.

16 [Table 3 about here.]

17 [Table 4 about here.]

18 The proposed architecture scored an accuracy across subjects (mean  $\pm$  std) of  $91.2 \pm 9.1$  % (inferior  
19 only to ShallowConvNet) and of  $72.8 \pm 12.9$  % (best overall) on ME- and MI-EEG signals,  
20 respectively. Compared to the baseline FBCSP+rLDA algorithm, ShallowConvNet and Sinc-  
21 ShallowNet performed significantly better on both ME- ( $P = 0.024$ ,  $P = 0.024$ , respectively) and  
22 MI-EEG signals ( $P = 0.046$ ,  $P = 0.031$ , respectively). Sinc-ShallowNet significantly outperformed  
23 DeepConvNet ( $P = 0.026$ ) on ME-EEG signals, and both EEGNet ( $P = 0.027$ ) and DeepConvNet  
24 ( $P = 0.027$ ) on MI-EEG signals. Lastly, ShallowConvNet significantly outperformed Sinc-  
25 ShallowNet ( $P = 0.040$ ) on ME-EEG signals; however, regarding this point, further considerations

1 can be drawn from the results of the post-hoc hyper-parameter evaluation (see Section 4.2 in the  
2 Discussion).

### 3 **3.2. Post-hoc hyper-parameter evaluation and training strategy evaluation**

4 The performance obtained with the basal Sinc-ShallowNet with ME- and MI-EEG signals was  
5 compared to the Sinc-ShallowNet variants, obtained by changing the hyper-parameters  $K_1, D_2, F_p, S_p$   
6 and by introducing an additional pointwise convolutional layer as first layer in block 2 (see Section  
7 2.3.5). Specifically, each variant was obtained by changing one hyper-parameter at a time while  
8 keeping the other hyper-parameters unchanged (see Table 2). In this comparison, both the basal Sinc-  
9 ShallowNet and each variant were trained adopting the trialwise training strategy (see Section 2.4.1).  
10 The overall effect of each hyper-parameter change was quantified jointly on ME- and MI-EEG signals  
11 by computing the difference in accuracy between the tested (variant) and basal configurations  $\Delta_{acc} =$   
12  $acc_{tested} - acc_{ref}$  (e.g.  $\Delta_{acc} = acc_{K_1=8} - acc_{K_1=32}$  for the comparison “ $K_1 = 8 - K_1 = 32$ ”,  
13 contrasting the configuration with  $K_1 = 8$  temporal filters and the basal configuration having  $K_1 =$   
14  $32$  filters). The results are shown in Figure 4a: a significant worsening of the performance occurred  
15 when  $K_1$  decreased ( $P = 0.005$  and  $P = 0.010$  when comparing  $K_1 = 8$  vs  $K_1 = 32$  and  $K_1 = 8$  vs  
16  $K_1 = 16$ , respectively), while no significant effect was induced by the other hyper-parameter  
17 changes.

18 We evaluated the impact of cropped training compared to trialwise training on Sinc-ShallowNet  
19 (in its basal configuration) and on each re-implemented SOA CNNs. As detailed in Section 2.4.2, the  
20 trialwise training strategy adopted for this analysis was designed with a different epoching of the MI-  
21 EEG signals (0.5-4 s rather than 0.5-2.5 s as adopted in the rest of the presented results) in order to  
22 follow the procedure used in (Schirrneister et al., 2017). Nevertheless, we verified that no statistically  
23 significant difference in performance emerged between the trialwise training implemented with the  
24 different epoching of MI-EEG signals ( $P = 0.441, P = 0.345, P = 0.347, P = 0.346$ , respectively  
25 for DeepConvNet, ShallowConvNet, Sinc-ShallowNet and EEGNet.). The overall effect of cropped

1 training on each CNN was quantified jointly on ME- and MI-EEG signals by computing the  
2 difference in accuracy between the cropped and the trialwise training strategies  $\Delta_{acc} = acc_{cropped} -$   
3  $acc_{trialwise}$ . The corresponding results are shown in Figure 4b. Only the deep architecture  
4 DeepConvNet significantly benefited from the cropped training strategy ( $P = 0.002$ ), while  
5 shallower architectures such as Sinc-ShallowNet and EEGNet performed significantly worse when  
6 trained with the cropped strategy ( $P = 0.008$  and  $P = 0.009$ ).

7 [Figure 4 about here.]

### 8 3.3. Interpretation

9 In order to illustrate feature interpretability and feature relevance evaluation enabled by the  
10 proposed approach, we provide the results of the interpretation techniques for one representative  
11 subject for each dataset (ME- and MI-EEG signals). These results refer to the basal Sinc-ShallowNet  
12 trained with the trialwise training strategy.

13 Figures 5a and 6a display the distribution of the temporal filters learned by the network for a  
14 specific subject in case of the ME- and MI-EEG signals, respectively. Most of the temporal band-  
15 pass filters belonged to specific EEG bands (a filter is considered belonging to an EEG band based  
16 on its central frequency, see Section 2.7.1). The learned band-pass filters mainly belonged to the  $\beta$ ,  
17 low  $\gamma$  and high  $\gamma$  bands in case of ME-EEG signals (Figure 5a) and to the  $\alpha$ ,  $\beta$  and low  $\gamma$  bands in case  
18 of the MI-EEG signals (Figure 6a). The corresponding gradients  $\hat{g}_{b,k}$  (see Equation 11 in Section  
19 2.7.2) obtained from the temporal sensitivity analysis at the level of EEG bands are displayed in  
20 Figures 5b and 6b. These visualizations suggest that the classification tasks rely differently on the  
21 EEG bands depending on the class. The high  $\gamma$  band resulted the most important EEG band for each  
22 class of ME-EEG signals (Figure 5b) in addition to the  $\beta$  band – for the “Right Hand” and “Left  
23 Hand” classes – and low  $\gamma$  band for the “Rest” and “Feet” classes. The  $\beta$  band resulted relevant for  
24 each class of MI-EEG signals (Figure 6b) in addition to the  $\alpha$  band – in particular for the “Left Hand”

1 but also for the “Right Hand” classes – and low  $\gamma$  in particular for “Tongue” and also for “Feet”  
2 classes.

3 [Figure 5 about here.]

4 [Figure 6 about here.]

5 Figures 7 and 8 report the results of the temporal sensitivity analysis performed at the level of  
6 the single band-pass filter for each decoded class, as to the same exemplary cases of Figures 5 and 6  
7 (ME- and MI-EEG signals, respectively). In each panel (bar plot), both the normalized gradient  $\hat{g}_{j,k}$   
8 (Equation 10, length of the black line) and the rescaled gradient  $\hat{g}'_{j,k}$  (Equation 12, length of the  
9 coloured bar), are displayed for each learned filter, together with the indication (colour-coded) of the  
10 band the filter belong to. By looking at  $\hat{g}_{j,k}$ , the filters belonging to each band assumed different  
11 importance depending on the class, in agreement with Figures 5b and 6b. For example, as to Figure  
12 7, filters in the low  $\gamma$  band had on average larger values of  $\hat{g}_{j,k}$  for the “Rest” and “Feet” classes than  
13 for the “Hand” classes. Moreover, within each class, filters in the high  $\gamma$  band had on average larger  
14 values of  $\hat{g}_{j,k}$  compared to filters in the other bands, especially for the “Rest” and “Feet”. However,  
15 by looking at the single filters, some of them had very similar gradient values  $\hat{g}_{j,k}$  across all classes  
16 (for example filters #26, #28, #30 in Figure 7a, and filters #1, #7 in Figure 8b). The rescaled gradient  
17  $\hat{g}'_{j,k}$  allows the identification of the more relevant and more class-specific band-pass filters, as  
18 described in Section 2.7.2. Specifically, for each of the two more discriminative EEG bands (as  
19 obtained via the temporal sensitivity analysis at the level of EEG bands, Figures 5b and 6b), the two  
20 more relevant band-pass filters were selected as the two filters (belonging to that band) that scored  
21 the two highest values of  $\hat{g}'_{j,k}$  with  $\hat{g}'_{j,k} > \hat{g}_{j,k}$ . For the so-selected temporal filters, the  $D_2 = 2$   
22 learned spatial filters were displayed as to their absolute values (insets within each panel of Figures  
23 7 and 8). The blue regions correspond to weights that are around 0 indicating electrode locations with  
24 a low discriminant power, and vice versa for the red regions. Thus, spatial filters extremely focalized  
25 to specific subsets of electrodes were learned for both the decoding tasks. In particular, a clear contra-

1 **laterality** in the scalp weight distributions can be observed in case of the hand movements (both  
2 executed and imagined) compared to the other classes.

3 [Figure 7 about here.]

4 [Figure 8 about here.]

#### 5 **4. DISCUSSION**

6 In this study Sinc-ShallowNet, a novel lightweight and interpretable CNN for EEG decoding,  
7 was designed and applied to motor execution and imagery tasks. The use of a band-pass filtering  
8 specialized convolutional layer (sinc-convolutional layer) and a spatial filtering with a reduced CNN  
9 channel connectivity (depthwise convolutional layer) enables the learning of band-pass filters and  
10 directly associated spatial filters. Thus, the proposed CNN is fully-interpretable and optimized in its  
11 convolutional module (i.e. feature extractor). In particular, the following points of strength can be  
12 emphasized:

- 13 i. Easy interpretation of both spectral and spatial features. The trainable parameters of the sinc-  
14 convolutional layer are directly interpretable (cutoff frequencies instead of mere kernel values  
15 **as** in a traditional convolutional layer) and the spatial filters are directly tied to specific band-  
16 pass filters.
- 17 ii. High optimization in terms of number of trainable parameters. The adopted sinc-convolution  
18 **trains** only 2 cutoff frequencies for each temporal filter **and the** depthwise convolution **reduces**  
19 the connections across **the CNN channels**.
- 20 iii. Computational efficiency. Due to the symmetry of the parametrized function adopted in the  
21 sinc-convolution, only half of the kernel values need to be computed.

22 In addition, the interpretation of the learned spectral and spatial features was further enriched  
23 thanks to the temporal sensitivity analysis; **this analysis** allows the identification of the more  
24 discriminative EEG bands (temporal sensitivity analysis at the level of EEG bands), and the more

1 relevant and more class-specific band-pass filters (temporal sensitivity analysis at the level of single  
2 band-pass filter) together with their spatial distribution.

### 3 **4.1. Classification performance and comparison with state-of-the-art approaches**

4 The results on the ME and MI decoding tasks suggest that Sinc-ShallowNet significantly  
5 outperformed the traditional FBCSP+rLDA decoding pipeline. Among the re-implemented SOA  
6 CNNs, only ShallowConvNet (but not DeepConvNet and EEGNet) performed significantly better  
7 than the traditional machine learning approach, in agreement with results by Schirrneister et al.  
8 (2017).

9 By comparing Sinc-ShallowNet with the re-implemented CNNs, the following considerations  
10 can be drawn. First, ShallowConvNet significantly outperformed Sinc-ShallowNet on ME- but not  
11 on MI- EEG signals (see Table 3). This is the only case in which Sinc-ShallowNet performed worse  
12 compared to the other considered CNNs. Nevertheless, it is worth noticing that Sinc-ShallowNet  
13 introduces 13828 and 5508 trainable parameters, that corresponds only to the 16.7% and 13.6% of  
14 those introduced by ShallowConvNet in case of ME- and MI-EEG signals (82564 and 40644),  
15 respectively. Therefore, the proposed architecture finalized the classification tasks in a more  
16 computationally efficient way, by introducing a lower number of trainable parameters. Furthermore,  
17 ShallowConvNet architecture was developed specifically for sensorimotor rhythm classification  
18 forcing the extraction of log band-power features (task-specific CNN), while Sinc-ShallowNet was  
19 not restricted to specific feature learning. Second, in the comparison with a general-purpose shallow  
20 architecture (EEGNet), Sinc-ShallowNet performed significantly better on MI-EEG signals, while  
21 performed comparably on ME-EEG signals. The lower performance of EEGNet may derive from the  
22 extremely lightweight architecture that used only  $K_1 = 8$  temporal filters. Accordingly, the decoding  
23 of MI-EEG signals may benefit from a higher number of temporal filters (e.g. 32 as in the architecture  
24 proposed here). The introduction of the temporal sinc-convolutional layer that reduces the number of  
25 trainable parameters (i.e. only the two cutoff frequencies for each temporal filter) may be particularly  
26 beneficial for the decoding of MI-EEG dataset. Indeed, this dataset is characterized by a low number

1 of training examples that requires the number of trainable parameters to be carefully maintained  
2 limited **in order** to avoid overfitting and achieve a good fit. Furthermore, when comparing Sinc-  
3 ShallowNet with DeepConvNet, the first provided significantly higher decoding accuracy on both  
4 ME- and MI-EEG signals. This may be attributable to the higher number of trainable parameters  
5 introduced by DeepConvNet (298229 and 278079 in case of ME- and MI-EEG signals, respectively),  
6 leading to an architecture more prone to overfitting especially in case of small datasets as for the  
7 adopted MI dataset.

#### 8 **4.2. Design choices of Sinc-ShallowNet**

9 The post-hoc hyper-parameter evaluation (Figure 4a), revealed a significant negative effect of  
10 lowering  $K_1$  on Sinc-ShallowNet performance, with an average  $\Delta_{acc} = -4\%$  and  $\Delta_{acc} = -2\%$ , when  
11 using 8 and 16 band-pass filters compared to 32 filters, respectively. Thus, Sinc-ShallowNet benefits  
12 from an increased set of band-pass filters that enrich the temporally filtered representation of the  
13 input. Furthermore, Sinc-ShallowNet performance on both datasets when using  $K_1 = 8$  was not  
14 different from EEGNet that **uses this** number of temporal filters.

15 The analysis on  $D_2$  and on the optional recombination deserves some comments. Increasing  $D_2$   
16 did not lead to significant increase in the performance. However, it is interesting to note that when  
17 the effect of  $D_2$  was disaggregated between the two datasets (ME-EEG and MI-EEG dataset), an  
18 opposite **behaviour** tends to appear, with an average  $\Delta_{acc} = +0.4\%$  and  $\Delta_{acc} = -0.2\%$  on ME- and  
19 MI-EEG signals respectively (although not statistical significance was reached in either dataset). This  
20 different **behaviour** might be explained considering that when a CNN is trained with EEG signals  
21 containing a lower number of frequency components (such as MI-EEG signals), the band-pass  
22 temporal filters lie into a **narrower** frequency range and thus the probability that two different  
23 temporal filters have similar cutoff frequencies is higher. In this scenario, a lower number of spatial  
24 filters ( $D_2$ ) for each temporal filter could be sufficient to retain enough capacity of the CNN, because  
25 **close** temporal filters could compensate for the lower  $D_2$ . Indeed, different spatial filters could be  
26 learned for similar temporal filters obtaining a cumulative set (across similar temporal filters) of band-

1 specific spatial filters. Conversely, ME-EEG signals having wider frequency content can benefit from  
2 a larger number  $D_2$  of spatial filters. Recombining the spatial activations via an additional pointwise  
3 convolutional layer did not improve accuracy. However, in this case too, by disaggregating the effect  
4 on the two datasets, an opposite behaviour tends to appear with an average  $\Delta_{acc} = +0.5\%$  and  $\Delta_{acc} =$   
5  $-2.6\%$  in case of ME- and MI-EEG signals respectively (although not statistical significance was  
6 reached in either dataset). This may be due to the learning of a useful recombination of frequency-  
7 specific spatial features learned across a wide frequency range, in case of signals with broad  
8 frequency content as ME-EEG signals. Finally, it is worth noticing that both increasing  $D_2$  and  
9 including a pointwise convolutional layer lead to an increase in the number of trainable parameters  
10 that might be critical in applications involving small datasets (e.g. the adopted MI dataset). Overall,  
11 these considerations remain quite speculative and further experiments are required, for example  
12 testing Sinc-ShallowNet and its different design choices on other datasets having larger and smaller  
13 size than those used here and having various frequency contents. However, it is interesting to note  
14 that the small accuracy increase ( $\Delta_{acc} = +0.5\%$ ) in case of ME-EEG signals obtained introducing the  
15 pointwise convolutional layer led to a significant better performance of Sinc-ShallowNet compared  
16 to EEGNet ( $P = 0.046$ ) and to comparable performance with ShallowConvNet ( $P = 0.090$ ); at the  
17 same time, the accuracy decrease in case of MI-EEG signals ( $\Delta_{acc} = -2.6\%$ ) did not change the  
18 statistical significance ( $P = 0.049$  vs. EEGNet and DeepConvnet,  $P = 0.340$  vs. ShallowConvNet).  
19 Thus, the proposed Sinc-ShallowNet architecture integrated with the recombination of the spatial  
20 activations led to a CNN that performs better than or at least as well as the SOA CNNs on both  
21 datasets, at the expense of the number of trainable parameters (17924 and 9604 in case of ME and  
22 MI datasets respectively).

23 Lastly, changing the average pooling strategy by using larger pool and stride sizes did not affect  
24 the performance.

1 In conclusion, this analysis suggests that the proposed Sinc-ShallowNet in its basal version (see  
2 Table 1) resulted in a good compromise between performance and parsimony with enough capacity  
3 to solve both the decoding tasks.

### 4 **4.3. Training strategies**

5 The overall effect of the training strategy on the performance metric (Figure 4b) resulted in a  
6 significantly increase of the decoding accuracy for a deeper architecture as DeepConvNet (on average  
7  $\Delta_{acc} = +4.6\%$ ), while a significant worsening of the performance was observed as the CNN  
8 architecture becomes shallower and more lightweight (no significant effect on ShallowConvNet,  
9  $\Delta_{acc} = -2.9\%$  for Sinc-ShallowNet and  $\Delta_{acc} = -4.7\%$  for EEGNet on average ). This different  
10 behaviour of cropped training on shallow and deep architectures is in line with the results reported  
11 by Schirrneister et al. (2017) when examining ShallowConvNet and DeepConvNet, i.e. no  
12 improvements for ShallowConvNet and significant improvement for DeepConvNet. The present  
13 study further confirmed those previous results and extended them to other shallow architectures (i.e.  
14 EEGNet and Sinc-ShallowNet). Thus, a data-intensive CNN (e.g. DeepConvNet) improved its  
15 performance with cropped training – which acts as a data augmentation procedure – while lightweight  
16 CNNs did not. In contrast to deeper network, shallow CNNs like EEGNet and Sinc-ShallowNet  
17 performed well in both the decoding tasks without the need of any data augmentation procedure that,  
18 conversely, worsened their performance.

### 19 **4.4. Interpretation**

20 The band-pass filters mainly belonged to the  $\beta$ , low  $\gamma$  and high  $\gamma$  EEG bands when the network  
21 was trained with ME-EEG signals (Figure 5a), and to the  $\alpha$ ,  $\beta$  and low  $\gamma$  EEG bands when the network  
22 was trained with MI-EEG signals (Figure 6a). The latter result agreed with that obtained by Lawhern  
23 et al. (2018) using EEGNet on the same decoded subject. In particular, Lawhern et al. (2018)  
24 estimated each band-pass filter learned by the temporal convolutional layer simply by counting the  
25 number of cycles of the specific temporal kernel in the corresponding temporal window. In Sinc-

1 ShallowNet, each band-pass filter is implicitly defined by the temporal sinc-convolutional layer that  
2 directly provides the two cutoff frequencies.

3 When the CNN was trained on ME-EEG signals, the temporal sensitivity analysis at the level of  
4 EEG bands (Figure 5b) indicates that the most relevant bands were  $\beta$ , high  $\gamma$  for the “Right Hand”  
5 and “Left Hand” classes, and low  $\gamma$ , high  $\gamma$  for the “Feet” and “Rest” classes. In addition, the high  $\gamma$   
6 band emerged as more important than the  $\beta$  and low  $\gamma$  bands for each decoded class, confirming the  
7 relevance not only of the  $\beta$  but also of the high  $\gamma$  band in the decoding task as previously evidenced  
8 by Schirrneister et al. (2017). Ball et al. (2008) found an increase in the high  $\gamma$  activity within the 60-  
9 90 Hz range, in addition to lower frequencies activity ( $\alpha$ ,  $\beta$ ), in human sensorimotor cortex during  
10 ME. Interestingly, in the exemplary case shown in Figure 5, most of the band-pass kernels belonging  
11 to the high  $\gamma$  band fell within this range (7 out of 10).

12 When the CNN was trained on MI-EEG signals, the temporal sensitivity analysis at the level of  
13 EEG bands (Figure 6b) indicates that the most relevant bands were  $\alpha$ ,  $\beta$  for the “Left Hand” and  
14 “Right Hand” classes, and  $\beta$ , low  $\gamma$  for the “Feet” and “Tongue” classes. These results are in line with  
15 previous studies showing that also the low  $\gamma$  band, together with the  $\alpha$  and  $\beta$  bands, provides  
16 information on MI (Crone et al., 1998). This was further confirmed by (Mirnaziri et al., 2013), where  
17 adding low  $\gamma$  features to  $\alpha$  and  $\beta$  features led to better performance using the same MI dataset.

18 Thanks to the use of spatial depthwise convolution, the proposed architecture ties spatial kernels  
19 to each band-pass filter and thus, the relevance, as quantified by the temporal sensitivity analysis, can  
20 be propagated from each band-pass filter to the associated spatial filters. In particular, the more  
21 relevant and more class-specific spatial filters can be identified – as those associated to the band-pass  
22 filters scored by the highest rescaled gradients  $\hat{g}'_{j,k}$ , (i.e. temporal sensitivity analysis at the level of  
23 single band-pass filter) – and visualized. These spatial filters show a highly localized distributions in  
24 the scalp maps (Figures 7a-7d and 8a-8d, respectively for ME- and MI-EEG signals). Among the  
25 spatial filters specific for the hand movements, some filters have the most discriminative electrodes  
26 located in the contralateral hemisphere to the executed and imagined hand movement, approximately

1 above the primary *sensorimotor* hand representation areas (i.e. around C3 and C4). Regarding the  
2 executed and imagined feet movements, some filters have the most discriminative electrodes located  
3 more centrally, approximately above the primary motor foot area (i.e. around CPz, Cz and FCz).  
4 Finally, regarding the imagined tongue movement, the most discriminative electrodes are placed not  
5 only around C3 and C4, but also approximately above the somatosensory cortex (i.e. area below Cz),  
6 representing the brain region triggered by the imagination of tongue movements (Zhao et al., 2019).

7 Therefore, by interpreting the features exploited by the network for the classification task, it turns  
8 out that Sinc-ShallowNet was capable of learning features related to known neurophysiological  
9 phenomena without relying on artefact or noise sources in the EEG signals.

10 As underlined previously, the interpretation capabilities of the network are provided by coupling  
11 an interpretable layer (sinc-convolutional layer) with an optimized layer (depthwise convolutional  
12 layer), and by using a post-hoc gradient-based technique alongside with spatial and temporal filter  
13 visualizations. Therefore, interpretation capabilities of Sinc-ShallowNet are intrinsically linked to  
14 some specific design choices and specifically implemented post-hoc analyses. However, other more  
15 general-purpose techniques adopted in our network (e.g. batch normalization or dropout), that  
16 introduce a regularization effect, contribute to increase the neurophysiological reliability of feature  
17 interpretation by improving the performance on unseen examples. For example, we verified that when  
18 training Sinc-ShallowNet by removing the batch normalization layers in the blocks 1, 2 (and leaving  
19 all the other hyper-parameters unchanged), a significant decrease of the decoding accuracies  
20 occurred:  $\Delta_{acc} = -4.8\%$  ( $P = 0.002$ , Wilcoxon signed-rank test),  $\Delta_{acc} = -14.8\%$  ( $P = 0.008$ ,  
21 Wilcoxon signed-rank test) respectively for ME- and MI-EEG signals, where  $\Delta_{acc} = acc_{w/o BN} -$   
22  $acc_{w/BN}$ . These simulations confirmed the important regularization introduced by batch  
23 normalization that significantly increased network accuracy on unseen examples. Accordingly,  
24 although batch normalization does not contribute directly to the interpretation capabilities of the  
25 network (omitting it the inner interpretation capabilities of the network are not altered), its inclusion

1 increases the neurophysiological significance of the interpreted features via accuracy improvement.  
2 Indeed, the band-pass filters and spatial filters learned by the batch-normalized Sinc-ShallowNet turn  
3 out to be more class-discriminative (as they provide higher accuracies). Therefore, the learned  
4 spectral and spatial features are more likely to reflect neurophysiological aspects (in terms of more  
5 relevant EEG bands and electrodes) linked to the investigated tasks (i.e. motor execution and motor  
6 imagery decoding).

7 Finally, we would like to provide some comments on other CNNs in the literature that adopt a  
8 non-traditional convolutional layer designed to perform a specific input transformation (here the sinc-  
9 convolutional layer forcing band-pass filtering). First, it is worth noticing that, at best of our  
10 knowledge, only two previous (and very recent) studies (Zeng et al., 2019; Zhao et al., 2019) include  
11 a similar layer within a CNN architecture, indicating that this represents an innovative and emerging  
12 approach in the field of EEG decoding. Zhao et al. (2019) proposed a CNN for MI classification  
13 including a time-frequency convolutional layer based on [wavelets](#) and interpreted the learned  
14 features. Differently from the architecture proposed here, they adopted a traditional spatial  
15 convolutional layer and tested the network only on MI decoding tasks. Comparing the decoding  
16 accuracy reported in the original paper (Zhao et al., 2019) with Sinc-ShallowNet accuracy on the  
17 same MI-EEG signals, Sinc-ShallowNet scored an average accuracy +5.8% with respect to the  
18 architecture proposed by [Zhao et al. \(2019\)](#), although without reaching statistical significance ( $P =$   
19  $0.086$ , Wilcoxon signed ranked test). However, the network by Zhao et al. (2019), [due to the adoption](#)  
20 [of a standard](#) spatial convolutional layer (that by itself involves 13775 trainable parameters, including  
21 bias), has a larger number of trainable parameters compared to Sinc-ShallowNet (1408 for MI-EEG  
22 signals). In an even more recent paper, Zeng et al. (2019) included a sinc-convolutional layer into a  
23 deep 1D CNN (3 convolutional layers and 4 fully-connected layers) for EEG emotion classification.  
24 The proposed solution appears more robust and more performing than other classifiers (and thus  
25 possibly confirming the potentiality of this kind of layer). However, the network by Zeng et al. (2019)

1 introduced a large number of trainable parameters, especially due to the use of 3 hidden fully-  
2 connected layers having thousands of neurons. Moreover, the authors did not face the interpretation  
3 of the learned features; in particular, the adoption of a reshaped input representation (2D-to-1D  
4 reshaping) and of traditional convolutions hinder the interpretability of the CNN. In future, it will be  
5 interesting to test Sinc-ShallowNet on the same decoding task tackled by Zeng et al. (2019).

## 6 **5. CONCLUSIONS**

7 In conclusion, we proposed a novel CNN named Sinc-ShallowNet, characterized by an  
8 interpretable and efficient (in terms of number of trainable parameters) convolutional module. This  
9 module includes a temporal sinc-convolutional layer, forcing the learning of band-pass filters with  
10 only two trainable parameters per filter, and a spatial depthwise convolution that learns spatial  
11 features tied to each band-pass filter. The proposed design provides direct interpretability of the  
12 learned spectral-spatial features, at the same time limiting the number of trainable parameters.  
13 Furthermore, a gradient-based technique (temporal sensitivity analysis) was introduced in order to  
14 identify the more relevant and more class-specific features. Overall, the proposed CNN, tested on  
15 motor execution and motor imagery EEG signals, outperformed other state-of-the-art CNNs and a  
16 traditional machine learning algorithm. The analyses on the design choices and training strategies  
17 confirmed that the proposed architecture is a good compromise between decoding performance and  
18 an efficient use of trainable parameters. The post-hoc interpretation techniques suggest that the  
19 features learned by the convolutional module matched well-known EEG motor-related activity, both  
20 in the frequency and spatial domains. While Sinc-ShallowNet was applied only to motor-related EEG  
21 decoding, it was not specifically tailored to decoding sensorimotor rhythm and may be used also in  
22 other EEG decoding tasks (e.g. P300 detection or other ERP classification tasks). Furthermore, if a  
23 specific decoding task benefits from deeper architectures, the interpretable and optimized  
24 convolutional module proposed in Sinc-ShallowNet could be easily employed to design deeper CNNs  
25 by stacking more convolutional layers on it. In particular, due to its augmented interpretability, Sinc-  
26 ShallowNet or a deeper CNN based on it, may be applied to investigate cognitive and/or motor aspects

- 1 for which the distinctive EEG correlates are less known (e.g. attention, emotion, creativity, movement
- 2 trajectory/kinematics etc.).

## 1 **APPENDIX A: State-of-the-art CNNs**

2 The SOA CNN architectures considered for the comparison with Sinc-ShallowNet are reported  
3 in Tables A.1, A.2 and A.3, respectively for EEGNet (Lawhern et al., 2018), DeepConvNet and  
4 ShallowNet (Schirrneister et al., 2017).

5 [Table A.1 about here.]

6 [Table A.2 about here.]

7 [Table A.3 about here.]

## 8 **APPENDIX B: FBCSP+rLDA**

9 As traditional machine learning decoding algorithm, we used a pipeline previously validated and  
10 adopted in Schirrneister et al. (2017). Two different overlapped filter banks were designed for ME  
11 and MI-EEG signals. Starting from a frequency value of 4 Hz, frequency bands were selected with 6  
12 Hz width and overlap factor of 3 Hz up to 16 Hz, and frequency bands with 8 Hz width and overlap  
13 factor of 4 Hz for frequencies above 13 Hz (up to 121 Hz and 37 Hz for ME- and MI-EEG signals,  
14 respectively). Thus, 29 and 8 band-pass filters were computed for ME- and MI-EEG signals. For each  
15 of these manually designed filters, EEG signals were band-pass filtered. Two CSP filter pairs (four  
16 filters total) for each filter bank were computed on the training data. Since a few spatial filters  
17 computed often are enough to reach good decoding performance while using all the spatial filters  
18 may lead to overfitting (Blankertz et al., 2008; Chin et al., 2009), we included the feature selection  
19 procedure adopted in (Schirrneister et al., 2017).

20 As the decoding task is multi-class, the problem was transformed into several binary  
21 classification tasks via a one-vs-one reduction (OVO), where binary classifiers learned to  
22 discriminate each pair of classes. Then, a majority weighted voting was applied at prediction time.  
23 To do so, we trained a rLDA classifier with shrinkage regularization (Ledoit & Wolf, 2004), widely  
24 used in EEG decoding (Lotte et al., 2018) for each pair of classes, summed up the classifier outputs  
25 and the class with higher sum was decoded as the predicted one (Chin et al., 2009).

1 Comparing FBCSP+rLDA results – obtained by re-implementing the steps adopted in (Schirrmeister  
2 et al., 2017) – with another study (Sakhavi et al., 2015) that used the same MI dataset, no significant  
3 difference was observed ( $P = 0.441$  Wilcoxon signed-ranked test, average accuracy across subjects:  
4 67.5 vs. 67.0 % (Sakhavi et al., 2015)). This validated the FBCSP+rLDA re-implementation adopted  
5 in this study.

6

1 **Acknowledgments**

2 We gratefully acknowledge the support of NVIDIA Corporation with the donation of the TITAN V  
3 used for this research. This work is part of the “Department of Excellence” Project of the Department  
4 of Electrical, Electronic and Information Engineering, University of Bologna, funded by the Italian  
5 Ministry of Education, Universities and Research (MIUR).

6

7 **Conflicts of interest**

8 The authors declare that there is no conflict of interest regarding the publication of this paper.

9

## 1 **References**

- 2 Ang, K. K., Chin, Z. Y., Wang, C., Guan, C., & Zhang, H. (2012). Filter Bank Common Spatial  
3 Pattern Algorithm on BCI Competition IV Datasets 2a and 2b. *Frontiers in Neuroscience*, 6,  
4 39. <https://doi.org/10.3389/fnins.2012.00039>
- 5 Ang, K. K., Chin, Z. Y., Zhang, H., & Guan, C. (2008). Filter bank common spatial pattern  
6 (FBCSP) in brain-computer interface. *2008 IEEE International Joint Conference on Neural  
7 Networks (IEEE World Congress on Computational Intelligence)*, 2390–2397.
- 8 Ball, T., Demandt, E., Mutschler, I., Neitzel, E., Mehring, C., Vogt, K., Aertsen, A., & Schulze-  
9 Bonhage, A. (2008). Movement related activity in the high gamma range of the human  
10 EEG. *NeuroImage*, 41, 302–310.
- 11 Bashashati, A., Fatourechi, M., Ward, R. K., & Birch, G. E. (2007). A survey of signal processing  
12 algorithms in brain–computer interfaces based on electrical brain signals. *Journal of Neural  
13 Engineering*, 4(2), R32–R57.
- 14 Bashivan, P., Rish, I., Yeasin, M., & Codella, N. (2015). Learning Representations from EEG with  
15 Deep Recurrent-Convolutional Neural Networks. *CoRR*, <abs/1511.06448>.
- 16 Benjamini, Y., & Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and  
17 Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B  
18 (Methodological)*, 57(1), 289–300.
- 19 Blankertz, B., Tomioka, R., Lemm, S., Kawanabe, M., & Muller, K. (2008). Optimizing Spatial  
20 filters for Robust EEG Single-Trial Analysis. *IEEE Signal Processing Magazine*, 25(1), 41–  
21 56. <https://doi.org/10.1109/MSP.2008.4408441>
- 22 Borra, D., Fantozzi, S., & Magosso, E. (2020a). Convolutional Neural Network for a P300 Brain-  
23 Computer Interface to Improve Social Attention in Autistic Spectrum Disorder. In J.  
24 Henriques, N. Neves, & P. de Carvalho (Eds.), *XV Mediterranean Conference on Medical  
25 and Biological Engineering and Computing – MEDICON 2019* (pp. 1837–1843). Springer  
26 International Publishing.

- 1 Borra, D., Fantozzi, S., & Magosso, E. (2020b). EEG Motor Execution Decoding via Interpretable  
2 Sinc-Convolutional Neural Networks. In J. Henriques, N. Neves, & P. de Carvalho (Eds.),  
3 *XV Mediterranean Conference on Medical and Biological Engineering and Computing –*  
4 *MEDICON 2019* (pp. 1113–1122). Springer International Publishing.
- 5 Cecotti, H., & Graser, A. (2011). Convolutional Neural Networks for P300 Detection with  
6 Application to Brain-Computer Interfaces. *IEEE Transactions on Pattern Analysis and*  
7 *Machine Intelligence*, 33(3), 433–445.
- 8 Chin, Z. Y., Ang, K. K., Wang, C., Guan, C., & Zhang, H. (2009). Multi-class filter bank common  
9 spatial pattern for four-class motor imagery BCI. *2009 Annual International Conference of*  
10 *the IEEE Engineering in Medicine and Biology Society*, 571–574.  
11 <https://doi.org/10.1109/IEMBS.2009.5332383>
- 12 Chollet, F. (2016). Xception: Deep Learning with Depthwise Separable Convolutions. *2017 IEEE*  
13 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 1800–1807.
- 14 Clevert, D.-A., Unterthiner, T., & Hochreiter, S. (2015). Fast and accurate deep network learning by  
15 exponential linear units (elus). *ArXiv Preprint*.
- 16 Crone, N. E., Miglioretti, D. L., Gordon, B., & Lesser, R. P. (1998). Functional mapping of human  
17 sensorimotor cortex with electrocorticographic spectral analysis. II. Event-related  
18 synchronization in the gamma band. *Brain : A Journal of Neurology*, 121 ( Pt 12), 2301–  
19 2315.
- 20 Farahat, A., Reichert, C., Sweeney-Reed, C., & Hinrichs, H. (2019). Convolutional neural networks  
21 for decoding of covert attention focus and saliency maps for EEG feature visualization.  
22 *Journal of Neural Engineering*. <http://iopscience.iop.org/10.1088/1741-2552/ab3bb4>
- 23 Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural  
24 networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence*  
25 *and Statistics*, 249–256.

- 1 Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., & Bengio, Y. (2013). Maxout  
2 networks. *Proceedings of the 30th International Conference on International Conference on*  
3 *Machine Learning-Volume 28*, III–1319.
- 4 Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by  
5 Reducing Internal Covariate Shift. In F. Bach & D. Blei (Eds.), *Proceedings of the 32nd*  
6 *International Conference on Machine Learning* (Vol. 37, pp. 448–456). PMLR.
- 7 Jonas, S., Rossetti, A. O., Oddo, M., Jenni, S., Favaro, P., & Zubler, F. (2019). EEG-based outcome  
8 prediction after cardiac arrest with convolutional neural networks: Performance and  
9 visualization of discriminative features. *Human Brain Mapping, 40*(16), 4606–4617.  
10 <https://doi.org/10.1002/hbm.24724>
- 11 Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *ArXiv Preprint*.
- 12 Lawhern, V. J., Solon, A. J., Waytowich, N. R., Gordon, S. M., Hung, C. P., & Lance, B. J. (2018).  
13 EEGNet: A compact convolutional neural network for EEG-based brain–computer  
14 interfaces. *Journal of Neural Engineering, 15*(5), 056013.
- 15 Ledoit, O., & Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance  
16 matrices. *Journal of Multivariate Analysis, 88*(2), 365–411. <https://doi.org/10.1016/S0047->  
17 [259X\(03\)00096-4](https://doi.org/10.1016/S0047-259X(03)00096-4)
- 18 Leeuwen, K. G. van, Sun, H., Tabaeizadeh, M., Struck, A. F., Putten, M. J. A. M. van, & Westover,  
19 M. B. (2019). Detecting abnormal electroencephalograms using deep convolutional  
20 networks. *Clinical Neurophysiology, 130*(1), 77–84.
- 21 Liu, M., Wu, W., Gu, Z., Yu, Z., Qi, F., & Li, Y. (2018). Deep learning based on Batch  
22 Normalization for P300 signal detection. *Neurocomputing, 275*, 288–297.  
23 <https://doi.org/10.1016/j.neucom.2017.08.039>
- 24 Lotte, F. (2015). Signal Processing Approaches to Minimize or Suppress Calibration Time in  
25 Oscillatory Activity-Based Brain–Computer Interfaces. *Proceedings of the IEEE, 103*(6),  
26 871–890. <https://doi.org/10.1109/JPROC.2015.2404941>

- 1 Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., & Yger, F.  
2 (2018). A review of classification algorithms for EEG-based brain–computer interfaces: A  
3 10 year update. *Journal of Neural Engineering*, *15*(3), 031005.
- 4 Mak, J. N., & Wolpaw, J. R. (2009). Clinical Applications of Brain-Computer Interfaces: Current  
5 State and Future Prospects. *IEEE Reviews in Biomedical Engineering*, *2*, 187–199.  
6 <https://doi.org/10.1109/RBME.2009.2035356>
- 7 Manor, R., & Geva, A. B. (2015). Convolutional Neural Network for Multi-Category Rapid Serial  
8 Visual Presentation BCI. *Frontiers in Computational Neuroscience*, *9*, 146.  
9 <https://doi.org/10.3389/fncom.2015.00146>
- 10 McFarland, D. J., Anderson, C. W., Muller, K.-, Schlogl, A., & Krusienski, D. J. (2006). BCI  
11 meeting 2005-workshop on BCI signal processing: Feature extraction and translation. *IEEE*  
12 *Transactions on Neural Systems and Rehabilitation Engineering*, *14*(2), 135–138.
- 13 Mirnaziri, M., Rahimi, M., Alavikakhaki, S., & Ebrahimpour, R. (2013). Using Combination of  $\mu$ ,  $\beta$   
14 and  $\gamma$  Bands in Classification of EEG Signals. *Basic and Clinical Neuroscience*.
- 15 Montavon, G., Samek, W., & Müller, K.-R. (2018). Methods for interpreting and understanding  
16 deep neural networks. *Digital Signal Processing*, *73*, 1–15.  
17 <https://doi.org/10.1016/j.dsp.2017.10.011>
- 18 Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A.,  
19 Antiga, L., & Lerer, A. (2017). Automatic differentiation in PyTorch. *NIPS-W*.
- 20 Pfurtscheller, G. (1981). Central beta rhythm during sensorimotor activities in man.  
21 *Electroencephalography and Clinical Neurophysiology*, *51*(3), 253–264.  
22 [https://doi.org/10.1016/0013-4694\(81\)90139-5](https://doi.org/10.1016/0013-4694(81)90139-5)
- 23 Pfurtscheller, G., & Aranibar, A. (1977). Event-related cortical desynchronization detected by  
24 power measurements of scalp EEG. *Electroencephalography and Clinical Neurophysiology*,  
25 *42*(6), 817–826. [https://doi.org/10.1016/0013-4694\(77\)90235-8](https://doi.org/10.1016/0013-4694(77)90235-8)

- 1 Pfurtscheller, G., & Berghold, A. (1989). Patterns of cortical activation during planning of  
2 voluntary movement. *Electroencephalography and Clinical Neurophysiology*, 72(3), 250–  
3 258. [https://doi.org/10.1016/0013-4694\(89\)90250-2](https://doi.org/10.1016/0013-4694(89)90250-2)
- 4 Pfurtscheller, G., Brunner, C., Schlögl, A., & Silva, F. H. L. da. (2006). Mu rhythm  
5 (de)synchronization and EEG single-trial classification of different motor imagery tasks.  
6 *NeuroImage*, 31(1), 153–159. <https://doi.org/10.1016/j.neuroimage.2005.12.003>
- 7 Pfurtscheller, G., & Silva, F. H. L. da. (1999). Event-related EEG/MEG synchronization and  
8 desynchronization: Basic principles. *Clinical Neurophysiology*, 110(11), 1842–1857.  
9 [https://doi.org/10.1016/S1388-2457\(99\)00141-8](https://doi.org/10.1016/S1388-2457(99)00141-8)
- 10 Pfurtscheller, Gert, Flotzinger, D., & Neuper, C. (1994). Differentiation between finger, toe and  
11 tongue movement in man based on 40 Hz EEG. *Electroencephalography and Clinical*  
12 *Neurophysiology*, 90(6), 456–460. [https://doi.org/10.1016/0013-4694\(94\)90137-6](https://doi.org/10.1016/0013-4694(94)90137-6)
- 13 Ravanelli, M., & Bengio, Y. (2018). Speaker Recognition from Raw Waveform with SincNet. 2018  
14 *IEEE Spoken Language Technology Workshop (SLT)*, 1021–1028.  
15 <https://doi.org/10.1109/SLT.2018.8639585>
- 16 Roy, Y., Banville, H., Albuquerque, I., Gramfort, A., Falk, T. H., & Faubert, J. (2019). Deep  
17 learning-based electroencephalography analysis: A systematic review. *Journal of Neural*  
18 *Engineering*, 16(5), 051001. <https://doi.org/10.1088/1741-2552/ab260c>
- 19 Sakhavi, S., Guan, C., & Yan, S. (2015). Parallel convolutional-linear neural network for motor  
20 imagery classification. 2015 23rd European Signal Processing Conference (EUSIPCO),  
21 2736–2740. <https://doi.org/10.1109/EUSIPCO.2015.7362882>
- 22 Schirrmester, R. T., Springenberg, J. T., Fiederer, L. D. J., Glasstetter, M., Eggenberger, K.,  
23 Tangermann, M., Hutter, F., Burgard, W., & Ball, T. (2017). Deep learning with  
24 convolutional neural networks for EEG decoding and visualization. *Human Brain Mapping*,  
25 38(11), 5391–5420.

- 1 Shamwell, J., Lee, H., Kwon, H., Marathe, A. R., Lawhern, V., & Nothwang, W. (2016). Single-  
2 trial EEG RSVP classification using convolutional neural networks. In T. George, A. K.  
3 Dutta, & M. S. Islam (Eds.), *Micro- and Nanotechnology Sensors, Systems, and*  
4 *Applications VIII* (Vol. 9836, pp. 373 – 382). SPIE. <https://doi.org/10.1117/12.2224172>
- 5 Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep inside convolutional networks:  
6 Visualising image classification models and saliency maps. *ArXiv Preprint*.
- 7 Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A  
8 simple way to prevent neural networks from overfitting. *The Journal of Machine Learning*  
9 *Research, 15*(1), 1929–1958.
- 10 Tabar, Y. R., & Halici, U. (2016). A novel deep learning approach for classification of EEG motor  
11 imagery signals. *Journal of Neural Engineering, 14*(1), 016003.
- 12 Tang, Z., Li, C., & Sun, S. (2017). Single-trial EEG classification of motor imagery using deep  
13 convolutional neural networks. *Optik, 130*, 11–18.  
14 <https://doi.org/10.1016/j.ijleo.2016.10.117>
- 15 Tangermann, M., Müller, K.-R., Aertsen, A., Birbaumer, N., Braun, C., Brunner, C., Leeb, R.,  
16 Mehring, C., Miller, K. J., Mueller-Putz, G., & others. (2012). Review of the BCI  
17 competition IV. *Frontiers in Neuroscience, 6*, 55.
- 18 Zeng, H., Wu, Z., Zhang, J., Yang, C., Zhang, H., Dai, G., & Kong, W. (2019). EEG Emotion  
19 Classification Using an Improved SincNet-Based Deep Learning Model. *Brain Sciences,*  
20 *9*(11). <https://doi.org/10.3390/brainsci9110326>
- 21 Zhao, D., Tang, F., Si, B., & Feng, X. (2019). Learning joint space–time–frequency features for  
22 EEG decoding on small labeled data. *Neural Networks, 114*, 67–77.

23  
24

## 1 LEGENDS TO FIGURES

2 **Figure 1** – Electrode locations for the two examined datasets. (a) ME-EEG dataset. (b) MI-EEG  
3 dataset.

4 **Figure 2** – Architecture of Sinc-ShallowNet. For simplicity, the figure shows only the more  
5 significant layers within each of the three blocks (see also [Sections 2.3.2, 2.3.3, 2.3.4](#) and [Table 1](#)).

6 **Figure 3** – Confusion matrices of FBCSP+rLDA ((a) and (c)) and of Sinc-ShallowNet ((b) and (d)).  
7 Sinc-ShallowNet was trained with trialwise strategy (see [Section 2.4.1](#)). [Matrices \(a\) and \(b\)](#) were  
8 computed across subject-specific classifiers on ME-EEG signals belonging to the test set, [while \(c\)](#)  
9 [and \(d\) were computed on MI-EEG signals](#) belonging to the test set. Each cell contains the [total](#)  
10 [number of trials across subjects given a specific prediction and target label, and the ratio between this](#)  
11 [number and the total number of trials for each target label. For each \(i,j\) location \(16 in total\) of the](#)  
12 [confusion matrix \(predicted class i, true class j\), a Wilcoxon signed-rank test was performed between](#)  
13 [the entries of the subject-specific confusion matrices obtained with FBCSP+rLDA and with Sinc-](#)  
14 [ShallowNet, separately for the two datasets. Correction for multiple comparisons was obtained via](#)  
15 [the Benjamini-Hochberg procedure. The corrected p-value resulting from each comparison is](#)  
16 [displayed inside the corresponding cell of the matrices reporting Sinc-ShallowNet results.](#)

17 **Figure 4** – Results of the analyses on Sinc-ShallowNet design choices and on training strategies. (a)  
18 Effect of the changes in the hyper-parameters of Sinc-ShallowNet (see [Table 2](#)) on the performance  
19 metric. The changes in accuracy ( $\Delta_{acc}$ ) were computed as the difference between the tested and the  
20 reference (i.e. basal) configuration ( $\Delta_{acc} = acc_{tested} - acc_{ref}$ , e.g.  $acc_{K_1=8} - acc_{K_1=32}$ ). (b) Effect  
21 of the two different training strategies applied to each SOA CNN and to Sinc-ShallowNet on the  
22 performance metric. The changes in accuracy ( $\Delta_{acc}$ ) were computed as the difference between the  
23 cropped and trialwise training strategies ( $\Delta_{acc} = acc_{cropped} - acc_{trialwise}$ ). For this comparison, MI-  
24 EEG signals were epoched between 0.5 and 4 s (see [Section 2.4.2](#)). In both panels,  $\Delta_{acc}$  obtained with

1 ME-EEG signals (◦) and with MI-EEG signals (+) were grouped together. The corrected P values are  
2 reported (Sinc-ShallowNet vs. each variant, trialwise vs. cropped training).

3 **Figure 5** – Visualization and interpretation of the features learned by the temporal sinc-convolutional  
4 layer of Sinc-ShallowNet in case of ME-EEG signals of subject 12 (decoding accuracy 95.6%). (a)  
5 Visualization of the passband learned by each of the 32 filters. Each passband is displayed as a black  
6 line, with the end points representing  $f_{1,j}$  and  $f_{2,j}$  of the  $j$ -th learned filter. The colour-code used is:  
7 gray- $\theta$ , green- $\alpha$ , yellow- $\beta$ , red-low  $\gamma$ , blue-high  $\gamma$ . (b) Results of the temporal sensitivity analysis at  
8 the level of EEG bands: the normalized gradient averaged across the band-pass filters belonging to a  
9 specific EEG band ( $\hat{g}_{b,k}$ ) is displayed (colour-coded) for each class and each EEG band.

10 **Figure 6** – Visualization and interpretation of the features learned by the temporal sinc-convolutional  
11 layer of Sinc-ShallowNet in case of MI-EEG signals of subject 3 (decoding accuracy 86.1%). (a)  
12 Visualization of the passband learned by each of the 32 filters. Each passband is displayed as a black  
13 line, with the end points representing  $f_{1,j}$  and  $f_{2,j}$  of the  $j$ -th learned filter. The colour-code used is:  
14 gray- $\theta$ , green- $\alpha$ , yellow- $\beta$ , red-low  $\gamma$ . (b) Results of the temporal sensitivity analysis at the level of  
15 EEG bands: the normalized gradient averaged across the band-pass filters belonging to a specific  
16 EEG band ( $\hat{g}_{b,k}$ ) is displayed (colour-coded) for each class and each EEG band.

17 **Figure 7** – Spatial distribution of the more relevant and more class-specific band-pass filters learned  
18 by Sinc-ShallowNet in case of ME-EEG signals of subject 12 (the same as in Figure 5). Each panel  
19 refers to a specific class (a-d for “Right Hand”, “Left Hand”, “Rest”, and “Feet”, respectively) and  
20 shows the results of the temporal sensitivity analysis at the level of each single band-pass filter by  
21 displaying both the normalized gradient ( $\hat{g}_{j,k}$ ) and rescaled ( $\hat{g}'_{j,k}$ ) gradient of the single filters for that  
22 specific class. The coloured bars denote the rescaled gradients (the colour indicates the EEG band the  
23 filter belongs to, i.e. gray- $\theta$ , green- $\alpha$ , yellow- $\beta$ , red-low  $\gamma$ , blue-high  $\gamma$ ), while the black lines denote  
24 the normalized gradients. The latter are reported in order to identify an increase in the rescaled  
25 gradients. For each class, the two more important band-pass filters within each of the two more

1 important EEG bands (according to Figure 5b) are selected depending on the value of the increased  
2 rescaled gradients. For the so-selected band-pass filters, the spatial distribution is displayed by  
3 drawing the absolute values of the corresponding two spatial filters. In case of the “Right Hand” class,  
4 only one band-pass filter (#26) within the high  $\gamma$  band was selected for this visualization since it was  
5 the only one having  $\hat{g}'_{j,k} > \hat{g}_{j,k}$ .

6 **Figure 8** – Spatial distribution of the more relevant and more class-specific band-pass filters learned  
7 by Sinc-ShallowNet in case of MI-EEG signals of subject 3 (the same as in Figure 6). Each panel  
8 refers to a specific class (a-d for “Left Hand”, “Right Hand”, “Feet”, and “Tongue”, respectively) and  
9 shows the results of the temporal sensitivity analysis at the level of each single band-pass filter by  
10 displaying both the normalized gradient ( $\hat{g}_{j,k}$ ) and rescaled gradient ( $\hat{g}'_{j,k}$ ) of the single filters for that  
11 specific class. The coloured bars denote the rescaled gradients (the colour indicates the EEG band the  
12 filter belongs to, i.e., gray- $\theta$ , green- $\alpha$ , yellow- $\beta$ , red-low  $\gamma$ ), while the black lines denote the  
13 normalized gradients. The latter are reported in order to identify an increase in the rescaled gradients.  
14 For each class, the two more important band-pass filters within each of the two more important EEG  
15 bands (according to Figure 6b) are selected depending on the value of the increased rescaled  
16 gradients. For the so-selected band-pass filters, the spatial distribution is displayed by drawing the  
17 absolute values of the corresponding two spatial filters. In case of the “Right Hand” class, the band-  
18 pass filters within the  $\alpha$  band (#1 and #7) were not selected for the visualization since  $\hat{g}'_{j,k} < \hat{g}_{j,k}$  for  
19 these filters.

20

1 **Table 1** – Architecture details of Sinc-ShallowNet. The architecture corresponding the hyper-  
2 parameters reported here is denoted as “basal” Sinc-ShallowNet (variants of this basal architecture  
3 are also tested, see Table 2). Each layer is provided with its name, main hyper-parameters, output  
4 shape and number of trainable parameters and adopted activation function.  $C$  and  $T$  represent the  
5 number of electrodes and time samples of the network input, respectively.  $N_c$  is the number of the  
6 classes. See Section 2.3 for the meaning of the other symbols. The output shapes of the layers within  
7 the first and second blocks are described by tuples of three integers (in brackets) denoting the number  
8 of feature maps (CNN channel dimension) and the number of spatial and temporal samples within  
9 each map, respectively. The input layer provides an output of shape  $(1, C, T)$  since it is assumed to  
10 just replicate the original input matrix with shape  $(C, T)$ , providing a single feature map as output  
11 (coincident with its input). The output shapes in the third block are 1D, thus described by a single  
12 number. \*Kernel maximum norm constraint was used, enforcing an absolute upper bound on the  
13 magnitude of the weights.

Block	Layer name	Hyper-parameters	Output shape	Number of parameters	Activation
1	Input		$(1, C, T)$	0	
	Sinc-Conv2D	$K_1 = 32$ $F_1 = (1, 65)$ $S_1 = (1, 1)$ $P_1 = (0, 0)$	$(K_1, C, T_1)$	$2 \cdot K_1$	Linear
	BatchNorm2D	$m = 0.99$	$(K_1, C, T_1)$	$2 \cdot K_1$	
	DW-Conv2D*	$K_2 = K_1 \cdot D_2$ $F_2 = (C, 1)$ $D_2 = 2$ $S_2 = (1, 1)$ $P_2 = (0, 0)$	$(K_2, 1, T_1)$	$F_2[0] \cdot K_2$	Linear
2	BatchNorm2D	$m = 0.99$	$(K_2, 1, T_1)$	$2 \cdot K_2$	
	Activation	$\alpha = 1$	$(K_2, 1, T_1)$	0	ELU
	AvgPool2D	$F_p = (1, 109)$ $S_p = (1, 23)$	$(K_2, 1, T_p)$	0	
	Dropout	$p = 0.5$	$(K_2, 1, T_p)$	0	
3	Flatten		$(K_2 \cdot T_p)$	0	
	Fully-Connected*	$N_c = 4$	$(N_c)$	$N_c \cdot T_p \cdot K_2 + N_c$	
	Activation		$(N_c)$	0	Softmax

14

1 **Table 2** – Investigated design choices.

Architectural aspect	Basal	Variants	Motivation
Number of temporal filters $K_1$ of block 1	$K_1 = 32$	$K_1 = 8$ $K_1 = 16$	We wanted to test if lowering the number of the temporal kernels worsened the performance, i.e. to check if all the 32 temporal filters were needed or some of them <a href="#">were</a> redundant. Furthermore, since there is a consistent variability in the number of temporal kernels within CNNs for EEG decoding (e.g. 8 in EEGNet, 25 in DeepConvNet, 40 in ShallowConvNet), this test on Sinc-ShallowNet may gain insights about the effect of this hyper-parameter on the decoding performance. Of course, a larger number of the band-pass filters implied a larger number of trainable parameters but this effect was limited since the sinc-convolutional layer learns only 2 parameters for each temporal filter.
Number of spatial filters per temporal filter $D_2$ of block 1	$D_2 = 2$	$D_2 = 4$	We wanted to test if increasing the number of the spatial filters for each band-pass filter increased the performance. We expected that a higher $D_2$ was more beneficial for those applications in which the band-pass kernels were more dispersed across a large frequency range, i.e. in case the signals contained more frequency components, such as the investigated ME-EEG signals. In case of less dispersed band-pass filters, there is high probability that neighbor band-pass kernels are learned; the neighbor band-pass kernels can compensate for the reduction in $D_2$ as <a href="#">they may be tied with different</a> spatial filters <a href="#">learned during training</a> , actually providing an augmented set of spatial filters for a given band-pass filtering. The drawback of <a href="#">an increase of <math>D_2</math></a> was an increased number of trainable parameters.
Pooling size $F_p$ and stride $S_p$ of block 2	$F_p = (1,109)$ $S_p = (1,23)$	$F_p = (1,71)$ $S_p = (1,15)$	We wanted to evaluate the impact of a shorter average pooling on the performance. The modified values of these hyper-parameters <a href="#">corresponded</a> to the extraction of averaged spatial activations of 325 ms with a stride of 70 ms (similarly as done in (Schirrneister et al., 2017; Zhao et al., 2019)). This variant resulted in an increased number of trainable parameters due to a convolutional-to-dense transition involving more units.
Recombination of the spatial activations via an additional pointwise convolution in block 2	No recomb.	Recomb.	We wanted to evaluate the impact of the recombination of the spatial activations on the performance. A pointwise convolutional layer was introduced immediately after the spatial depthwise convolutional layer, in order to recombine the learned spatial activations across the feature map dimension. The hyper-parameters of this layer were $K_3 = K_2 = K_1 \cdot D_2$ , $F_3 = (1,1)$ , $S_3 = (1,1)$ , $P_3 = (0,0)$ . The combination of a depthwise and a pointwise convolution is called separable convolution (Chollet, 2016). The introduction of pointwise convolution increase the number of trainable parameters by $(K_2)^2$ and the resulting architecture may need <a href="#">a</a> large training set. Thus, this modification could be more beneficial in case of the investigated ME-EEG signals.

2

1 **Table 3** – Accuracies (mean  $\pm$  std across subjects) of the basal Sinc-ShallowNet and SOA algorithms,  
 2 obtained with ME-EEG signals belonging to the test set. Here, the trialwise training was adopted. For  
 3 each CNN, the total number of trainable parameters is reported in brackets. The corrected P values  
 4 are reported ( $P_1$  for each CNN vs. FBCSP+rLDA,  $P_2$  for Sinc-ShallowNet vs. each SOA CNN).

Algorithm	Accuracy (%)	$P_1$	$P_2$
FBCSP+rLDA	86.0 $\pm$ 9.0		
EEGNet (2604)	88.5 $\pm$ 11.0	0.158	0.158
DeepConvNet (298229)	88.4 $\pm$ 8.8	0.158	0.026
ShallowConvNet (82564)	93.9 $\pm$ 9.3	0.024	0.040
Sinc-ShallowNet (13828)	91.2 $\pm$ 9.1	0.024	

5

1 **Table 4** – Accuracies (mean  $\pm$  std across subjects) of the basal Sinc-ShallowNet and SOA algorithms,  
 2 obtained with MI-EEG signals belonging to the test set. Here, the trialwise training was adopted  
 3 (signal epoching 0.5-2.5 s). For each CNN, the total number of trainable parameters is reported in  
 4 brackets. The corrected P values are reported ( $P_1$  for each CNN vs. FBCSP+rLDA,  $P_2$  for Sinc-  
 5 ShallowNet vs. each SOA CNN).

Algorithm	Accuracy (%)	$P_1$	$P_2$
FBCSP+rLDA	67.5 $\pm$ 13.9		
EEGNet (1932)	66.0 $\pm$ 13.1	0.575	0.027
DeepConvNet (278079)	50.5 $\pm$ 19.6	0.031	0.027
ShallowConvNet (40644)	71.6 $\pm$ 14.2	0.046	0.302
Sinc-ShallowNet (5508)	72.8 $\pm$ 12.9	0.031	

6

1 **Table A.1** – Architecture details of EEGNet. Each layer is provided with its name, main hyper-  
2 parameters, number of trainable parameters and activation function. See Section 2.3 for the meaning  
3 of the symbols. \*Kernel maximum norm constraint at 1 and 0.25, respectively for the depthwise  
4 convolutional and fully-connected layers.

Layer name	Hyper-parameters	Number of parameters	Activation
Input		0	
Conv2D	$K_1 = 8$ $F_1 = (1,65)$ $S_1 = (1,1)$ $P_1 = (0,32)$	$F_1[1] \cdot K_1$	Linear
BatchNorm2D	$m = 0.99$	$2 \cdot K_1$	
DW-Conv2D*	$K_2 = K_1 \cdot D_2$ $F_2 = (C, 1)$ $D_2 = 2$ $S_2 = (1,1)$ $P_2 = (0,0)$	$F_2[0] \cdot K_2$	Linear
BatchNorm2D	$m = 0.99$	$2 \cdot K_2$	
Activation	$\alpha = 1$	0	ELU
AvgPool2D	$F_{p1} = (1,8)$ $S_{p1} = (1,8)$	0	
Dropout	$p = 0.5$	0	
Sep-Conv2D	$K_3 = K_2 \cdot D_3$ $F_3 = (1,33)$ $D_3 = 1$ $S_3 = (1,1)$ $P_3 = (0,16)$	$F_3[1] \cdot K_3 + (K_3)^2$	Linear
BatchNorm2D	$m = 0.99$	$2 \cdot K_3$	
Activation	$\alpha = 1$	0	ELU
AvgPool2D	$F_{p2} = (1,16)$ $S_{p2} = (1,16)$	0	
Dropout	$p = 0.5$	0	
Flatten		0	
Fully-Connected*	$N_c = 4$	$N_c \cdot T_{p2} \cdot K_3 + N_c$	
Activation		0	Softmax

5

6

1 **Table A.2** – Architecture details of DeepConvNet. Each layer is provided with its name, main hyper-  
2 parameters, number of trainable parameters and activation function. See Section 2.3 for the meaning  
3 of the symbols. \*Kernel maximum norm constraint at 2 and 0.5, respectively for the convolutional  
4 and fully-connected layers. For numerical stability, batch normalization  $\varepsilon$  parameter was set to 1e-5.

Layer name	Hyper-parameters	Number of parameters	Activation
Input		0	
Conv2D*	$K_1 = 25$ $F_1 = (1,10)$ $S_1 = (1,1)$ $P_1 = (0,0)$	$F_1[1] \cdot K_1 + K_1$	Linear
Conv2D*	$K_2 = 25$ $F_2 = (C, 1)$ $S_2 = (1,1)$ $P_2 = (0,0)$	$K_1 \cdot F_2[0] \cdot K_2$	Linear
BatchNorm2D	$m = 0.9$	$2 \cdot K_2$	
Activation	$\alpha = 1$	0	ELU
MaxPool2D	$F_{p1} = (1,2)$ $S_{p1} = (1,2)$	0	
Dropout	$p = 0.5$	0	
Conv2D*	$K_3 = 50$ $F_3 = (1,10)$ $S_3 = (1,1)$ $P_3 = (0,0)$	$K_2 \cdot F_3[1] \cdot K_3$	Linear
BatchNorm2D	$m = 0.9$	$2 \cdot K_3$	
Activation	$\alpha = 1$	0	ELU
MaxPool2D	$F_{p2} = (1,2)$ $S_{p2} = (1,2)$	0	
Dropout	$p = 0.5$	0	
Conv2D*	$K_4 = 100$ $F_4 = (1,10)$ $S_4 = (1,1)$ $P_4 = (0,0)$	$K_3 \cdot F_4[1] \cdot K_4$	Linear
BatchNorm2D	$m = 0.9$	$2 \cdot K_4$	
Activation	$\alpha = 1$	0	ELU
MaxPool2D	$F_{p3} = (1,2)$ $S_{p3} = (1,2)$	0	
Dropout	$p = 0.5$	0	
Conv2D*	$K_5 = 200$ $F_5 = (1,10)$ $S_5 = (1,1)$ $P_5 = (0,0)$	$K_4 \cdot F_5[1] \cdot K_5$	Linear
BatchNorm2D	$m = 0.9$	$2 \cdot K_5$	
Activation	$\alpha = 1$	0	ELU
MaxPool2D	$F_{p4} = (1,2)$ $S_{p4} = (1,2)$	0	
Flatten		0	
Fully-Connected*	$N_c = 4$	$N_c \cdot T_{p4} \cdot K_5 + N_c$	
Activation		0	Softmax

1 **Table A.3** – Architecture details of ShallowNet. Each layer is provided with its name, main hyper-  
 2 parameters, number of trainable parameters and activation function. See Section 2.3 for the meaning  
 3 of the symbols. \*Kernel maximum norm constraint at 2 and 0.5, respectively for the convolutional  
 4 and fully-connected layers. For numerical stability, batch normalization  $\varepsilon$  parameter was set to  $1e-5$ ,  
 5 while the log function input was clipped at  $\varepsilon = 1e - 6$ .

Layer name	Hyper-parameters	Number of parameters	Activation
Input		0	
Conv2D*	$K_1 = 40$ $F_1 = (1,25)$ $S_1 = (1,1)$ $P_1 = (0,0)$	$F_1[1] \cdot K_1 + K_1$	Linear
Conv2D*	$K_2 = 40$ $F_2 = (C, 1)$ $S_2 = (1,1)$ $P_2 = (0,0)$	$K_1 \cdot F_2[0] \cdot K_2$	Linear
BatchNorm2D	$m = 0.9$	$2 \cdot K_2$	
Activation	$\alpha = 1$	0	Square
AvgPool2D	$F_p = (1,75)$ $S_p = (1,15)$	0	
Activation		0	Log
Dropout	$p = 0.5$	0	
Flatten		0	
Fully-Connected*	$N_c = 4$	$N_c \cdot T_p \cdot K_2 + N_c$	
Activation		0	Softmax

6

Figure 1

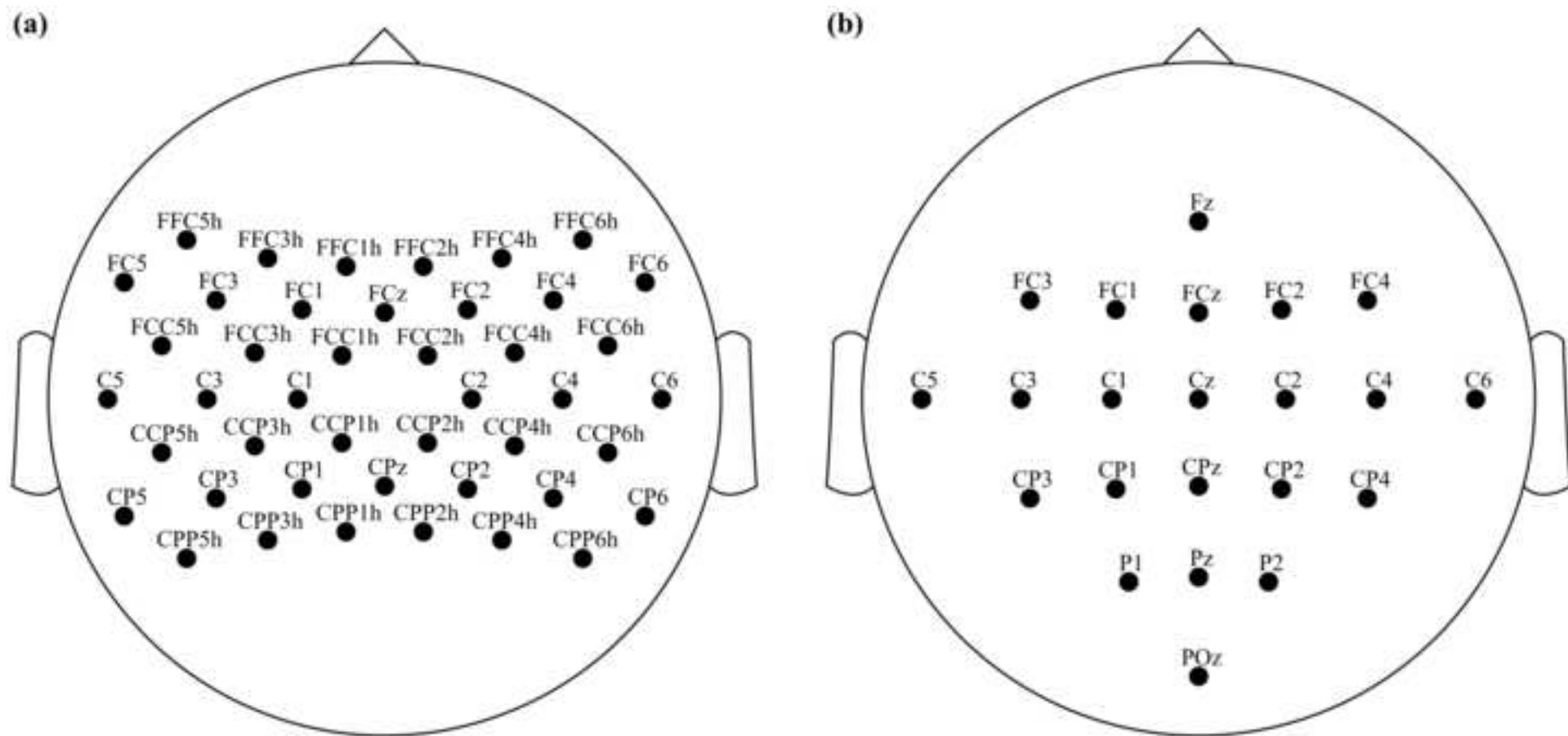


Figure 2

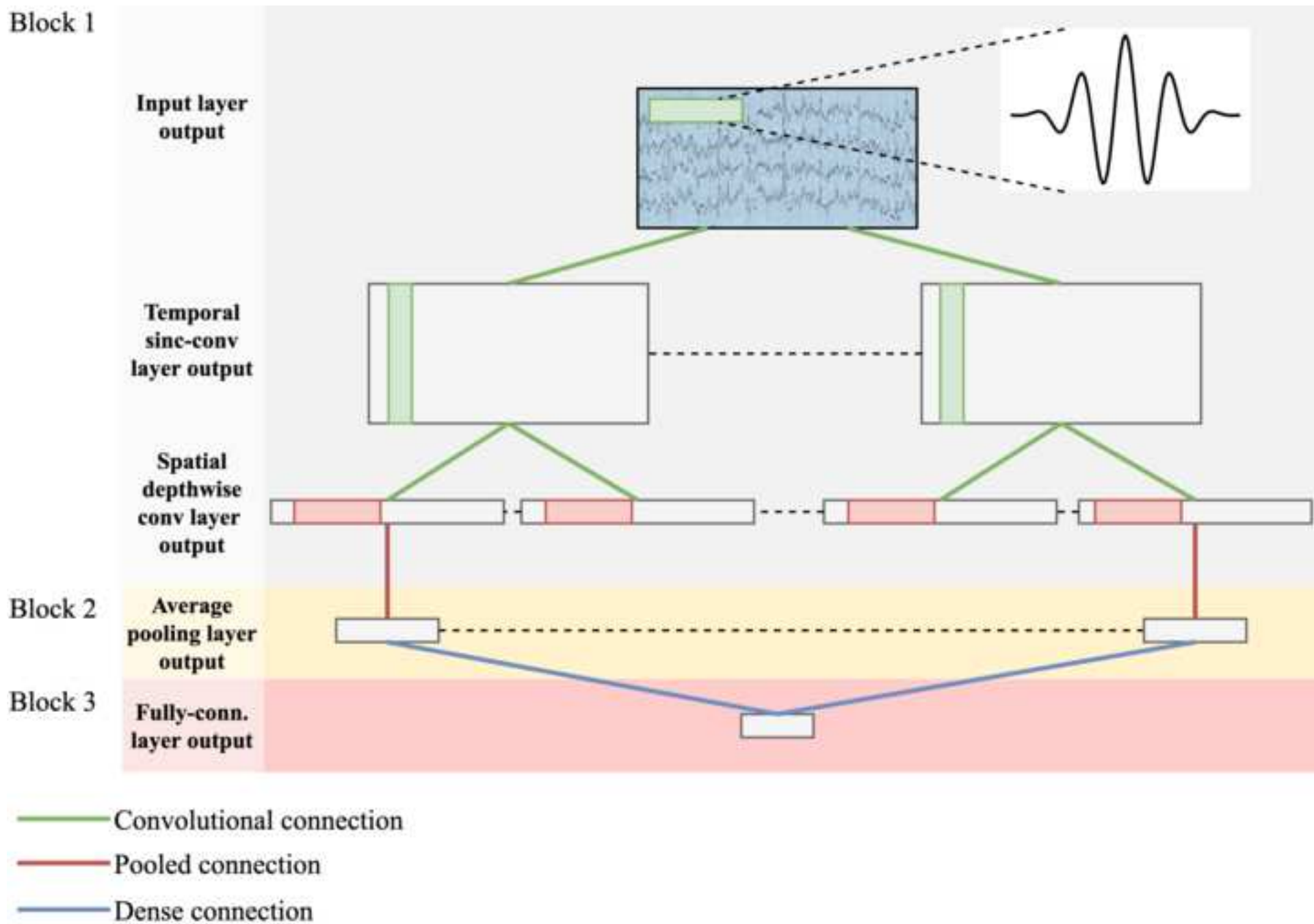


Figure 3

(a) FBCSP+rLDA with ME signals

Predictions	Right Hand	500 89.3%	69 12.3%	34 6.1%	25 4.5%
	Left Hand	41 7.3%	477 85.2%	27 4.8%	16 2.9%
	Rest	19 3.4%	12 2.1%	480 85.7%	49 8.8%
	Foot	0 0.0%	2 0.4%	19 3.4%	470 83.9%
		Right Hand	Left Hand	Rest	Foot
		Targets			

(b) Sinc-ShallowNet with ME signals

Predictions	Right Hand	492 88.0% P=0.722	24 4.3% P=0.099	4 0.7% P=0.036	4 0.7% P=0.117
	Left Hand	66 11.8% P=0.219	534 95.5% P=0.036	28 5.0% P=0.916	20 3.6% P=0.722
	Rest	0 0.0% P=0.042	1 0.2% P=0.117	496 88.6% P=0.682	17 3.0% P=0.117
	Foot	1 0.2% P=0.423	0 0.0% P=0.423	32 5.7% P=0.215	519 92.7% P=0.036
		Right Hand	Left Hand	Rest	Foot
		Targets			

(c) FBCSP+rLDA with MI signals

Predictions	Left Hand	482 74.4%	99 15.3%	87 13.4%	109 16.8%
	Right Hand	97 15.0%	482 74.4%	57 8.8%	76 11.7%
	Foot	53 8.2%	46 7.1%	423 65.3%	100 15.4%
	Tongue	16 2.5%	21 3.2%	81 12.5%	363 56.0%
		Left Hand	Right Hand	Foot	Tongue
		Targets			

(d) Sinc-ShallowNet with MI signals

Predictions	Left Hand	478 73.8% P=1.000	90 13.9% P=0.924	50 7.7% P=1.000	58 9.0% P=0.552
	Right Hand	89 13.7% P=1.000	478 73.8% P=1.000	50 7.7% P=1.000	48 7.4% P=0.924
	Foot	39 6.0% P=0.924	32 4.9% P=0.924	446 68.8% P=0.924	58 9.0% P=0.275
	Tongue	42 6.5% P=0.842	48 7.4% P=0.275	102 15.7% P=0.924	484 74.7% P=0.203
		Left Hand	Right Hand	Foot	Tongue
		Targets			

Figure 4

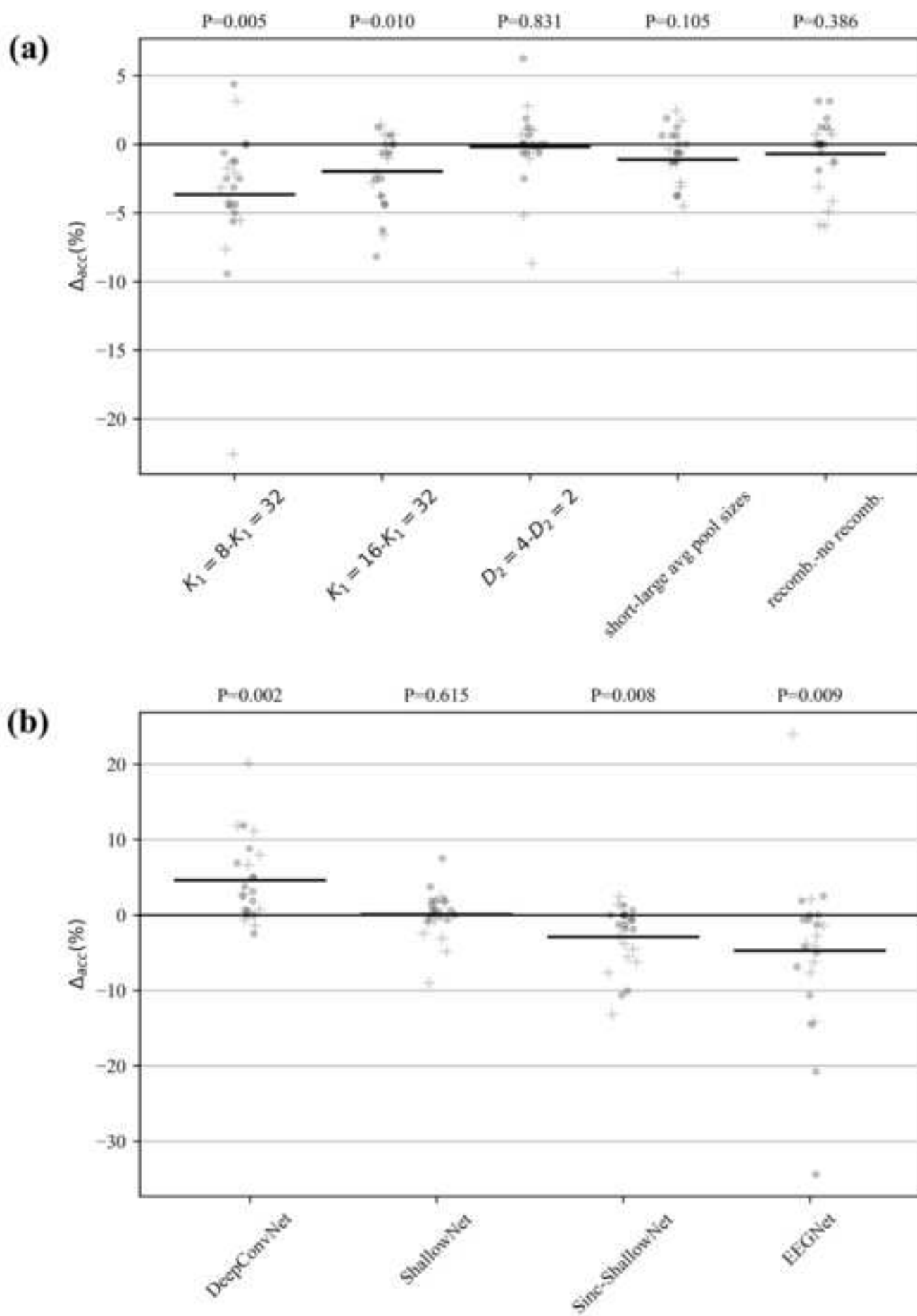


Figure 5

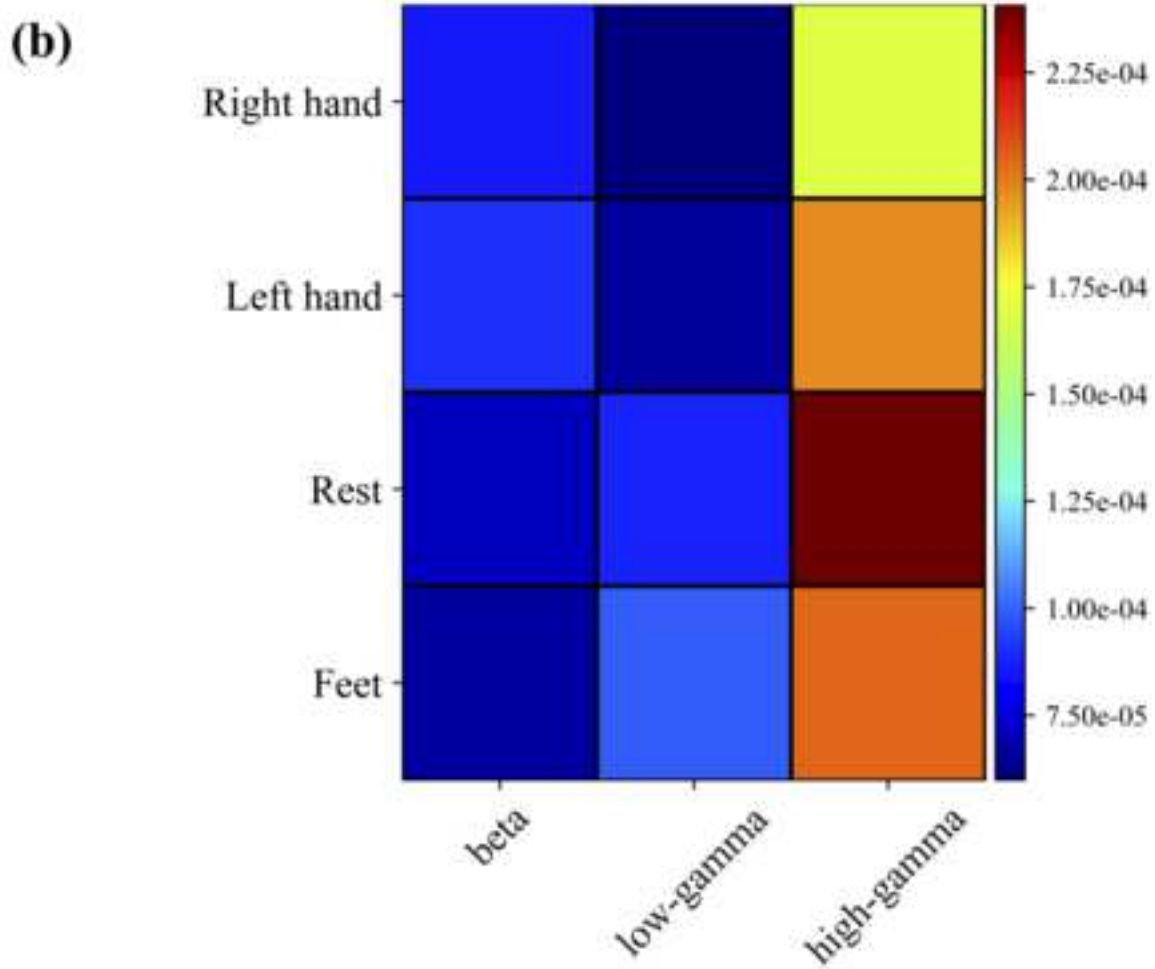
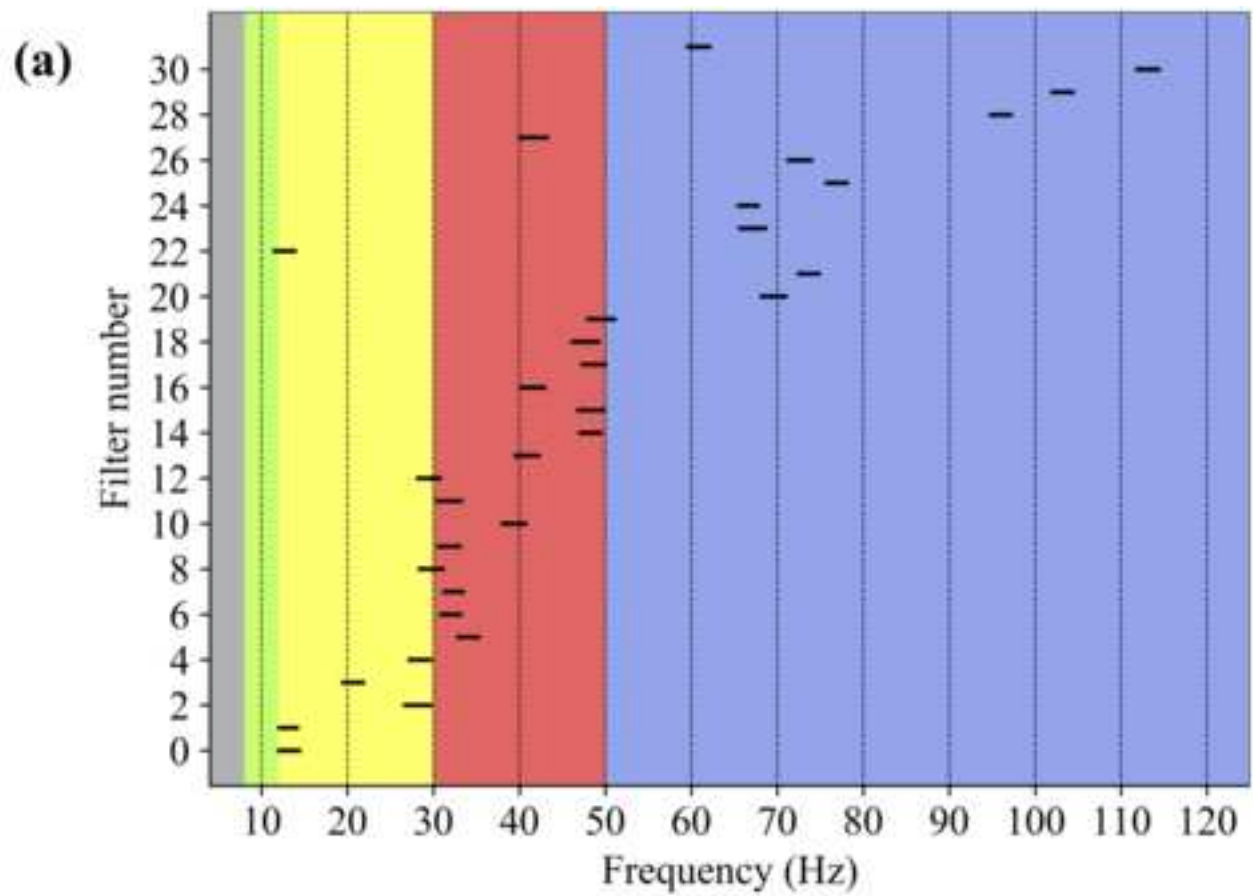
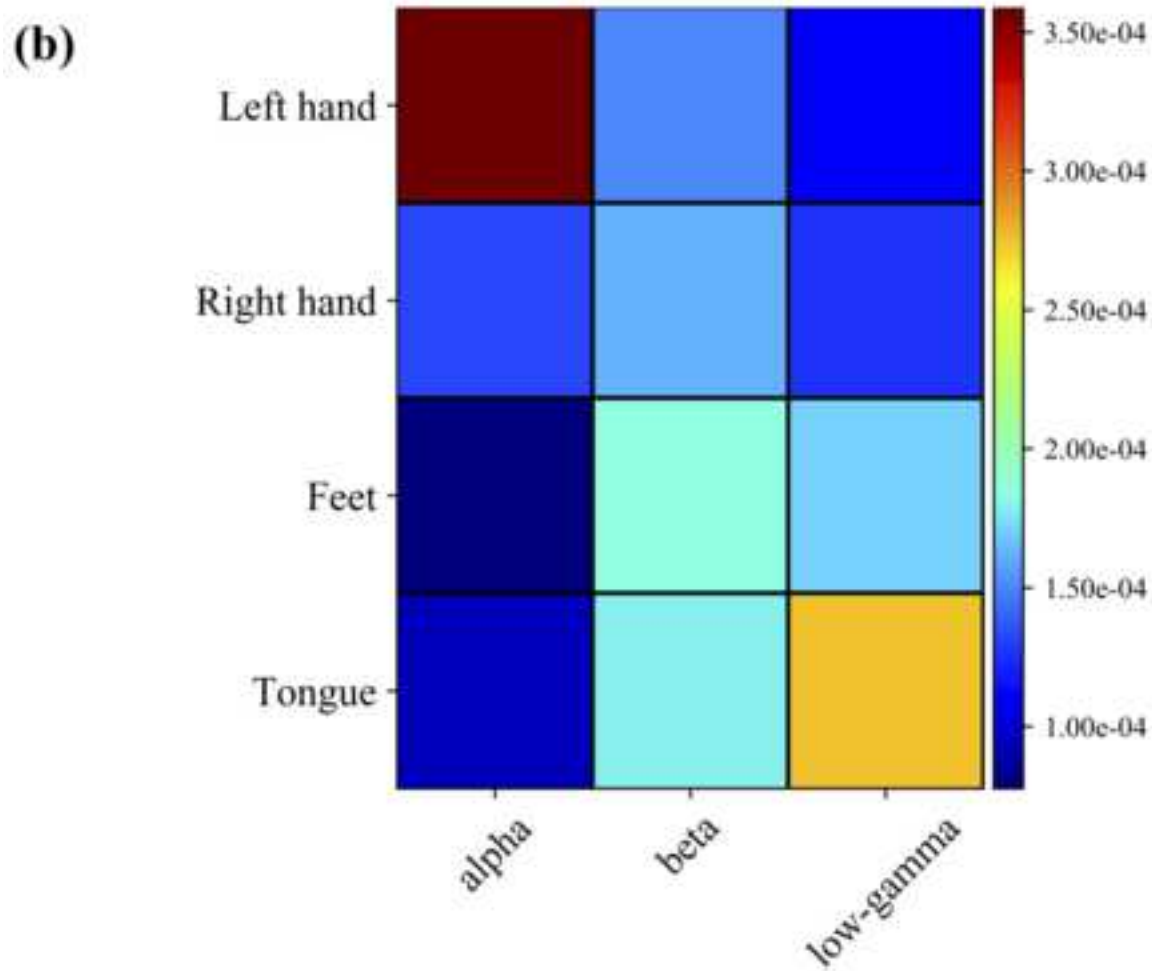
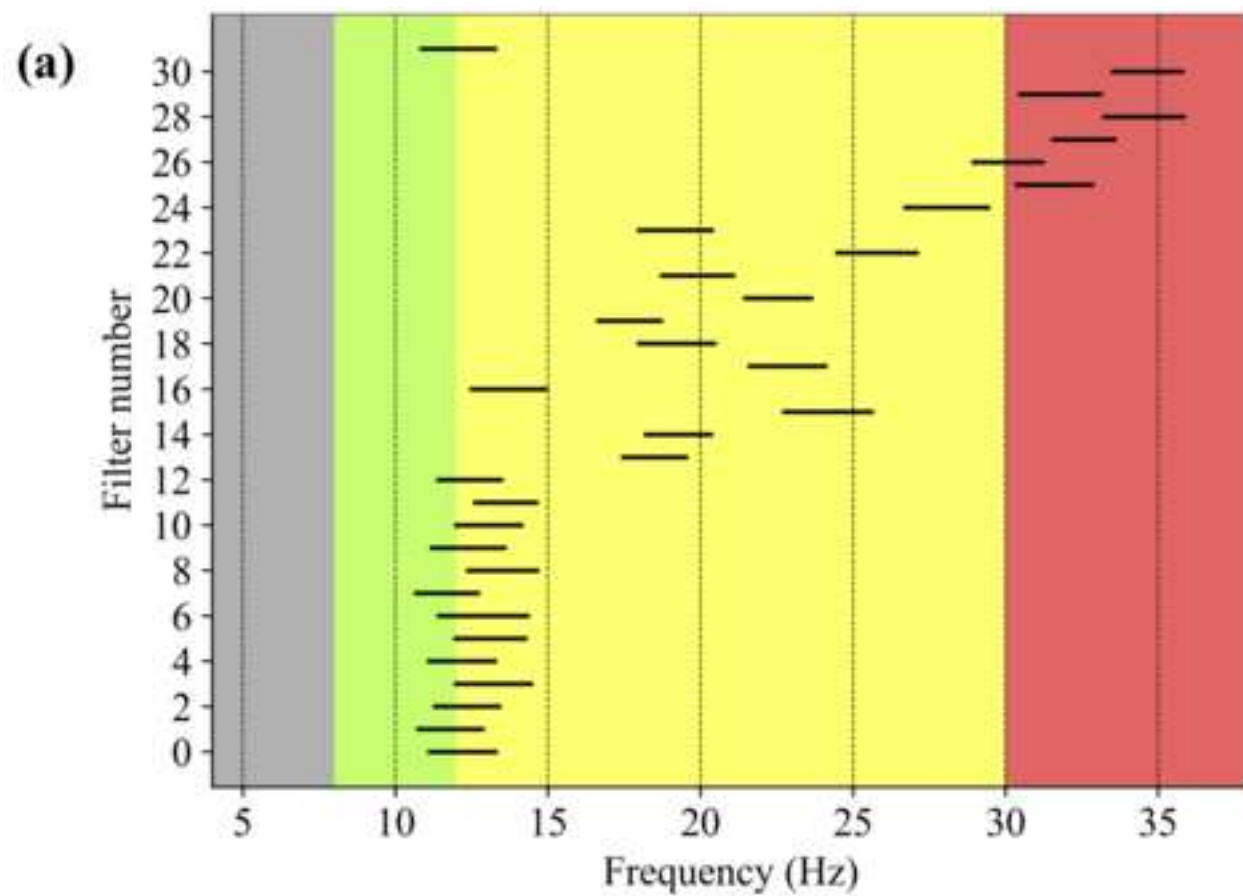


Figure 6



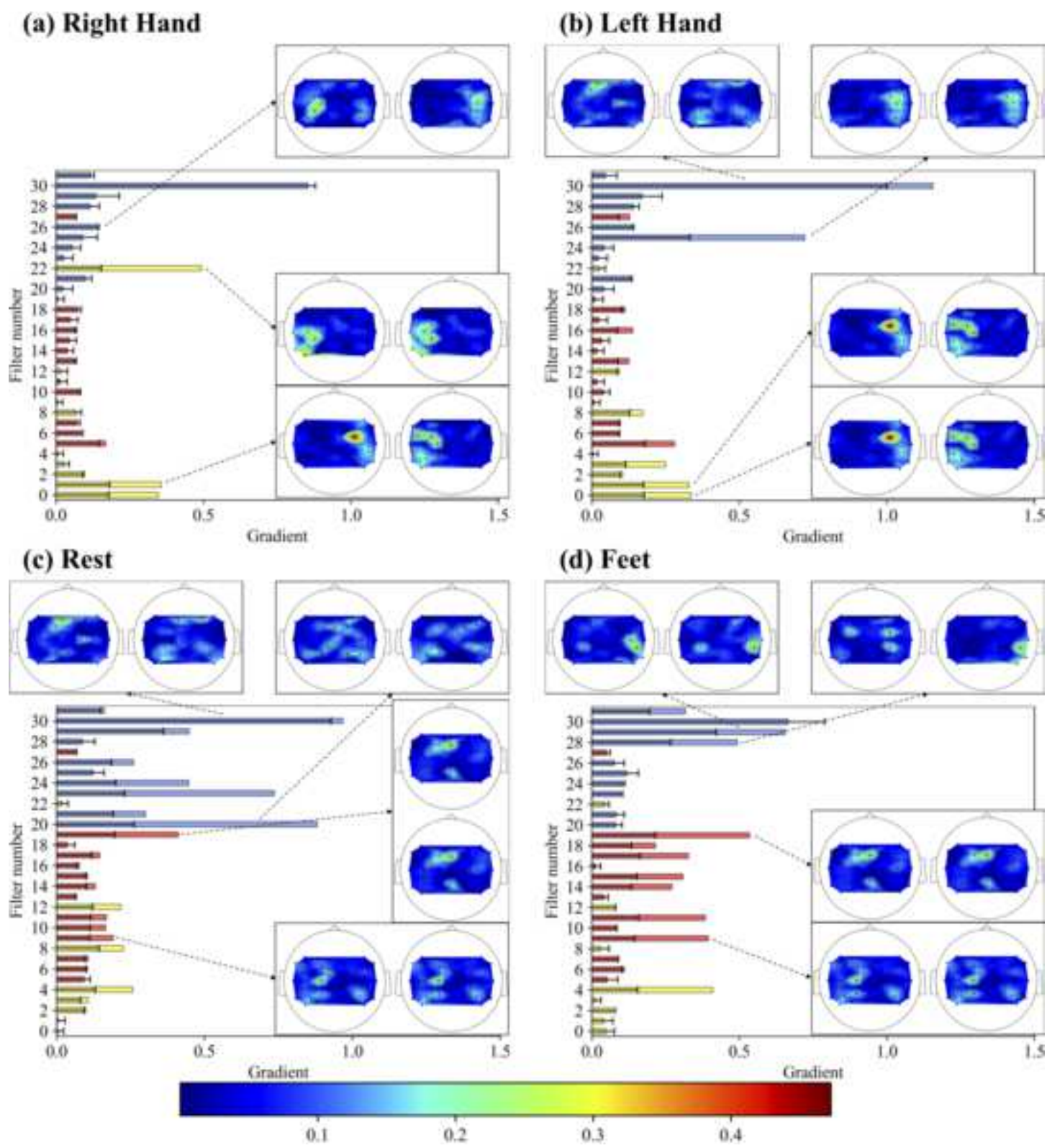
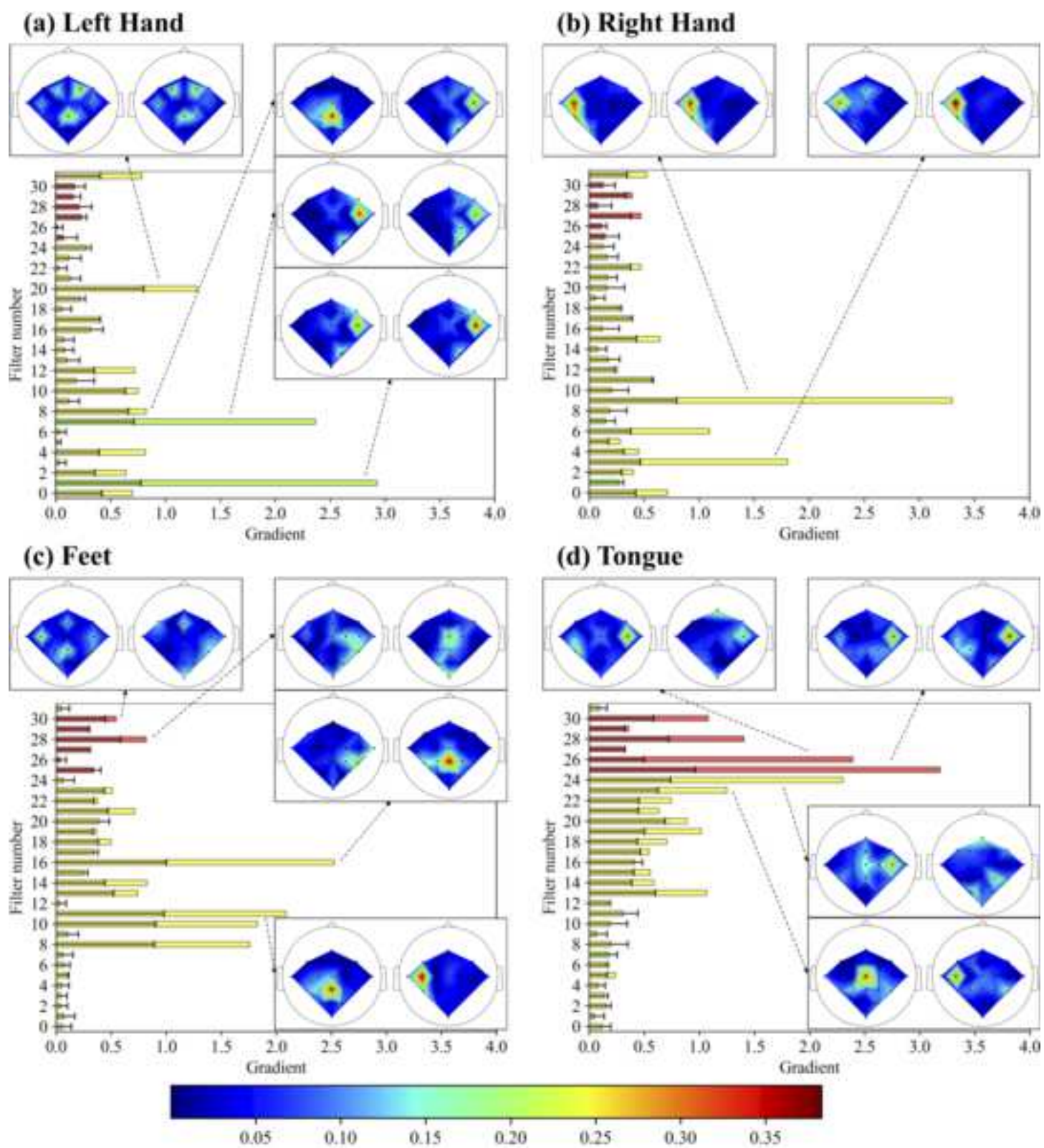


Figure 8



**Declaration of interests**

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

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: