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

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

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.

Title Page (With all author details listed)

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

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

ABSTRACT (max 250)

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

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

1. INTRODUCTION

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

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

An efficient way to provide EEG signals as input to CNNs is to design a 2D input representation with the electrodes along one dimension and time steps along the other (Borra et al., 2020a, 2020b; Cecotti & Graser, 2011; Farahat et al., 2019; Lawhern et al., 2018; Leeuwen et al., 2019; Manor & Geva, 2015; Schirrmeister et al., 2017; Shamwell et al., 2016; Tang et al., 2017; Zeng et al., 2019; Zhao et al., 2019), preserving the original EEG representation i.e. non-transformed representation. Other input representations, e.g. transformed representations such as time-frequency decomposition (Bashivan et al., 2015; Sakhavi et al., 2015; Tabar & Halici, 2016), generally increase data dimensionality requiring more training data and/or regularization to learn meaningful features. CNNs with a non-transformed representation are typically designed by stacking individual temporal and spatial convolutional layers or a single spatio-temporal convolutional layer, and eventually deeper convolutional layers that learn patterns on the spatially filtered activations. CNNs based on these architectures have been successfully applied to several EEG decoding tasks, such as P300 detection tasks (Borra et al., 2020a; Cecotti & Graser, 2011; Farahat et al., 2019; Lawhern et al., 2018; Manor & Geva, 2015; Shamwell et al., 2016), motor imagery and execution decoding tasks (Schirrmeister et al., 2017; Lawhern et al., 2018; Tang et al., 2017; Zhao et al., 2019; Borra et al., 2020b), anomaly detection tasks (Leeuwen et al., 2019), emotion classification (Zeng et al., 2019), and they have been generally proved to outperform traditional machine learning approaches. Despite these effective applications of CNNs in EEG decoding, there are still a number of critical issues that require further investigation. Indeed, CNNs introduce a large number of trainable parameters requiring large training datasets to obtain a good fit, have a longer training time compared to simpler models, introduce many hyper-parameters (e.g. number of kernels, kernel sizes, number of layers, type of activation functions, etc.), and the automatically learned features are difficult to be interpreted. In particular, techniques that increase the interpretability of the learned features are receiving growing interest as key ingredients to achieve more robust validation when using CNNs (Montavon et al., 2018). In the field of CNN-based EEG decoding, increasing the interpretability may be particularly relevant for neuroscientists as to the following aspects: (i) check the correct learning by verifying that the models do not rely on artefactual sources but on neurophysiological features; (ii) enable the understanding of which EEG features better discriminate the investigated classes; (iii) potentially characterize new features exploited by the network for the classification, and thus increase the insight into the neural correlates underlying the classified behaviours.

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

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

The purpose of this work is to contribute to the recent developments of CNN-based EEG decoding by designing and analysing a novel CNN that includes interpretable and optimized layers, able to increase the overall interpretability of the network, reduce the number of trainable parameters and, at the same time, ensure good performances compared to existing state-of-the art (SOA) algorithms. The CNN proposed here is a lightweight shallow CNN, named Sinc-ShallowNet, obtained by stacking two convolutional layers that extract spectral and spatial EEG features respectively, followed by a fully-connected layer finalizing the classification. The two convolutional layers are specifically devised to increase interpretability and decrease the number of trainable parameters and consist of a temporal sinc-convolutional layer and a spatial depthwise convolutional layer. The spatial depthwise convolutional layer ties spatial filters to each particular band-pass filter learned by the temporal sinc-convolutional layer, enabling the learning of spatial features related to specific frequency ranges. The proposed architecture was applied to decode sensorimotor rhythms both during motor execution (ME) and motor imagery (MI) using public benchmark datasets. Moreover, an extensive analysis of Sinc-ShallowNet was performed including the following aspects:

i. Comparison of the decoding performance of Sinc-ShallowNet with SOA decoding algorithms, including one traditional machine learning pipeline based on FBCSP coupled with regularized Linear Discriminant Analysis (rLDA) and other three CNNs (ShallowConvNet and DeepConvNet (Schirrmeister et al., 2017), EEGNet (Lawhern et al., 2018)).

ii. Assessment of some design choices on Sinc-ShallowNet performance in a post-hoc hyper-parameter evaluation procedure inspired by Schirrmeister et al. (2017). The evaluated design choices concern: the number of the temporal band-pass filters, the number of spatial filters for each temporal filter, the introduction of an optional recombination of the spatial activations, and the size of activation aggregation (average pooling) before the fully-connected layer.

iii. Evaluation of the effect of increasing the training data size via cropped training compared to trialwise training. Indeed, the effect of cropped training on different CNN architectures is still unclear. Schirrmeister et al. (2017) found that cropped training significantly increased the performance of deep architectures (DeepConvNet), while no significant effect was obtained with shallow architectures (ShallowConvNet). Despite this, other shallow architectures (Zhao et al., 2019) were trained with a cropped strategy. Therefore, we evaluated the effect of the training strategy on the performance of Sinc-ShallowNet and of the re-implemented SOA CNNs.

iv. Feature interpretation. Since the trainable parameters of the temporal sinc-convolutional layer are the cutoff frequencies of the learned band-pass filters, the learned spectral features can be directly visualized and interpreted once the training ends. Furthermore, inspired from the saliency maps (Simonyan et al., 2013), we designed a post-hoc interpretation technique named "temporal sensitivity analysis" (as it acts on the kernels of the temporal sinc-convolutional layer). This technique enables the identification of the more relevant and more class-specific band-pass filters and the spatial features (as learned in the depthwise convolutional layer) related to these band-pass filters can be visualized.

2. METHODS

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

2.1. Problem definition and notations

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

Section 2.2.2) movements of different body parts. Thus, the classes discriminate among the specific body part moved (or imagined to be moved) during each trial (e.g. l_0 = "Right Hand", l_1 = "Left Hand" etc.).

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

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

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 2D matrix of shape (C, T), having the time steps along the width and the electrodes along the height.

2.2. Datasets

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

2.2.1. Motor execution: High-Gamma dataset

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

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

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

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

2.2.2. Motor imagery: BCI-IV2a dataset

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

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

Here we used the same training set (288 trials per subject, balanced between the classes) and test set (288 trials per subject, balanced between the classes) provided by the organizers of the competition. The training set was further split into a validation set (20% of the training set) in order to perform early stopping during the first step of the optimization process (see Section 2.4).

[Figure 1 about here.]

2.3. Sinc-ShallowNet

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

[Figure 2 about here.]

[Table 1 about here.]

In the first two blocks, the output of each layer can be interpreted as a collection of spatio-temporal feature maps. Thus, its shape can be described by a tuple of 3 integers, with the first integer indicating the number of feature maps provided by the layer, the second and third integers the number of spatial and temporal samples within each map, respectively. Each convolutional layer in these blocks is characterized by the number of convolutional kernels (K), kernel size (F), stride size (S), and padding size (P). In addition, depthwise convolution introduces also a depth multiplier (D), that specifies the number of kernels to learn for each input feature map. Since Sinc-ShallowNet has two convolutional layers, the previous symbols are provided with subscript (“1”, “2”). The pooling layer in block 2 is described by the pool size (F_p) and pool stride (S_p). Since the adopted convolutions and 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 element refers to the spatial dimension, while the second to the temporal dimension.

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

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

2.3.1. Sinc-convolutional layer

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

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

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

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

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

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

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

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

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

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

The so-defined convolutional layer can be integrated into a traditional CNN to learn band-pass filters in the first layer, with only the two cutoff frequencies as trainable parameters. In this study, these frequencies were randomly initialized from a uniform distribution in the frequency range of interest: (4,125] Hz and (4,38] Hz for ME- and MI-EEG signals, respectively. During the optimization, these frequencies were updated in the range of interest by keeping $f_{2,j} > f_{1,j}$.

2.3.2. Block 1: Spectral and spatial feature extraction

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

2.3.3. Block 2: Aggregation

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

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

2.3.4. Block 3: Classification

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

Accordingly, the entire CNN maps the input data of the i-th trial $X_i^{(s)}$ to one real number per class, i.e. $h\left(X_i^{(s)}; \theta^{(s)}\right): \mathbb{R}^{C \times T} \longrightarrow \mathbb{R}^{N_c}$. These N_c outputs are transformed via a softmax activation function to obtain the conditional probabilities of the labels $l_k\ \forall k \in L = \{l_0, l_1, \dots, l_{N_c-1}\}$: $p\left(l_k \middle| X_i^{(s)}, \theta^{(s)}\right) = \frac{\exp h_k(X_i^{(s)}; \theta^{(s)})}{\sum_{j=0}^{N_c-1} h_j(X_i^{(s)}; \theta^{(s)})}$. Since the training strategy adopted was a within-subject training (see Section 2.4), the softmax provides subject-specific conditional distribution over the N_c classes for each example. The final classification is performed by assigning the label with the maximum conditional probability to the trial $X_i^{(s)}$, i.e. $y_i^{(s)} = f\left(X_i^{(s)}; \theta^{(s)}\right) = arg\ \max_{l_k} p\left(l_k \middle| X_i^{(s)}, \theta^{(s)}\right)$.

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

2.3.5. Design choices

In the following, the Sinc-ShallowNet described as in the previous sections (with the corresponding hyper-parameters, see Table 1) will be denoted as the “basal” Sinc-ShallowNet. In order to evaluate the influence of specific hyper-parameters of interest on the performance metric, a post-hoc hyper-parameter evaluation was performed by testing some variants compared to the basal architecture. The investigated hyper-parameters were: (i) the number of the temporal filters in block 1 (K_1); (ii) the number of the spatial filters per temporal filter in block 1 (D_2); (iii) the pooling size F_p and stride S_p of the average pooling in block 2; (iv) the recombination of the spatial activations. In the condition (iv), a pointwise convolution was included as the first layer in block 2 (fed by the outputs of the spatial depthwise convolution), followed by the other layers of block 2 (i.e. batch normalization, non-linear activation, average pooling, dropout).

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

[Table 2 about here.]

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

2.4. Training

2.4.1. Trialwise training strategy

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

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

where

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

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

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

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

2.4.2. Cropped training strategy

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

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

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

2.5. Regularization

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

i. Dropout (Srivastava et al., 2014). This technique randomly sets the outputs of the previous layer to zero with a probability p, during each training update. This helps to prevent co-adaptation (i.e. that some neurons are highly dependent to others) which could lead to overfitting. In the proposed network, dropout with $p = 0.5$ was introduced in block 2 immediately after the average pooling layer.

ii. Batch normalization (Ioffe & Szegedy, 2015). This technique mitigates a phenomenon named “internal covariate shift”, i.e. the change in the distribution of the layers’ activation due to the change of the trainable parameters during training (Ioffe & Szegedy, 2015). This phenomenon hinders the learning since the layers continuously need to adapt to the changed distribution while training and is particularly severe in deep neural networks. Batch normalization reduces the internal covariate shift, and consequently accelerates the training, by normalizing the output feature maps of intermediate layers to zero mean and unit variance across each training mini-batch. This technique introduces two trainable parameters since the normalization is followed by a channelwise affine transformation (that serves to maintain the expressive power of the network), whose parameters of scaling and shift are learned during training. Batch normalization enables higher learning rates without the risk of divergence, reduces the influence of a specific initialization scheme on the training, and also regularizes the model (Ioffe & Szegedy, 2015). While this technique is commonly used in deep neural networks, also shallow neural networks adopting batch normalization have been proposed in the literature. In particular, shallow CNNs including batch normalization have been recently applied to EEG signals for ME and MI decoding tasks (Lawhern et al., 2018; Schirrmeister et al., 2017), and for P300 detection (Liu et al., 2018). Importantly, Schirrmeister et al. (2017) reported an improved performance both in their shallow and deeper neural networks when using batch normalization compared to omitting it. Motivated by these previous results, we adopted this technique in our shallow CNN (blocks 1, 2) too, by applying it to the output of the convolutional layer immediately before the non-linearity, as recommended in the original paper (Ioffe & Szegedy, 2015), with a momentum term of $m = 0.99$ and with $\varepsilon = 1e - 3$ for numerical stability.

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

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

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

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

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

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

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

2.7. Interpretation

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

2.7.1. Temporal and spatial kernels visualization

The visualization of the learned kernels of the first block allows the interpretation of the temporal and spatial convolutional layers. The temporal sinc-convolutional layer introduced in the Sinc-ShallowNet architecture allows a direct interpretation of the learned parameters, which are the lower and upper cutoff frequencies $f_{1,j}$ and $f_{2,j}$ of the K_1 band-pass filters. Hence, for each subject, the distribution of the learned temporal kernels can be visualized by displaying how their passbands are distributed within the frequency range of the input signals (i.e. (4,125] Hz for ME- and (4,38] Hz for MI-EEG signals), and the preferred EEG rhythm (e.g. α, β, etc.) can be immediately derived. In particular, the following EEG bands b were considered: θ = (4,8] Hz, α = (8,12] Hz, β = (12,30] Hz, low γ = (30,50] Hz, high γ = (50, 125] Hz. A temporal filter was considered belonging to a specific band b if its central frequency fell within that band (actually, in most cases the band-pass filters had narrow passbands totally falling within a specific band range, see also Section 3.3 in Results).

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

2.7.2. Temporal sensitivity analysis

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

1. Gradient computation. Given a class k of interest and the i-th test trial of the s-th subject $X_i \in \mathbb{R}^{C \times T}$ 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 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 input) be the class score (i.e. output of the block 3 fully-connected layer, immediately before the softmax activation function). The class score z_k is a highly non-linear function of Y_j; given the input test trial X_i, this function can be approximated by a linear function in the neighbourhood of $Y_{i,j}$ by computing the first-order Taylor expansion (Simonyan et al., 2013) (Equation 8):

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

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

2. Gradient processing

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 the spatial and temporal dimension to obtain a scalar value $\overline{|G_{i,j,k}|}$.

b) Quantities $\overline{|G_{i,j,k}|}$ related to trials belonging to each specific class were averaged together, resulting in the absolute gradient value $g_{j,k}$ (scalar value):

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

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

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

$$\hat{g}_{j,k} = \frac{g_{j,k}}{\max\limits_{j,k} g_{j,k}}. \quad (10)$$

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

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

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

$$\hat{g}_{b,k} = \frac{1}{K_{1,b}} \sum_j \hat{g}_{j,k}. \tag{11}$$

In Equation 11, the sum runs over the $K_{1,b}$ band-pass filters belonging to the b band, i.e. $\left\{j : f_{c,j} = \frac{f_{1,j}+f_{2,j}}{2} \in (f_{1,b}, f_{2,b}]\right\}$, where $(f_{1,b}, f_{2,b}]$ denotes the frequency range of the band.

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

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

Based on this scaling, the quantity $\hat{g}'_{j,k}$ assumes larger values ($\gamma_{j,k} > 1$) when the impact of j-th temporal filter on the score of the specific class k is higher than its average impact on the other three classes; vice versa it assumes lower values ($\gamma_{j,k} < 1$) when the j-th temporal filter impacts on average more on the other three classes than on the considered k class. 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 having a discriminative power relatively heavier for that class than for the other classes on average. Thus, considering a class k, the more relevant and more class-specific temporal band-pass filters can be identified as the filters with $\gamma_{j,k} > 1$ and that scored higher $\hat{g}'_{j,k}$ values. Lastly, the spatial kernels associated with the so selected band-pass filters can be visualized as described in Section 2.7.1.

3. RESULTS

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

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

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

[Figure 3 about here.]

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

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

[Table 3 about here.]

[Table 4 about here.]

The proposed architecture scored an accuracy across subjects (mean ± std) of 91.2±9.1 % (inferior only to ShallowConvNet) and of 72.8±12.9 % (best overall) on ME- and MI-EEG signals, respectively. Compared to the baseline FBCSP+rLDA algorithm, ShallowConvNet and Sinc-ShallowNet performed significantly better on both ME- ($P = 0.024$, $P = 0.024$, respectively) and MI-EEG signals ($P = 0.046$, $P = 0.031$, respectively). Sinc-ShallowNet significantly outperformed DeepConvNet ($P = 0.026$) on ME-EEG signals, and both EEGNet ($P = 0.027$) and DeepConvNet ($P = 0.027$) on MI-EEG signals. Lastly, ShallowConvNet significantly outperformed Sinc-ShallowNet ($P = 0.040$) on ME-EEG signals; however, regarding this point, further considerations can be drawn from the results of the post-hoc hyper-parameter evaluation (see Section 4.2 in the Discussion).

3.2. Post-hoc hyper-parameter evaluation and training strategy evaluation

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

We evaluated the impact of cropped training compared to trialwise training on Sinc-ShallowNet (in its basal configuration) and on each re-implemented SOA CNNs. As detailed in Section 2.4.2, the trialwise training strategy adopted for this analysis was designed with a different epoching of the MI-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 follow the procedure used in (Schirrmeister et al., 2017). Nevertheless, we verified that no statistically significant difference in performance emerged between the trialwise training implemented with the different epoching of MI-EEG signals ($P = 0.441, P = 0.345, P = 0.347, P = 0.346$, respectively for DeepConvNet, ShallowConvNet, Sinc-ShallowNet and EEGNet.). The overall effect of cropped training on each CNN was quantified jointly on ME- and MI-EEG signals by computing the difference in accuracy between the cropped and the trialwise training strategies $\Delta_{acc} = acc_{cropped} - acc_{trialwise}$. The corresponding results are shown in Figure 4b. Only the deep architecture DeepConvNet significantly benefited from the cropped training strategy ($P = 0.002$), while shallower architectures such as Sinc-ShallowNet and EEGNet performed significantly worse when trained with the cropped strategy ($P = 0.008$ and $P = 0.009$).

[Figure 4 about here.]

3.3. Interpretation

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

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

[Figure 5 about here.]

[Figure 6 about here.]

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

[Figure 7 about here.]

[Figure 8 about here.]

4. DISCUSSION

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

i. Easy interpretation of both spectral and spatial features. The trainable parameters of the sinc-convolutional layer are directly interpretable (cutoff frequencies instead of mere kernel values as in a traditional convolutional layer) and the spatial filters are directly tied to specific band-pass filters.

ii. High optimization in terms of number of trainable parameters. The adopted sinc-convolution trains only 2 cutoff frequencies for each temporal filter and the depthwise convolution reduces the connections across the CNN channels.

iii. Computational efficiency. Due to the symmetry of the parametrized function adopted in the sinc-convolution, only half of the kernel values need to be computed.

In addition, the interpretation of the learned spectral and spatial features was further enriched thanks to the temporal sensitivity analysis; this analysis allows the identification of the more discriminative EEG bands (temporal sensitivity analysis at the level of EEG bands), and the more relevant and more class-specific band-pass filters (temporal sensitivity analysis at the level of single band-pass filter) together with their spatial distribution.

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

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

By comparing Sinc-ShallowNet with the re-implemented CNNs, the following considerations can be drawn. First, ShallowConvNet significantly outperformed Sinc-ShallowNet on ME- but not on MI- EEG signals (see Table 3). This is the only case in which Sinc-ShallowNet performed worse compared to the other considered CNNs. Nevertheless, it is worth noticing that Sinc-ShallowNet introduces 13828 and 5508 trainable parameters, that corresponds only to the 16.7% and 13.6% of those introduced by ShallowConvNet in case of ME- and MI-EEG signals (82564 and 40644), respectively. Therefore, the proposed architecture finalized the classification tasks in a more computationally efficient way, by introducing a lower number of trainable parameters. Furthermore, ShallowConvNet architecture was developed specifically for sensorimotor rhythm classification forcing the extraction of log band-power features (task-specific CNN), while Sinc-ShallowNet was not restricted to specific feature learning. Second, in the comparison with a general-purpose shallow architecture (EEGNet), Sinc-ShallowNet performed significantly better on MI-EEG signals, while performed comparably on ME-EEG signals. The lower performance of EEGNet may derive from the extremely lightweight architecture that used only $K_1 = 8$ temporal filters. Accordingly, the decoding of MI-EEG signals may benefit from a higher number of temporal filters (e.g. 32 as in the architecture proposed here). The introduction of the temporal sinc-convolutional layer that reduces the number of trainable parameters (i.e. only the two cutoff frequencies for each temporal filter) may be particularly beneficial for the decoding of MI-EEG dataset. Indeed, this dataset is characterized by a low number of training examples that requires the number of trainable parameters to be carefully maintained limited in order to avoid overfitting and achieve a good fit. Furthermore, when comparing Sinc-ShallowNet with DeepConvNet, the first provided significantly higher decoding accuracy on both ME- and MI-EEG signals. This may be attributable to the higher number of trainable parameters introduced by DeepConvNet (298229 and 278079 in case of ME- and MI-EEG signals, respectively), leading to an architecture more prone to overfitting especially in case of small datasets as for the adopted MI dataset.

4.2. Design choices of Sinc-ShallowNet

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

The analysis on D_2 and on the optional recombination deserves some comments. Increasing D_2 did not lead to significant increase in the performance. However, it is interesting to note that when the effect of D_2 was disaggregated between the two datasets (ME-EEG and MI-EEG dataset), an opposite behaviour tends to appear, with an average $\Delta_{acc} = +0.4\%$ and $\Delta_{acc} = -0.2\%$ on ME- and MI-EEG signals respectively (although not statistical significance was reached in either dataset). This different behaviour might be explained considering that when a CNN is trained with EEG signals containing a lower number of frequency components (such as MI-EEG signals), the band-pass temporal filters lie into a narrower frequency range and thus the probability that two different temporal filters have similar cutoff frequencies is higher. In this scenario, a lower number of spatial filters (D_2) for each temporal filter could be sufficient to retain enough capacity of the CNN, because close temporal filters could compensate for the lower D_2. Indeed, different spatial filters could be learned for similar temporal filters obtaining a cumulative set (across similar temporal filters) of band-specific spatial filters. Conversely, ME-EEG signals having wider frequency content can benefit from a larger number D_2 of spatial filters. Recombining the spatial activations via an additional pointwise convolutional layer did not improve accuracy. However, in this case too, by disaggregating the effect on the two datasets, an opposite behaviour tends to appear with an average $\Delta_{acc}= +0.5\%$ and $\Delta_{acc}= -2.6\%$ in case of ME- and MI-EEG signals respectively (although not statistical significance was reached in either dataset). This may be due to the learning of a useful recombination of frequency-specific spatial features learned across a wide frequency range, in case of signals with broad frequency content as ME-EEG signals. Finally, it is worth noticing that both increasing D_2 and including a pointwise convolutional layer lead to an increase in the number of trainable parameters that might be critical in applications involving small datasets (e.g. the adopted MI dataset). Overall, these considerations remain quite speculative and further experiments are required, for example testing Sinc-ShallowNet and its different design choices on other datasets having larger and smaller size than those used here and having various frequency contents. However, it is interesting to note that the small accuracy increase ($\Delta_{acc}= +0.5\%$) in case of ME-EEG signals obtained introducing the pointwise convolutional layer led to a significant better performance of Sinc-ShallowNet compared to EEGNet ($P = 0.046$) and to comparable performance with ShallowConvNet ($P = 0.090$); at the same time, the accuracy decrease in case of MI-EEG signals ($\Delta_{acc}= -2.6\%$) did not change the statistical significance ($P = 0.049$ vs. EEGNet and DeepConvnet, $P = 0.340$ vs. ShallowConvNet). Thus, the proposed Sinc-ShallowNet architecture integrated with the recombination of the spatial activations led to a CNN that performs better than or at least as well as the SOA CNNs on both datasets, at the expense of the number of trainable parameters (17924 and 9604 in case of ME and MI datasets respectively).

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

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

4.3. Training strategies

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

4.4. Interpretation

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

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

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

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

Thanks to the use of spatial depthwise convolution, the proposed architecture ties spatial kernels to each band-pass filter and thus, the relevance, as quantified by the temporal sensitivity analysis, can be propagated from each band-pass filter to the associated spatial filters. In particular, the more relevant and more class-specific spatial filters can be identified – as those associated to the band-pass filters scored by the highest rescaled gradients $\hat{g}'_{j,k}$, (i.e. temporal sensitivity analysis at the level of single band-pass filter) – and visualized. These spatial filters show a highly localized distributions in the scalp maps (Figures 7a-7d and 8a-8d, respectively for ME- and MI-EEG signals). Among the spatial filters specific for the hand movements, some filters have the most discriminative electrodes located in the contralateral hemisphere to the executed and imagined hand movement, approximately above the primary sensorimotor hand representation areas (i.e. around C3 and C4). Regarding the executed and imagined feet movements, some filters have the most discriminative electrodes located more centrally, approximately above the primary motor foot area (i.e. around CPz, Cz and FCz). Finally, regarding the imagined tongue movement, the most discriminative electrodes are placed not only around C3 and C4, but also approximately above the somatosensory cortex (i.e. area below Cz), representing the brain region triggered by the imagination of tongue movements (Zhao et al., 2019).

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

As underlined previously, the interpretation capabilities of the network are provided by coupling an interpretable layer (sinc-convolutional layer) with an optimized layer (depthwise convolutional layer), and by using a post-hoc gradient-based technique alongside with spatial and temporal filter visualizations. Therefore, interpretation capabilities of Sinc-ShallowNet are intrinsically linked to some specific design choices and specifically implemented post-hoc analyses. However, other more general-purpose techniques adopted in our network (e.g. batch normalization or dropout), that introduce a regularization effect, contribute to increase the neurophysiological reliability of feature interpretation by improving the performance on unseen examples. For example, we verified that when training Sinc-ShallowNet by removing the batch normalization layers in the blocks 1, 2 (and leaving all the other hyper-parameters unchanged), a significant decrease of the decoding accuracies occurred: $\Delta_{acc}= -4.8\%$ ($P = 0.002$, Wilcoxon signed-rank test), $\Delta_{acc}= -14.8\%$ ($P = 0.008$, Wilcoxon signed-rank test) respectively for ME- and MI-EEG signals, where $\Delta_{acc}= acc_{w/o\ BN} - acc_{w/\ BN}$. These simulations confirmed the important regularization introduced by batch normalization that significantly increased network accuracy on unseen examples. Accordingly, although batch normalization does not contribute directly to the interpretation capabilities of the network (omitting it the inner interpretation capabilities of the network are not altered), its inclusion increases the neurophysiological significance of the interpreted features via accuracy improvement. Indeed, the band-pass filters and spatial filters learned by the batch-normalized Sinc-ShallowNet turn out to be more class-discriminative (as they provide higher accuracies). Therefore, the learned spectral and spatial features are more likely to reflect neurophysiological aspects (in terms of more relevant EEG bands and electrodes) linked to the investigated tasks (i.e. motor execution and motor imagery decoding).

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

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

5. CONCLUSIONS

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

APPENDIX A: State-of-the-art CNNs

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

[Table A.1 about here.]

[Table A.2 about here.]

[Table A.3 about here.]

APPENDIX B: FBCSP+rLDA

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

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

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

Acknowledgments

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

Conflicts of interest

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

References

Ang, K. K., Chin, Z. Y., Wang, C., Guan, C., & Zhang, H. (2012). Filter Bank Common Spatial Pattern Algorithm on BCI Competition IV Datasets 2a and 2b. *Frontiers in Neuroscience*, *6*, 39. https://doi.org/10.3389/fnins.2012.00039

Ang, K. K., Chin, Z. Y., Zhang, H., & Guan, C. (2008). Filter bank common spatial pattern (FBCSP) in brain-computer interface. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2390–2397.

Ball, T., Demandt, E., Mutschler, I., Neitzel, E., Mehring, C., Vogt, K., Aertsen, A., & Schulze-Bonhage, A. (2008). Movement related activity in the high gamma range of the human EEG. *NeuroImage*, *41*, 302–310.

Bashashati, A., Fatourechi, M., Ward, R. K., & Birch, G. E. (2007). A survey of signal processing algorithms in brain–computer interfaces based on electrical brain signals. *Journal of Neural Engineering*, *4*(2), R32–R57.

Bashivan, P., Rish, I., Yeasin, M., & Codella, N. (2015). Learning Representations from EEG with Deep Recurrent-Convolutional Neural Networks. *CoRR*, *abs/1511.06448*.

Benjamini, Y., & Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, *57*(1), 289–300.

Blankertz, B., Tomioka, R., Lemm, S., Kawanabe, M., & Muller, K. (2008). Optimizing Spatial filters for Robust EEG Single-Trial Analysis. *IEEE Signal Processing Magazine*, *25*(1), 41–56. https://doi.org/10.1109/MSP.2008.4408441

Borra, D., Fantozzi, S., & Magosso, E. (2020a). Convolutional Neural Network for a P300 Brain-Computer Interface to Improve Social Attention in Autistic Spectrum Disorder. In J. Henriques, N. Neves, & P. de Carvalho (Eds.), *XV Mediterranean Conference on Medical and Biological Engineering and Computing – MEDICON 2019* (pp. 1837–1843). Springer International Publishing.

Borra, D., Fantozzi, S., & Magosso, E. (2020b). EEG Motor Execution Decoding via Interpretable Sinc-Convolutional Neural Networks. In J. Henriques, N. Neves, & P. de Carvalho (Eds.), *XV Mediterranean Conference on Medical and Biological Engineering and Computing – MEDICON 2019* (pp. 1113–1122). Springer International Publishing.

Cecotti, H., & Graser, A. (2011). Convolutional Neural Networks for P300 Detection with Application to Brain-Computer Interfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(3), 433–445.

Chin, Z. Y., Ang, K. K., Wang, C., Guan, C., & Zhang, H. (2009). Multi-class filter bank common spatial pattern for four-class motor imagery BCI. *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 571–574. https://doi.org/10.1109/IEMBS.2009.5332383

Chollet, F. (2016). Xception: Deep Learning with Depthwise Separable Convolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1800–1807.

Clevert, D.-A., Unterthiner, T., & Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). *ArXiv Preprint*.

Crone, N. E., Miglioretti, D. L., Gordon, B., & Lesser, R. P. (1998). Functional mapping of human sensorimotor cortex with electrocorticographic spectral analysis. II. Event-related synchronization in the gamma band. *Brain : A Journal of Neurology*, *121 (Pt 12)*, 2301–2315.

Farahat, A., Reichert, C., Sweeney-Reed, C., & Hinrichs, H. (2019). Convolutional neural networks for decoding of covert attention focus and saliency maps for EEG feature visualization. *Journal of Neural Engineering*. http://iopscience.iop.org/10.1088/1741-2552/ab3bb4

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 249–256.

Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., & Bengio, Y. (2013). Maxout networks. *Proceedings of the 30th International Conference on International Conference on Machine Learning-Volume 28*, III–1319.

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In F. Bach & D. Blei (Eds.), *Proceedings of the 32nd International Conference on Machine Learning* (Vol. 37, pp. 448–456). PMLR.

Jonas, S., Rossetti, A. O., Oddo, M., Jenni, S., Favaro, P., & Zubler, F. (2019). EEG-based outcome prediction after cardiac arrest with convolutional neural networks: Performance and visualization of discriminative features. *Human Brain Mapping*, *40*(16), 4606–4617. https://doi.org/10.1002/hbm.24724

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *ArXiv Preprint*.

Lawhern, V. J., Solon, A. J., Waytowich, N. R., Gordon, S. M., Hung, C. P., & Lance, B. J. (2018). EEGNet: A compact convolutional neural network for EEG-based brain–computer interfaces. *Journal of Neural Engineering*, *15*(5), 056013.

Ledoit, O., & Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis, 88*(2), 365–411. https://doi.org/10.1016/S0047-259X(03)00096-4

Leeuwen, K. G. van, Sun, H., Tabaeizadeh, M., Struck, A. F., Putten, M. J. A. M. van, & Westover, M. B. (2019). Detecting abnormal electroencephalograms using deep convolutional networks. *Clinical Neurophysiology*, *130*(1), 77–84.

Liu, M., Wu, W., Gu, Z., Yu, Z., Qi, F., & Li, Y. (2018). Deep learning based on Batch Normalization for P300 signal detection. *Neurocomputing*, *275*, 288–297. https://doi.org/10.1016/j.neucom.2017.08.039

Lotte, F. (2015). Signal Processing Approaches to Minimize or Suppress Calibration Time in Oscillatory Activity-Based Brain–Computer Interfaces. *Proceedings of the IEEE*, *103*(6), 871–890. https://doi.org/10.1109/JPROC.2015.2404941

Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., & Yger, F. (2018). A review of classification algorithms for EEG-based brain–computer interfaces: A 10 year update. *Journal of Neural Engineering*, *15*(3), 031005.

Mak, J. N., & Wolpaw, J. R. (2009). Clinical Applications of Brain-Computer Interfaces: Current State and Future Prospects. *IEEE Reviews in Biomedical Engineering*, *2*, 187–199. https://doi.org/10.1109/RBME.2009.2035356

Manor, R., & Geva, A. B. (2015). Convolutional Neural Network for Multi-Category Rapid Serial Visual Presentation BCI. *Frontiers in Computational Neuroscience*, *9*, 146. https://doi.org/10.3389/fncom.2015.00146

McFarland, D. J., Anderson, C. W., Muller, K.-, Schlogl, A., & Krusienski, D. J. (2006). BCI meeting 2005-workshop on BCI signal processing: Feature extraction and translation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *14*(2), 135–138.

Mirnaziri, M., Rahimi, M., Alavikakhaki, S., & Ebrahimpour, R. (2013). Using Combination of μ,β and γ Bands in Classification of EEG Signals. *Basic and Clinical Neuroscience*.

Montavon, G., Samek, W., & Müller, K.-R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, *73*, 1–15. https://doi.org/10.1016/j.dsp.2017.10.011

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. (2017). Automatic differentiation in PyTorch. *NIPS-W*.

Pfurtscheller, G. (1981). Central beta rhythm during sensorimotor activities in man. *Electroencephalography and Clinical Neurophysiology*, *51*(3), 253–264. https://doi.org/10.1016/0013-4694(81)90139-5

Pfurtscheller, G., & Aranibar, A. (1977). Event-related cortical desynchronization detected by power measurements of scalp EEG. *Electroencephalography and Clinical Neurophysiology*, *42*(6), 817–826. https://doi.org/10.1016/0013-4694(77)90235-8

Pfurtscheller, G., & Berghold, A. (1989). Patterns of cortical activation during planning of voluntary movement. *Electroencephalography and Clinical Neurophysiology*, *72*(3), 250–258. https://doi.org/10.1016/0013-4694(89)90250-2

Pfurtscheller, G., Brunner, C., Schlögl, A., & Silva, F. H. L. da. (2006). Mu rhythm (de)synchronization and EEG single-trial classification of different motor imagery tasks. *NeuroImage*, *31*(1), 153–159. https://doi.org/10.1016/j.neuroimage.2005.12.003

Pfurtscheller, G., & Silva, F. H. L. da. (1999). Event-related EEG/MEG synchronization and desynchronization: Basic principles. *Clinical Neurophysiology*, *110*(11), 1842–1857. https://doi.org/10.1016/S1388-2457(99)00141-8

Pfurtscheller, Gert, Flotzinger, D., & Neuper, C. (1994). Differentiation between finger, toe and tongue movement in man based on 40 Hz EEG. *Electroencephalography and Clinical Neurophysiology*, *90*(6), 456–460. https://doi.org/10.1016/0013-4694(94)90137-6

Ravanelli, M., & Bengio, Y. (2018). Speaker Recognition from Raw Waveform with SincNet. *2018 IEEE Spoken Language Technology Workshop (SLT)*, 1021–1028. https://doi.org/10.1109/SLT.2018.8639585

Roy, Y., Banville, H., Albuquerque, I., Gramfort, A., Falk, T. H., & Faubert, J. (2019). Deep learning-based electroencephalography analysis: A systematic review. *Journal of Neural Engineering*, *16*(5), 051001. https://doi.org/10.1088/1741-2552/ab260c

Sakhavi, S., Guan, C., & Yan, S. (2015). Parallel convolutional-linear neural network for motor imagery classification. *2015 23rd European Signal Processing Conference (EUSIPCO)*, 2736–2740. https://doi.org/10.1109/EUSIPCO.2015.7362882

Schirrmeister, R. T., Springenberg, J. T., Fiederer, L. D. J., Glasstetter, M., Eggensperger, K., Tangermann, M., Hutter, F., Burgard, W., & Ball, T. (2017). Deep learning with convolutional neural networks for EEG decoding and visualization. *Human Brain Mapping*, *38*(11), 5391–5420.

Shamwell, J., Lee, H., Kwon, H., Marathe, A. R., Lawhern, V., & Nothwang, W. (2016). Single-trial EEG RSVP classification using convolutional neural networks. In T. George, A. K. Dutta, & M. S. Islam (Eds.), *Micro- and Nanotechnology Sensors, Systems, and Applications VIII* (Vol. 9836, pp. 373 – 382). SPIE. https://doi.org/10.1117/12.2224172

Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *ArXiv Preprint*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

Tabar, Y. R., & Halici, U. (2016). A novel deep learning approach for classification of EEG motor imagery signals. *Journal of Neural Engineering*, *14*(1), 016003.

Tang, Z., Li, C., & Sun, S. (2017). Single-trial EEG classification of motor imagery using deep convolutional neural networks. *Optik*, *130*, 11–18. https://doi.org/10.1016/j.ijleo.2016.10.117

Tangermann, M., Müller, K.-R., Aertsen, A., Birbaumer, N., Braun, C., Brunner, C., Leeb, R., Mehring, C., Miller, K. J., Mueller-Putz, G., & others. (2012). Review of the BCI competition IV. *Frontiers in Neuroscience*, *6*, 55.

Zeng, H., Wu, Z., Zhang, J., Yang, C., Zhang, H., Dai, G., & Kong, W. (2019). EEG Emotion Classification Using an Improved SincNet-Based Deep Learning Model. *Brain Sciences*, *9*(11). https://doi.org/10.3390/brainsci9110326

Zhao, D., Tang, F., Si, B., & Feng, X. (2019). Learning joint space–time–frequency features for EEG decoding on small labeled data. *Neural Networks*, *114*, 67–77.

LEGENDS TO FIGURES

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

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

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

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

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

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

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

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

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

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

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

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 were 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 they may be tied with different spatial filters learned during training, actually providing an augmented set of spatial filters for a given band-pass filtering. The drawback of an increase of D_2 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 corresponded to the extraction of averaged spatial activations of 325 ms with a stride of 70 ms (similarly as done in (Schirrmeister 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 large training set. Thus, this modification could be more beneficial in case of the investigated ME-EEG signals.

Table 3 – Accuracies (mean ± std across subjects) of the basal Sinc-ShallowNet and SOA algorithms, obtained with ME-EEG signals belonging to the test set. Here, the trialwise training was adopted. For each CNN, the total number of trainable parameters is reported in brackets. The corrected P values 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±9.0		
EEGNet (2604)	88.5±11.0	0.158	0.158
DeepConvNet (298229)	88.4±8.8	0.158	0.026
ShallowConvNet (82564)	93.9±9.3	0.024	0.040
Sinc-ShallowNet (13828)	91.2±9.1	0.024	

Table 4 – Accuracies (mean ± std across subjects) of the basal Sinc-ShallowNet and SOA algorithms, obtained with MI-EEG signals belonging to the test set. Here, the trialwise training was adopted (signal epoching 0.5-2.5 s). For each CNN, the total number of trainable parameters is reported in brackets. The corrected P values 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	67.5±13.9		
EEGNet (1932)	66.0±13.1	0.575	0.027
DeepConvNet (278079)	50.5±19.6	0.031	0.027
ShallowConvNet (40644)	71.6±14.2	0.046	0.302
Sinc-ShallowNet (5508)	72.8±12.9	0.031	

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

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

Table A.2 – Architecture details of DeepConvNet. Each layer is provided with its name, main hyper-parameters, number of trainable parameters and activation function. See Section 2.3 for the meaning of the symbols. *Kernel maximum norm constraint at 2 and 0.5, respectively for the convolutional and fully-connected layers. For numerical stability, batch normalization ε 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

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

Layer name	Hyper-parameters	Number of parameters	Activation
Input		0	
Conv2D*	$K_1 = 40$	$F_1[1] \cdot K_1 + K_1$	Linear
	$F_1 = (1{,}25)$		
	$S_1 = (1{,}1)$		
	$P_1 = (0{,}0)$		
Conv2D*	$K_2 = 40$	$K_1 \cdot F_2[0] \cdot K_2$	Linear
	$F_2 = (C, 1)$		
	$S_2 = (1{,}1)$		
	$P_2 = (0{,}0)$		
BatchNorm2D	$m = 0.9$	$2 \cdot K_2$	
Activation	$\alpha = 1$	0	Square
AvgPool2D	$F_p = (1{,}75)$	0	
	$S_p = (1{,}15)$		
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

Figure 1

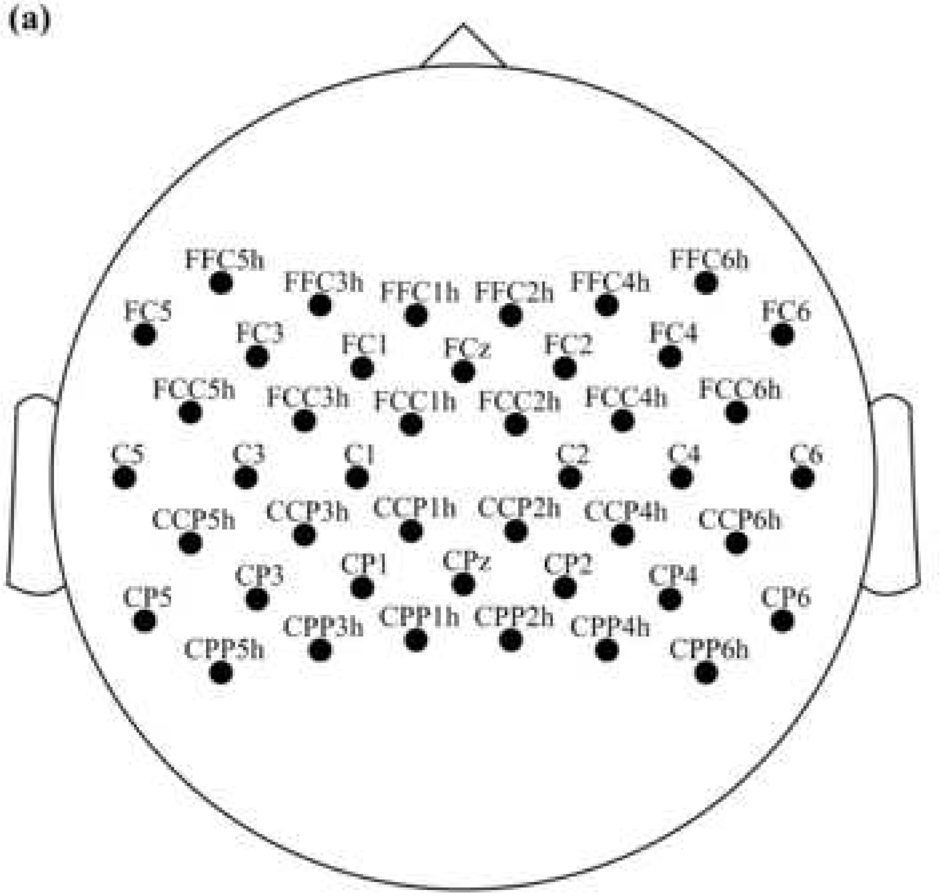

(a)
FFC5h
FFC3h
FFC1h
FFC2h
FFC4h
FFC6h
FC5
FC3
FC1
FCz
FC2
FC4
FC6
FCC5h
FCC3h
FCC1h
FCC2h
FCC4h
FCC6h
C5
C3
C1
C2
C4
C6
CCP5h
CCP3h
CCP1h
CCP2h
CCP4h
CCP6h
CP5
CP3
CP1
CPz
CP2
CP4
CP6
CPP5h
CPP3h
CPP1h
CPP2h
CPP4h
CPP6h
(b)
Fz
FC3
FC1
FCz
FC2
FC4
C5
C3
C1
Cz
C2
C4
C6
CP3
CP1
CPz
CP2
CP4
P1
Pz
P2
POz

Figure 2

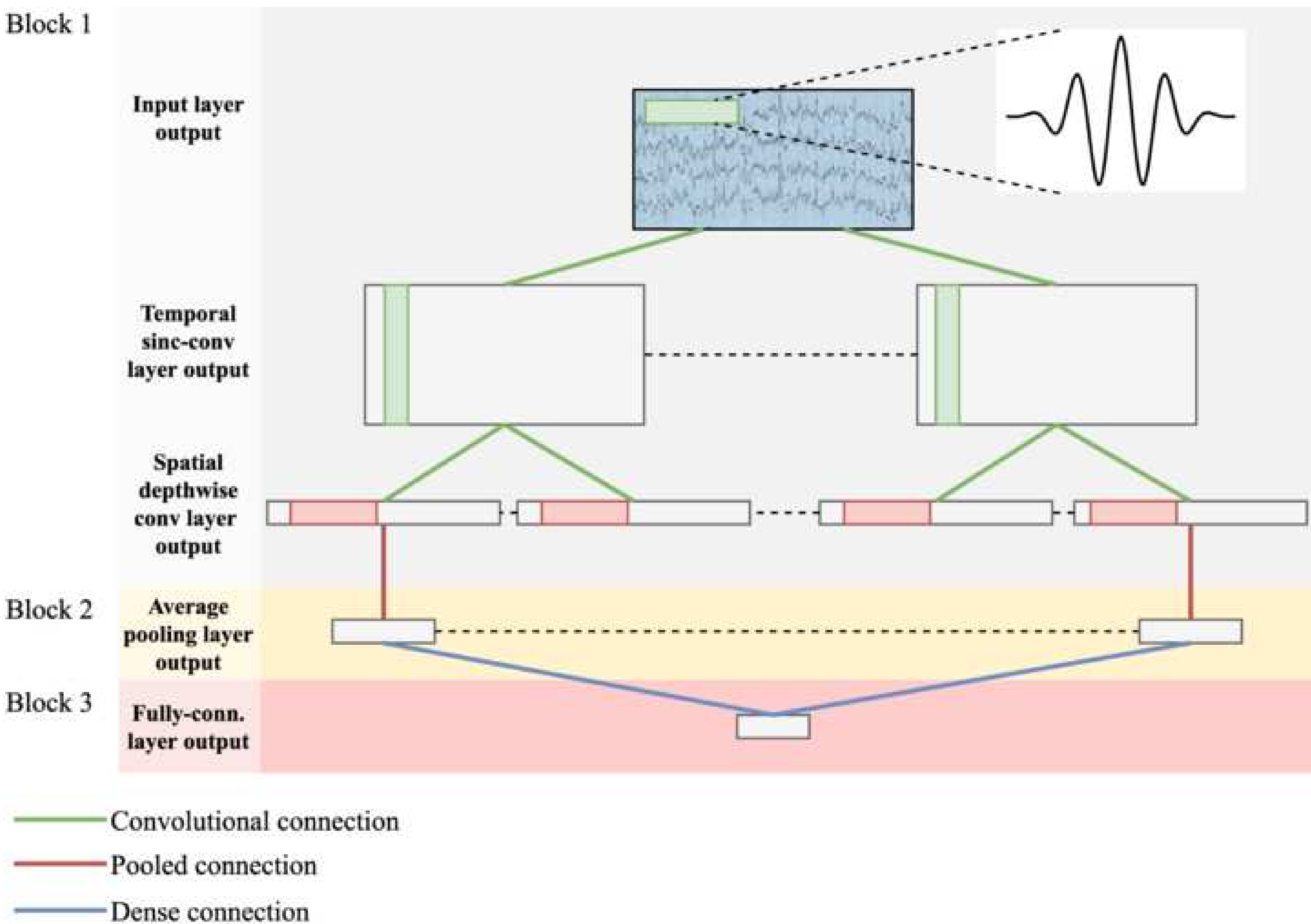
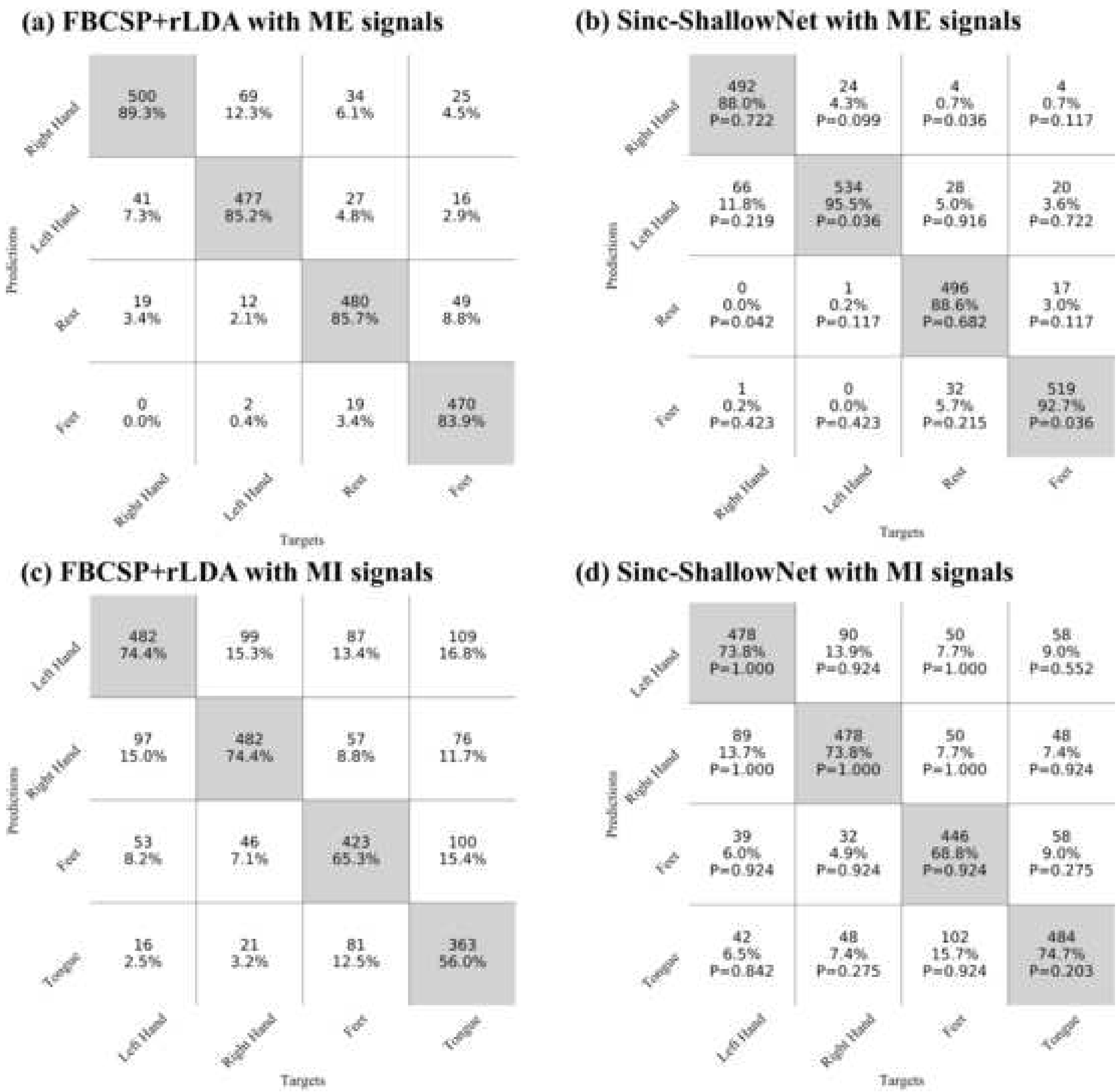
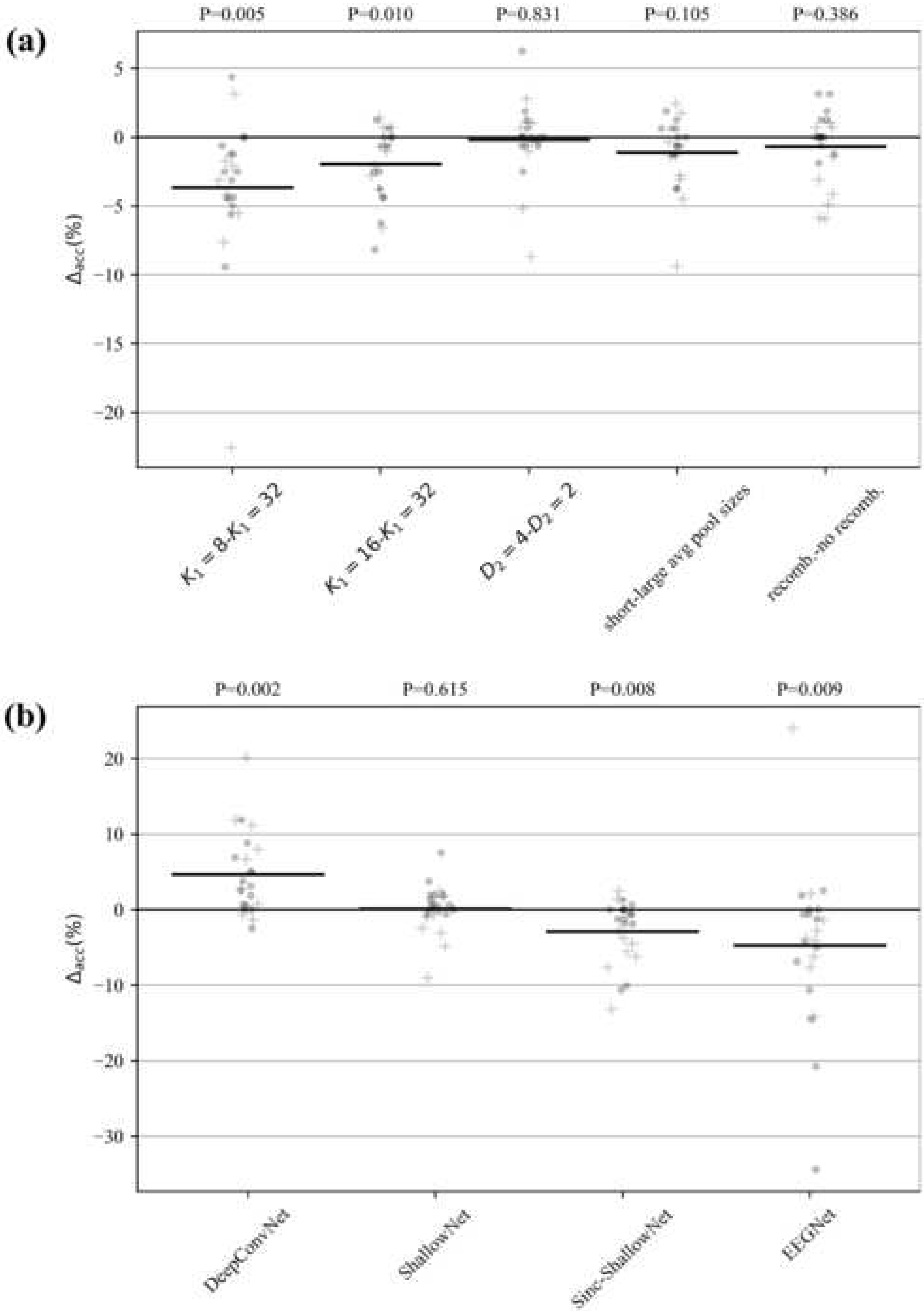

Block 1
Input layer output
Temporal sinc-conv layer output
Spatial depthwise conv layer output
Block 2
Average pooling layer output
Block 3
Fully-conn. layer output
Convolutional connection
Pooled connection
Dense connection

Figure 3



(a) FBCSP+rLDA with ME signals
(b) Sinc-ShallowNet with ME signals
(c) FBCSP+rLDA with MI signals
(d) Sinc-ShallowNet with MI signals
Predictions
Targets
Right Hand
Left Hand
Rest
Feet
Tongue
500 89.3%
69 12.3%
34 6.1%
25 4.5%
41 7.3%
477 85.2%
27 4.8%
16 2.9%
19 3.4%
12 2.1%
480 85.7%
49 8.8%
0 0.0%
2 0.4%
19 3.4%
470 83.9%
492 88.0% P=0.722
24 4.3% P=0.099
4 0.7% P=0.036
4 0.7% P=0.117
66 11.8% P=0.219
534 95.5% P=0.036
28 5.0% P=0.916
20 3.6% P=0.722
0 0.0% P=0.042
1 0.2% P=0.117
496 88.6% P=0.682
17 3.0% P=0.117
1 0.2% P=0.423
0 0.0% P=0.423
32 5.7% P=0.215
519 92.7% P=0.036
482 74.4%
99 15.3%
87 13.4%
109 16.8%
97 15.0%
482 74.4%
57 8.8%
76 11.7%
53 8.2%
46 7.1%
423 65.3%
100 15.4%
16 2.5%
21 3.2%
81 12.5%
363 56.0%
478 73.8% P=1.000
90 13.9% P=0.924
50 7.7% P=1.000
58 9.0% P=0.552
89 13.7% P=1.000
478 73.8% P=1.000
50 7.7% P=1.000
48 7.4% P=0.924
39 6.0% P=0.924
32 4.9% P=0.924
446 68.8% P=0.924
58 9.0% P=0.275
42 6.5% P=0.842
48 7.4% P=0.275
102 15.7% P=0.924
484 74.7% P=0.203

Figure 4



(a)
P=0.005
P=0.010
P=0.831
P=0.105
P=0.386
Δacc(%)
5
0
−5
−10
−15
−20
K1 = 8-K1 = 32
K1 = 16-K1 = 32
D2 = 4-D2 = 2
short-large avg pool sizes
recomb.-no recomb.
(b)
P=0.002
P=0.615
P=0.008
P=0.009
Δacc(%)
20
10
0
−10
−20
−30
DeepConvNet
ShallowNet
Sinc-ShallowNet
EEGNet

Figure 5

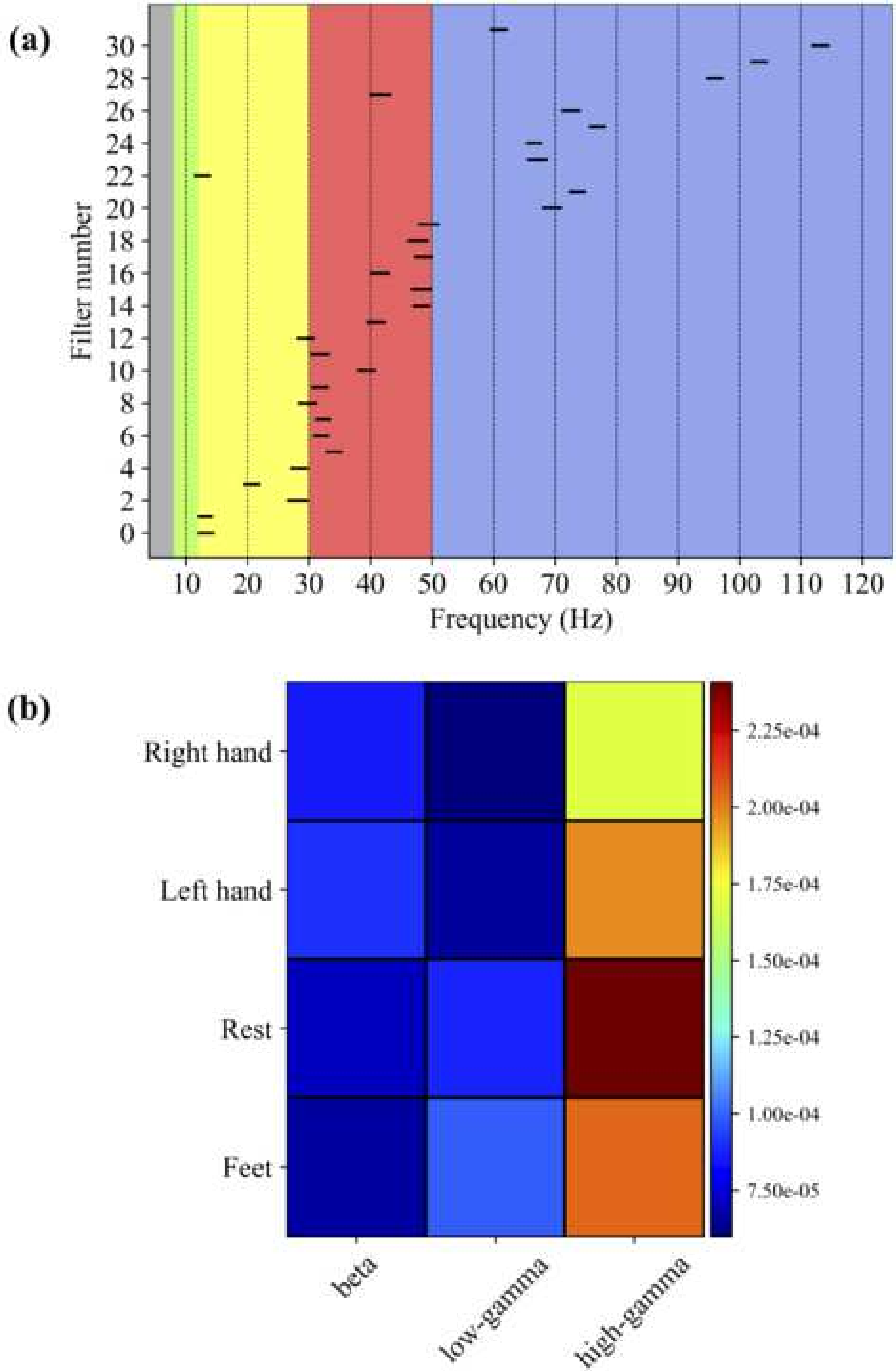
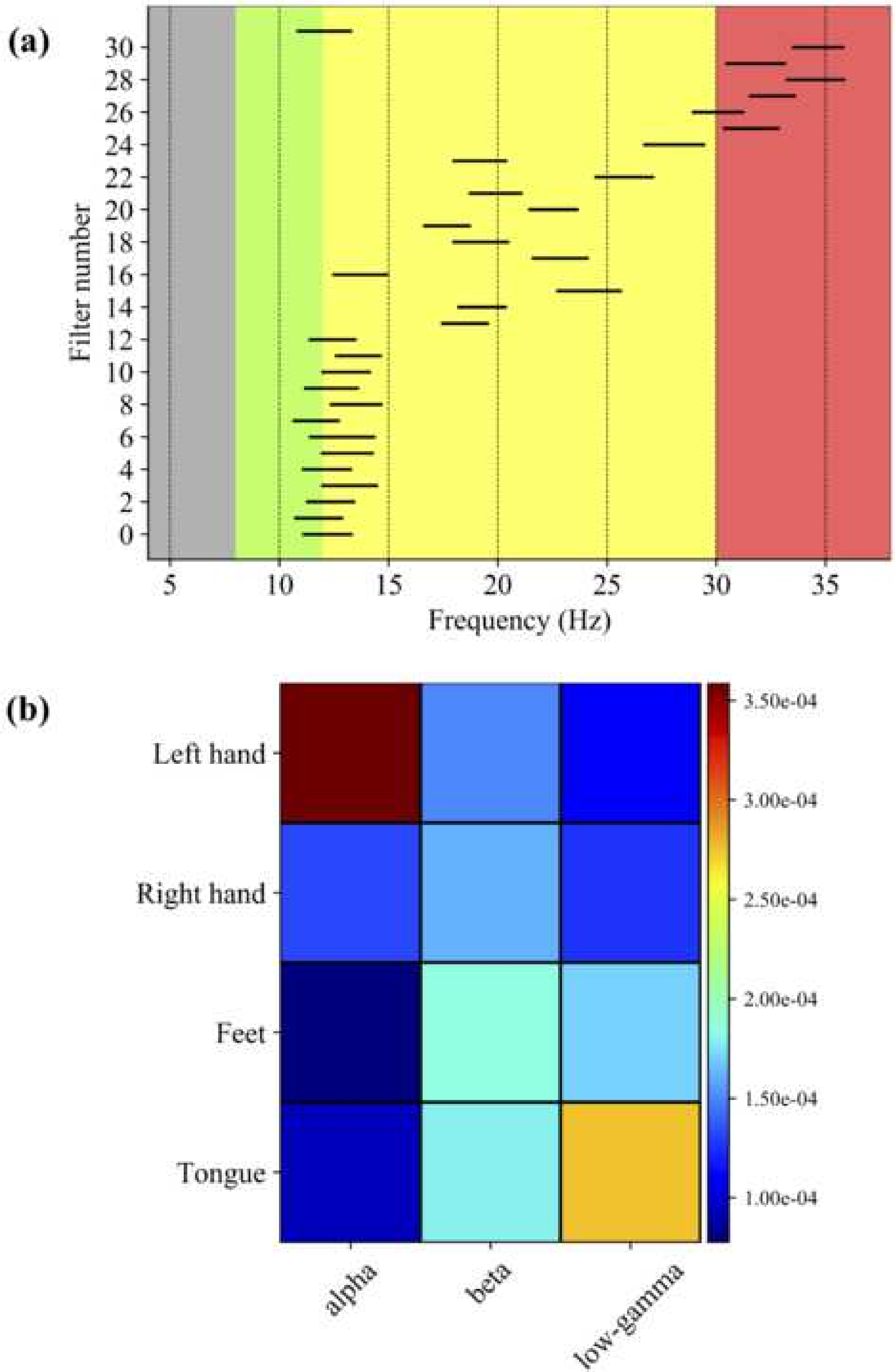
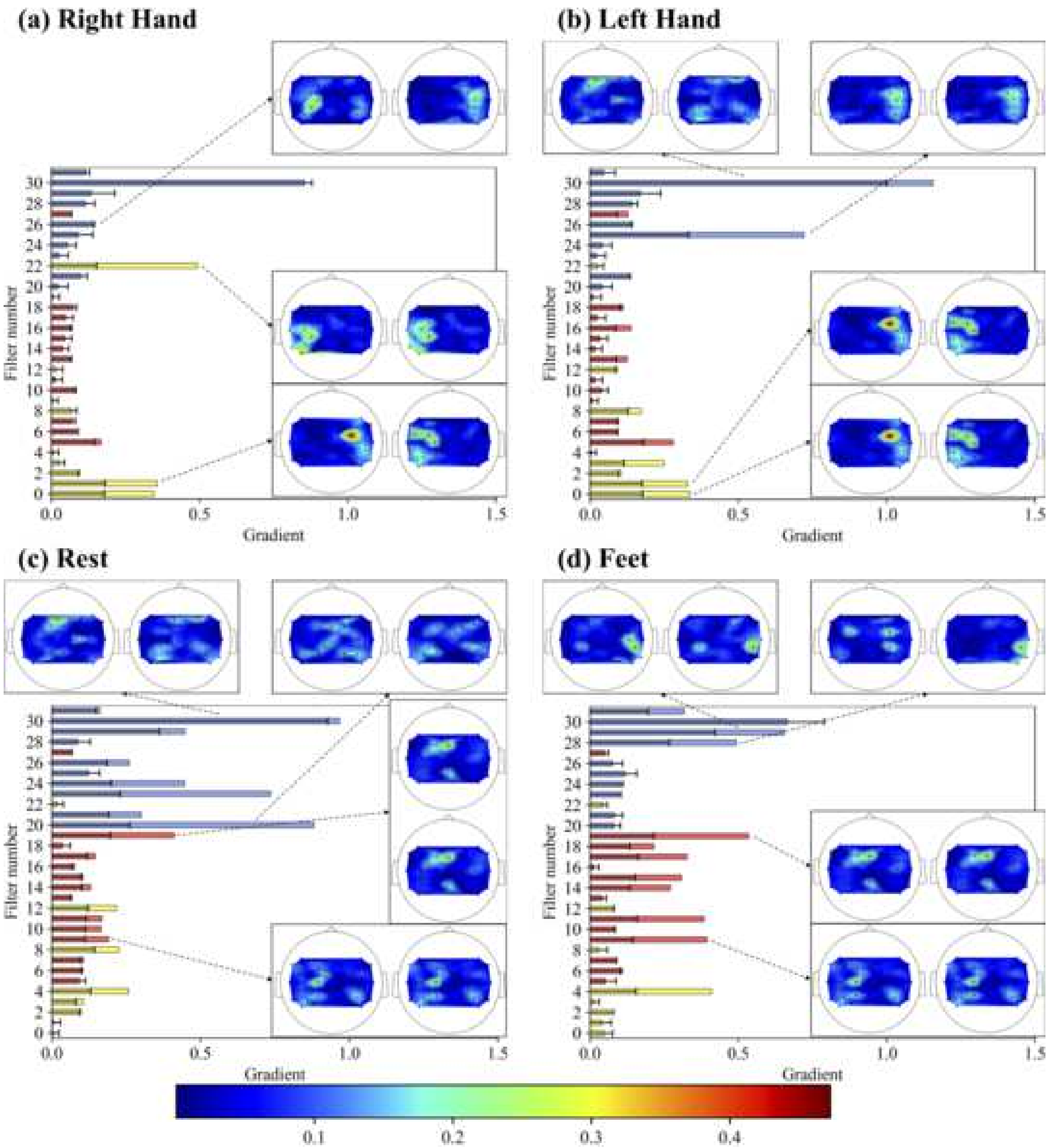

(a)
Filter number
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
10
20
30
40
50
60
70
80
90
100
110
120
Frequency (Hz)
(b)
Right hand
Left hand
Rest
Feet
beta
low-gamma
high-gamma
2.25e-04
2.00e-04
1.75e-04
1.50e-04
1.25e-04
1.00e-04
7.50e-05

Figure 6



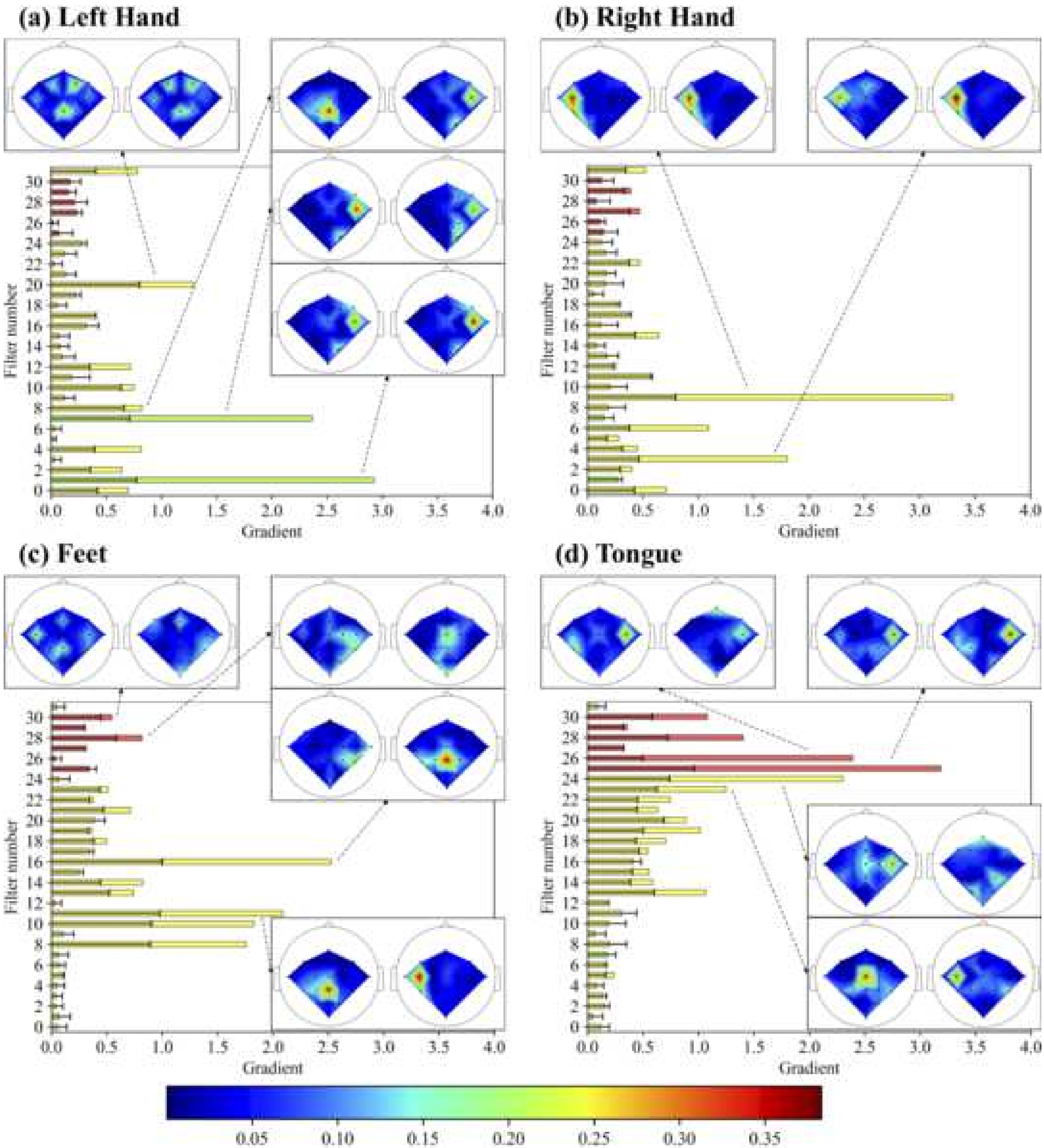
(a)
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
0
Filter number
5
10
15
20
25
30
35
Frequency (Hz)
(b)
Left hand
Right hand
Feet
Tongue
alpha
beta
low-gamma
3.50e-04
3.00e-04
2.50e-04
2.00e-04
1.50e-04
1.00e-04

Figure 7



(a) Right Hand
(b) Left Hand
(c) Rest
(d) Feet
Filter number
Gradient
0.0
0.5
1.0
1.5
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
0.1
0.2
0.3
0.4

Figure 8



(a) Left Hand
(b) Right Hand
(c) Feet
(d) Tongue
Filter number
Gradient
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0
0.05 0.10 0.15 0.20 0.25 0.30 0.35

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: