Computers in Biology and Medicine 165 (2023) 107323

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Decoding movement kinematics from EEG using an interpretable convolutional neural network

Davide Borra [a,*], Valeria Mondini [b], Elisa Magosso [a,c,d,1], Gernot R. Müller-Putz [b,e,1]

[a] Department of Electrical, Electronic and Information Engineering "Guglielmo Marconi" (DEI), University of Bologna, Cesena Campus, Cesena, Italy
[b] Institute of Neural Engineering, Graz University of Technology, Graz, Austria
[c] Alma Mater Research Institute for Human-Centered Artificial Intelligence, University of Bologna, Bologna, Italy
[d] Interdepartmental Center for Industrial Research on Health Sciences & Technologies, University of Bologna, Bologna, Italy
[e] BioTechMed, Graz, Austria

ARTICLE INFO

Keywords:
Electroencephalography (EEG)
Motor trajectory decoding
Deep neural networks
Interpretable neural networks
Brain-computer interface (BCI)

ABSTRACT

Continuous decoding of hand kinematics has been recently explored for the intuitive control of electroencephalography (EEG)-based Brain-Computer Interfaces (BCIs). Deep neural networks (DNNs) are emerging as powerful decoders, for their ability to automatically learn features from lightly pre-processed signals. However, DNNs for kinematics decoding lack in the interpretability of the learned features and are only used to realize within-subject decoders without testing other training approaches potentially beneficial for reducing calibration time, such as transfer learning. Here, we aim to overcome these limitations by using an interpretable convolutional neural network (ICNN) to decode 2-D hand kinematics (position and velocity) from EEG in a pursuit tracking task performed by 13 participants. The ICNN is trained using both within-subject and cross-subject strategies, and also testing the feasibility of transferring the knowledge learned on other subjects on a new one. Moreover, the network eases the interpretation of learned spectral and spatial EEG features. Our ICNN outperformed most of the other state-of-the-art decoders, showing the best trade-off between performance, size, and training time. Furthermore, transfer learning improved kinematics prediction in the low data regime. The network attributed the highest relevance for decoding to the delta-band across all subjects, and to higher frequencies (alpha, beta, low-gamma) for a cluster of them; contralateral central and parieto-occipital sites were the most relevant, reflecting the involvement of sensorimotor, visual and visuo-motor processing. The approach improved the quality of kinematics prediction from the EEG, at the same time allowing interpretation of the most relevant spectral and spatial features.

1. Introduction

Recent efforts in Brain-Computer Interfaces (BCIs) research have been focusing on the fine reconstruction of voluntary movement trajectories from brain signals, so to be able to control an actuator (e.g., a robotic arm or a neuroprosthesis) in a more intuitive and natural way [1–5]. To do so, decoders should be able to continuously and accurately predict executed or imagined trajectories [6,7] from brain signals, instead of discriminating between few discrete executed or imagined motor states (e.g., different moved body parts or movement types) [8,9]. Movement trajectories have been decoded from invasively recorded signals, like electrocorticographic [10,11], intracortical [5,12] or single-neuron recordings [13], and from non-invasive recordings as well, like magnetoencephalographic [14–18] or electroencephalographic (EEG) recordings [18–25], even for closed-loop control [26,27]. Remarkably, despite their low signal-to-noise ratio, non-invasive recordings proved to encode information about kinematics. Bradberry et al. [19] first proved the feasibility of reconstructing velocities from the EEG and paved the way for EEG-based prediction of kinematic variables. This is testified by many successful machine learning applications developed over the past years for predicting positions [18,20,23,25–27] and velocities [18–23,26,27] from the EEG (hereafter termed as 'EEG trajectory decoding').

The state-of-the-art (SOA) widely adopts traditional machine learning approaches for trajectory decoding, including linear models [28], such as partial least squares (PLS) regression [6,20,26], or the

[*] Corresponding author.
E-mail address: davide.borra2@unibo.it (D. Borra).
[1] Authors with shared senior authorship.

https://doi.org/10.1016/j.compbiomed.2023.107323
Received 19 May 2023; Received in revised form 28 July 2023; Accepted 7 August 2023
Available online 8 August 2023

combination of PLS with Kalman filters (KF), i.e., PLS + KF, to integrate the information of different decoding models [26]. However, when the only information of directional parameters (e.g., positions and velocities) was used for the decoding, an amplitude mismatch between the decoded and the actual trajectories could be observed [20,26], which suggested a role of non-directional parameters (e.g., distance and speed) in reconstructing the amplitude [18,27]. To integrate both types of information, a new PLS + Unscented Kalman Filter (PLS + UKF) non-linear decoder was introduced [10,18,27]. The PLS + UKF model was successful in alleviating the amplitude mismatch, and used both offline [18] and online [27] to decode the movement from the low-frequency EEG. Two main shortcomings can be identified affecting traditional EEG trajectory decoding approaches. First, EEG trajectory decoding was mainly performed exploiting low-frequency EEG (<3 Hz). This was motivated since low-frequency EEG activity in the range of delta-band was largely found to encode kinematic information not only during executed movements [18–21,23–27], but also during imagined [6,22,29] and observed [20,29] movements. However, recent research suggests that higher frequency components (e.g., beta and low-gamma) may carry additional movement-related information; therefore, using only low-frequency EEG may limit the quality of trajectory reconstruction. Second, due to the variability of EEG between subjects, decoders have been mainly trained each time anew, without exploring the possibility of transfer the knowledge from other subjects to a new one (subject-to-subject transfer learning), a practice useful to reduce BCI calibration times.

Over the last years, deep neural networks (DNNs) – mainly convolutional neural networks (CNNs) [30] but also recurrent neural networks (RNNs), e.g., long short-term memories (LSTMs) and gated recurrent units (GRUs) [31] – were applied to EEG decoding in several domains [32,33], such as classification of emotions, event-related potentials (e.g., P300) and executed or imagined movements (typically decoding the body part involved in the movement, e.g., hand vs. feet). The main advantage of DNNs over traditional machine learning decoders consist in the ability of automatically learning the most relevant features from raw or slightly pre-processed signals, without selecting features of the input EEG based on a priori knowledge [32], thus exploiting almost the entire information contained in data (e.g., without limiting the EEG input to a narrow frequency band). By virtue of the automatic feature learning, DNNs were proven to outperform traditional machine learning approaches [34–48], e.g., in motor classification, and continuous trajectory decoding problems. In addition, DNNs could be also preferred over traditional machine learning as they might facilitate transferring the knowledge learned from other subjects to a new one (transfer learning) [32], starting from cross-subject decoders. Specifically, cross-subject features learned on previous subjects by a DNN can be fine-tuned to the new user, improving the prediction, especially in case only few examples of the new user are available [38,40,47]. In a BCI, transferring knowledge from previous users would lead to shorter calibration times, which would improve practicality and usability of the interface, and maximize engagement and learning during feedback [38,40,47]].

Following the successful applications of DNNs to solve simpler EEG motor classification tasks, these were recently applied also to predict trajectories from the EEG, by adopting RNNs [46], CNNs [46,49], and combinations between CNNs and RNNs [46,49]. Notably, in these applications, the EEG activity was decoded also including high-frequency components. Nakagome et al. [46] compared traditional approaches (both linear, e.g., L2 regularized linear regressor, and non-linear, e.g., UKF) with RNNs for lower-limb trajectory decoding, including multi-layer LSTM (StackedLSTM), GRU (StackedGRU) and quasi recurrent neural networks (QRNNs) [50]. StackedGRUs and QRNNs proved to be the most accurate decoders, exhibiting similar performance but with StackedGRUs requiring less layers than QRNNs. Furthermore, the authors compared the decoding performance using low-frequency vs. high-frequency EEG (i.e., large bandwidth EEG up to gamma-band), and observed a performance improvement in case of high-frequency EEG especially in RNNs [46], suggesting that DNNs could be able to learn more meaningful features from high-frequency band for EEG trajectory decoding than traditional approaches. On the other hand, Chen et al. [49] tested CNNs previously proposed for EEG motor classification (DeepConvNet [34], ShallowConvNet [34] and EEGNet [35]) but modified for EEG trajectory decoding, and then proposed a novel hybrid convolutional-recurrent network, based on the combination between EEGNet and a sequence of two LSTM layers (named EEGNet-LSTM); the latter resulted more accurate than the considered CNNs.

Despite these first promising results [46,49], the development of DNN-based decoders for EEG trajectory decoding is still in its infancy compared to EEG motor classification [34,35,37,41–45] and two main limitations are currently affecting its advance. First, DNNs were mainly validated realizing within-subject decoders (i.e., decoders trained separately for each subject), without addressing subject-to-subject transfer learning. Notably, this shortcoming afflicts also EEG motor classification; indeed, transfer learning was mainly tested when classifying brain states other than motor ones, e.g., the P300 response [38,40,47]. Thus, the SOA has not entirely explored yet the potentialities of DNNs for reducing calibration times in BCIs either in case of continuous (e.g., regression of hand position) or discrete (e.g., classification of moved body part) motor decoding. Second, the designed DNNs lack in the interpretability of the learned features. The development and application of techniques to increase the interpretability DNNs are receiving growing interest [51]. Indeed, the interpretation of these features can be useful to check the correct learning by verifying that the network did not rely on artifactual but on neurophysiological features, and also to shed light on the EEG features that are most relevant for decoding the variable of interest (e.g., hand position) [37]. Crucially, in the context of CNN-based EEG motor classification, this limitation was overcome by directly incorporating interpretability into the CNN structure, by designing layers of artificial neurons (*interpretable layers*) whose parameters to fit are directly interpretable in a given domain (e.g., frequency domain), thus realizing an interpretable CNN (ICNN) [37,45]. In addition to interpretable layers, *explanation techniques* (ETs) such as saliency maps [52] can be used to explain network decision [37,38,40,47,53,69], by revealing which learned features result most discriminative for a specific output. Therefore, by properly designing an ICNN, the EEG features learned by the network can be easily interpreted in a given domain (spatial, temporal, spectral). Then, among these interpreted EEG features, the most relevant ones for predicting motor behavior (e.g., hand position) can be highlighted, by combining the ICNN with an ET.

In this study, we aim at advancing DNN-based trajectory decoding from EEG by overcoming the previously presented limitations, namely the absence of exploration of transfer learning and absence of feature interpretability, at the same time ensuring high performance. To this aim, we considered Sinc-ShallowNet [37], an ICNN that we previously validated for EEG motor classification, and we modified it to also learning deep temporal features at multiple time scales, a change known to generally improve EEG decoding capabilities in CNNs [38,40,42–44]. Specifically, with the proposed ICNN we aspire to:

i. Improve the performance of EEG trajectory decoding (2-D positions and velocities). To this aim, we benchmarked the proposed ICNN against a wide set of SOA decoders (7 in total).
ii. Explore the possibility of transferring the knowledge across subjects in EEG trajectory decoding. Besides the commonly used within-subject training strategy (using the calibration data entirely), we performed decoding with no calibration (leave-one-subject-out) or little calibration (subject-to-subject transfer learning).
iii. Provide an exemplary illustration of how to interpret the EEG features learned by the ICNN in the frequency and spatial domains and how, by combining the ICNN with saliency maps

(ICNN + ET), the most relevant features for trajectory decoding may be uncovered. This may be a practical exemplification of how the combination ICNN + ET might be used to gain insights into EEG features related to kinematics.

Even though the proposed approach exploits a neural network incorporating interpretable components that inherently reduce the model capacity (i.e., the ability of approximating a wide variety of functions) [37] in favor of an improved interpretability, we expect that an interpretable network can perform on par or outperform other less interpretable SOA decoders even when applied to a highly challenging problem such as EEG trajectory decoding, encouraged by recent results obtained in case of EEG motor classification [37,45]. Furthermore, as we recently obtained in other EEG non-motor decoding problems [38,47], we also expect that transfer learning helps reducing the number of calibration trials needed to train decoders for trajectory decoding, thus, reducing calibration time in a hypothetical BCI scenario. Lastly, we expect that the analysis of the most relevant features learned by the network in the frequency and in the spatial domains, help to support and/or to advance the current knowledge about the neural substrate of hand kinematics.

2. Methods

2.1. Data description and pre-processing

In this study, we used the data of the Graz BCI group recorded in Refs. [26,27], consisting of EEG signals of 13 healthy subjects (aged of 27 ± 4 years, mean ± standard deviation, 7 females, 1 left-handed) while they were performing a pursuit tracking task with their right hand/arm. The experimental procedure conformed to the Declaration of Helsinki and was approved by the ethics committee of the Medical University of Graz (protocol number 29–058 ex 16/17).

During the experiment, the subjects were asked to track a moving object displayed on a screen using a robotic arm; the latter was controlled by a mixture of hand kinematics and trajectories decoded from the EEG (see Fig. 1a). The experiment was composed by a calibration phase and a following online feedback phase, both collected in runs (5 calibration runs, 6 feedback runs), as schematized in Fig. 1b. Each run was composed of 10 trials in which the object was tracked for 23 s. Crucially, the trajectories of the object were generated offline to ensure uncorrelated positions and velocities across and within horizontal (x) and vertical (y) coordinates. Two additional special runs, called ‘eyeruns’, were performed to collect rest data, saccadic eye movements, and blinks, to fit a regression model to attenuate eye movement artifacts [54]. During the calibration phase, the robot was entirely controlled by the hand kinematics (see Fig. 1b). Afterwards, a linear [26] or non-linear [27] decoder could be fitted so to predict the kinematics from the EEG. During the online feedback phase, the subject could then gradually receive feedback on the decoded movements, as the control signal of the robot was progressively switched from hand kinematics to EEG-based decoded trajectories (from 33%, to 66%, to finally 100% of EEG control, see Fig. 1b). In this study, we used the signals collected during the calibration phase as training set for neural decoders, and the signals collected during the online feedback phase as test set, as performed in Refs. [26,27]. Therefore, the training and test sets were composed by 50 and 60 trials (each one lasting 23 s), respectively (see Sections 2.2 and 2.4 for further details about training set and test set split).

During the recordings, the 2-D positions and velocities of the right hand were recorded using an optical hand tracker, together with the EEG signals from 64 [26] or 60 [27] electrodes placed on the scalp according to the 10-10 system. A common subset of 53 electrodes between the two studies was identified and used here to perform trajectory decoding. Reference and ground electrodes were placed at the right mastoid and AFz, respectively. Additional electrodes were placed around the eyes to record the electrooculogram (EOG). Both the EEG and EOG signals were recorded at 500 Hz.

The 2-D hand trajectories were low-pass filtered using a cutoff frequency of 4 Hz as in Refs. [26,27] and downsampled at 100 Hz. The EEG processing pipeline was chosen as close as possible to the one used in the original studies [26,27] which the current data set comes from. The only change here was the use of a higher cutoff frequency of the low-pass filter, to leave the decoder free to explore a wider frequency spectrum than in Refs. [26,27]. The EEG preprocessing steps are:

i. Zero-phase, high-pass filtering (1st order Butterworth) with a cutoff frequency of 0.18 Hz.
ii. Zero-phase, low-pass filtering (4th order Butterworth) with a cutoff frequency of 40 Hz, instead of 1.5 Hz as in Refs. [26,27].
iii. Notch filtering at 50 Hz and 100 Hz.
iv. Downsampling at 100 Hz, instead of 20 Hz as in Refs. [26,27].
v. Bad channel marking via visual inspection, and linear interpolation from the 4 nearest neighboring channels.
vi. Eye artifact correction based on SGEYESUB algorithm [54] (after fitting the algorithm on signals of the eyeruns).
vii. Common average referencing.

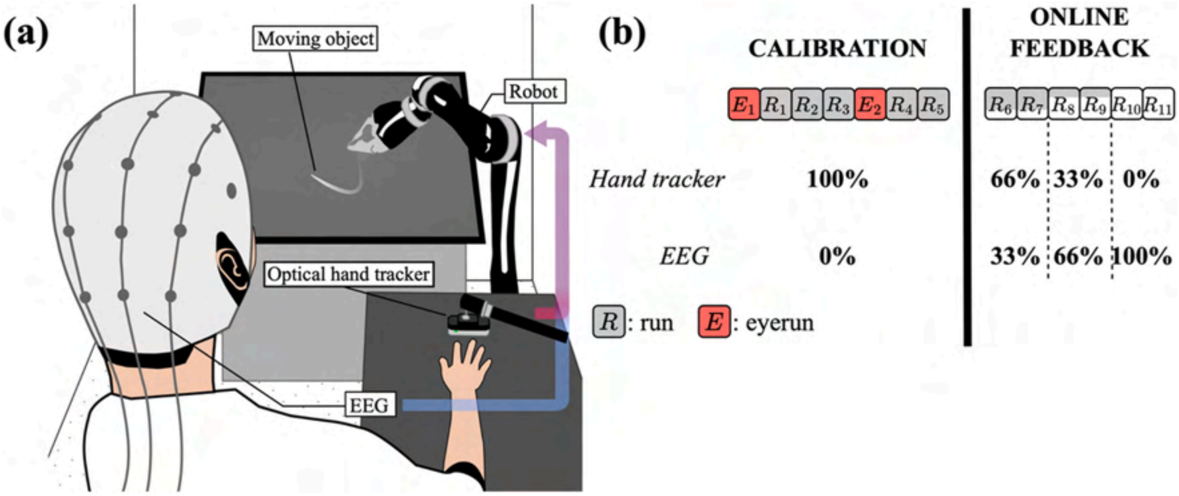

Fig. 1. (a) Schematics of the recording setup. (b) Structure of the BCI paradigm adopted in previous studies [26,27] from which the EEG and trajectory data used here were taken.

viii. Interpolation of the slow drifts/occasional electrode pops via HEAR algorithm [41] (fitting the algorithm on eye-corrected signals of the eyeruns).

The pre-processed neural signals and 2-D hand trajectories were then subjected to the framework depicted in Fig. 2a. First, the ICNN was used to decode hand trajectories from the EEG on sliding windows (blue lines in Fig. 2a) exploiting different training strategies. Then, the most relevant spectral and spatial features of the input EEG were revealed using the knowledge learned by the ICNN, by combining the ICNN using an ET (red lines in Fig. 2a). In the following sections, all steps of this framework will be detailed.

2.2. *Trajectory decoding on EEG sliding windows*

In this section, we formalize the problem of decoding hand trajectory from the EEG using sliding windows. In this study, for each subject, the EEG signals and the set of variables to be predicted, were continuously recorded in several trials. The dataset $D^{(s)}$ associated to the s-th subject can thus be expressed as:

$$D^{(s)} = \left\{ \left(X_0^{(s)}, Y_0^{(s)} \right), \ldots, \left(X_i^{(s)}, Y_i^{(s)} \right), \ldots, \left(X_{M^{(s)}-1}^{(s)}, Y_{M^{(s)}-1}^{(s)} \right) \right\}, \tag{1}$$

where $X_i^{(s)} \in \mathbb{R}^{C \times T}$ ($0 \leq i \leq M^{(s)} - 1$, $M^{(s)}$ denoting the number of trials for the subject s) contains the pre-processed EEG signals of the i-th trial recorded from the C electrode sites and consisting of T time samples, while $Y_i^{(s)} \in \mathbb{R}^{K \times T}$ contains the K pre-processed time series to be predicted organized by rows, recorded for T time samples. The dataset $D^{(s)}$ was divided into a training set used to optimize the set of trainable parameters (denoted by θ), and a test set used to test the algorithm on unseen examples. A separate validation set was extracted from the training set to define the stop criterion of the optimization. See Section 2.4 for additional details on the training/test set split.

In our trajectory decoding problem, the variables in $Y_i^{(s)}$ to be predicted correspond to the 2-D position (p_x, p_y) and/or 2-D velocity components (v_x, v_y) of the hand, resulting in a continuous decoding of kinematics variables from single-trial EEG. To perform such decoding, the decoder predicts the kinematic variables at each time sample by using chunks of the EEG signals, with each chunk consisting of T_z time samples. By indicating with T_s the stride used to sample these chunks, we can write:

$$\begin{cases} Z_{i,j}^{(s)} = X_i^{(s)}[\,:, jT_s : jT_s + T_z - 1] \in \mathbb{R}^{C \times T_z} \\ y_{i,j}^{(s)} = Y_i^{(s)}[jT_s + T_z - 1] \in \mathbb{R}^K \end{cases}, 0 \leq j \leq L - 1, \tag{2}$$

where L denotes the number of chunks that could be extracted using T_z and T_s as chunk size and stride, respectively, i.e., $L = (T - T_z)/\ T_s + 1$. These parameters were set as $T_z = 100$ (i.e., chunks of 1 s), and $T_s = 10$ (i.e., stride of 0.1 s) for EEG trials belonging to the training set, while $T_s = 1$ (i.e., stride of 0.01 s) for the trials belonging to the test set. T_s was set smaller during testing to provide an inference at the same sampling rate as the kinematics (i.e., 100 Hz), and was set larger during training to keep limited the training time. Specifically, for each trial, the number of training EEG chunks resulted $L = (2300 - 100)/10 + 1 = 221$.

The objective decoding problem can be formalized as the optimization of the parametrized regressor f implemented with a CNN, $f(Z_{i,j}^{(s)}; \theta) : \mathbb{R}^{C \times T_z} \to \mathbb{R}^K$, with its parameters contained in θ and that must be learned from the training set of examples to assign the correct label to the unseen examples of the test set. $Z_{i,j}^{(s)}$ represents the CNN input, containing a chunk of the EEG signals organized in a 2-D array of shape (C, T_z), with electrodes along the height and time steps along the width. $y_{i,j}^{(s)}$ represents the CNN output, containing the K values of the variables to be predicted, organized in a 1-D array of shape (K). Considering the dataset used in this study, $C = 53$, and $K = 4$ (corresponding to p_x, p_y, v_x, v_y).

2.3. *The interpretable CNN for trajectory decoding: MS-Sinc-ShallowNet*

The ICNN used in this study, named Multi Scale (MS)-Sinc-ShallowNet, is a modified version of an ICNN (Sinc-ShallowNet [37]) that we recently proposed for motor classification (classification of executed and imagined motor states). Sinc-ShallowNet is composed of several stacked layers grouped into two blocks: an interpretable spectral and spatial (ISS) feature extractor followed by a classification block (with a single fully-connected layer), performing classification. The ISS block is designed to increase the interpretability of the learned parameters in the frequency and spatial domains, at the same time keeping limited the model size (i.e., the number of trainable parameters).

Here, MS-Sinc-ShallowNet exploits the ISS block of Sinc-ShallowNet [37], and places it on top of a light multi-scale (MS) temporal feature extractor, which processes the output of the ISS block in the temporal domain at multiple scales. Lastly, a regressor block finalizes trajectory decoding. Therefore, in MS-Sinc-ShallowNet three main blocks are used. It is worth noticing that learning deep temporal features at multiple scales is a strategy that proved to increase the performances in EEG decoding of cognitive [38,40] and motor [42–44] states, compared to learning temporal features at a single scale. Fig. 2b reports a high-level scheme of the network, while Table 1 reports detailed information about the hyper-parameters, number of trainable parameters and output shape of each network layer. The three blocks defining MS-Sinc-ShallowNet are described in the following.

2.3.1. *Interpretable spectral and spatial (ISS) feature extractor*

The first block is based on the first layers of Sinc-ShallowNet [37] and is devoted to separately learn spectral and spatial features from the input EEG chunk in an easy interpretable way. The very first layer of ISS block is a temporal sinc-convolutional layer [37,55,56], learning $K_0^{ISS} = 16$ filters with filter size $F_0^{ISS} = (1, 51)$, unitary stride and zero-padding to preserve the number of input temporal samples. This temporal convolutional layer is devoted to filter each electrode signal in time. Thanks to the use of a sinc-convolutional layer to perform such processing step instead of a conventional convolutional layer, each convolutional filter was forced to describe a band-pass filter in the temporal domain as we adopted previously in Sinc-ShallowNet [37] (see Appendix A for a detailed description about the temporal sinc-convolutional layer). This way, for the l-th filter only the cutoff frequencies ($\{f_{0,l}, f_{1,l}\}$) of that band-pass filter are learned, reducing the number of trainable parameters (from 51 to only 2 per filter) and increasing the interpretability of the learned features, as these are directly related to a specific spectral content.

$$\theta_{spect,l} = \{f_{0,l}, f_{1,l}\} \in \theta, 0 \leq l \leq K_0^{ISS} - 1. \tag{3}$$

Thus, the output of this first layer consists of stacked feature maps containing band-pass filtered versions of the input EEG chunk within specific frequency ranges that were explicitly learned during training.

Downstream the temporal sinc-convolutional layer, a spatial depthwise convolutional layer is used: for each band-pass filtered map, $D_1^{ISS} = 2$ spatial filters are learned, having size $(C, 1)$ and unitary stride, i.e., D_1^{ISS} spatial combinations of electrodes are learned for each passband filtered map (D_1^{ISS} indicates the depth multiplier). Therefore, a total number of $K_1^{ISS} = K_0^{ISS} \bullet D_1^{ISS} = 32$ spatial filters are learned and constrained to have a norm upper bounded by $c = 1$ (kernel max-norm constraint). This type of convolution does not exploit dense connections across feature maps as in traditional convolutional layers, thus, reducing the number of trainable parameters. In addition, the combination of temporal sinc-convolution with spatial depthwise convolution provides an interpretable spectral-spatial feature learning, as each group of D_1^{ISS} spatial filters is strictly tied to a specific band-pass filter, i.e., to a specific frequency range:

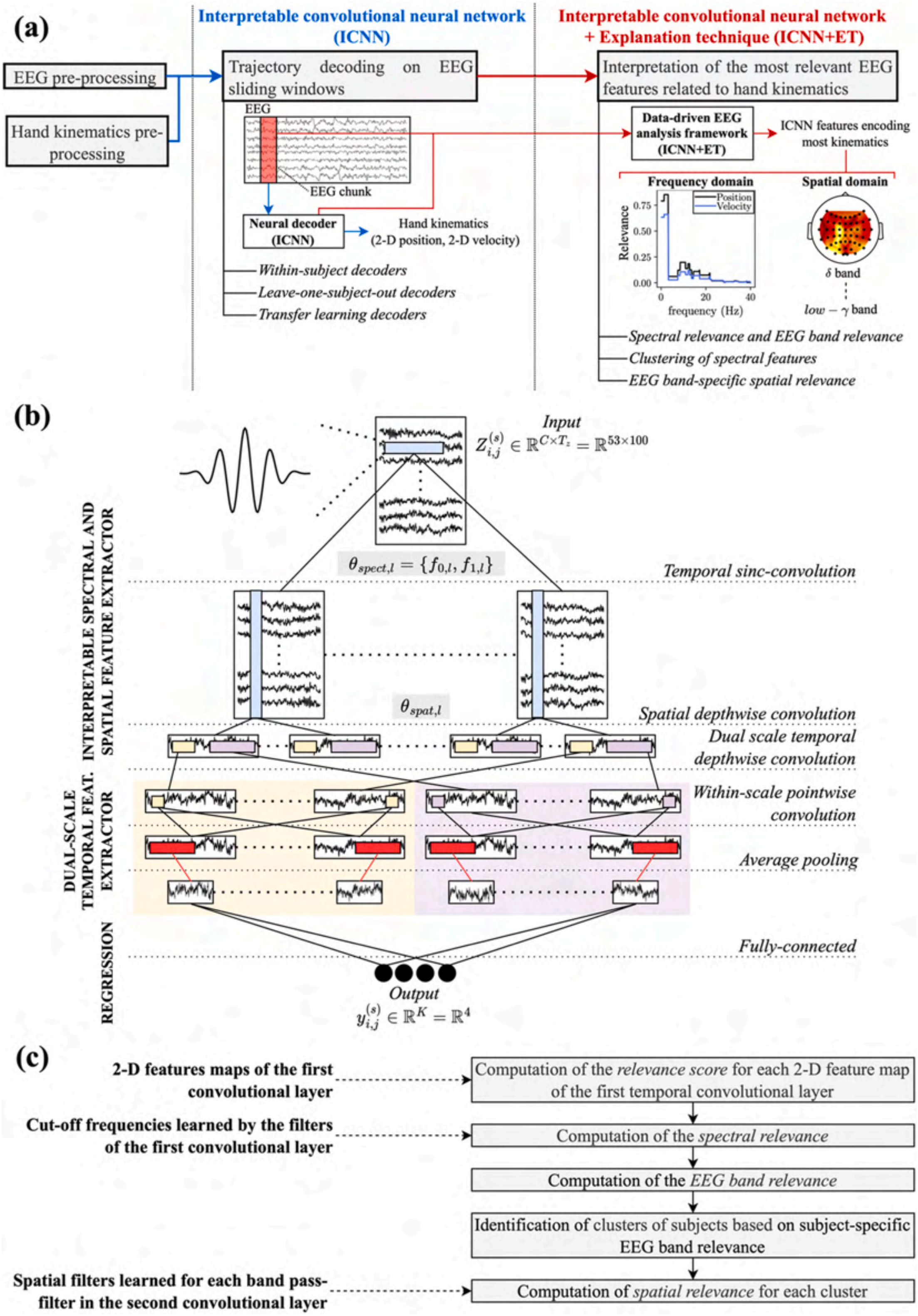
(a)
EEG pre-processing
Hand kinematics pre-processing
Interpretable convolutional neural network (ICNN)
Trajectory decoding on EEG sliding windows
EEG
EEG chunk
Neural decoder (ICNN)
Hand kinematics (2-D position, 2-D velocity)
Within-subject decoders
Leave-one-subject-out decoders
Transfer learning decoders
Interpretable convolutional neural network + Explanation technique (ICNN+ET)
Interpretation of the most relevant EEG features related to hand kinematics
Data-driven EEG analysis framework (ICNN+ET)
ICNN features encoding most kinematics
Frequency domain
Spatial domain
Position
Velocity
Relevance
0.75
0.50
0.25
0.00
0
20
40
frequency (Hz)
δ band
low − γ band
Spectral relevance and EEG band relevance
Clustering of spectral features
EEG band-specific spatial relevance
(b)
Input
$Z_{i,j}^{(s)} \in \mathbb{R}^{C \times T_z} = \mathbb{R}^{53 \times 100}$
$\theta_{spect,l} = \{f_{0,l}, f_{1,l}\}$
Temporal sinc-convolution
$\theta_{spat,l}$
Spatial depthwise convolution
Dual scale temporal depthwise convolution
Within-scale pointwise convolution
Average pooling
Fully-connected
Output
$y_{i,j}^{(s)} \in \mathbb{R}^{K} = \mathbb{R}^{4}$
INTERPRETABLE SPECTRAL AND SPATIAL FEATURE EXTRACTOR
DUAL-SCALE TEMPORAL FEAT. EXTRACTOR
REGRESSION
(c)
2-D features maps of the first convolutional layer
Cut-off frequencies learned by the filters of the first convolutional layer
Spatial filters learned for each band pass-filter in the second convolutional layer
Computation of the relevance score for each 2-D feature map of the first temporal convolutional layer
Computation of the spectral relevance
Computation of the EEG band relevance
Identification of clusters of subjects based on subject-specific EEG band relevance
Computation of spatial relevance for each cluster

Fig. 2. (a) High-level scheme of the framework proposed for decoding kinematics from the EEG and for analyzing the most relevant EEG features related to kinematics. A more detailed scheme of the ICNN architecture is reported in panel b, while a flow diagram of the explanation technique is reported in panel (c). (b) MS-Sinc-ShallowNet structure. The main layers are listed on the right, while block names are reported on the left. Boxes represent the output feature maps of each layer, and colored rectangles represent convolutional and pooling (red) kernels. Blue kernels belonged to the interpretable spectral and spatial feature extractor, while yellow and purple kernels to the dual-scale temporal feature extractor, respectively for the short and large time scale. (c) Flow diagram describing the sequence of computations used to interpret the spectral and spatial features learned by the network, identifying the most relevant features for trajectory decoding.

Table 1
MS-Sinc-ShallowNet. Each layer is provided with its name, main hyper-parameters, number of trainable parameters, and output shape. See Sections 2.2 and 2.3 for the meaning of the symbols. In all layers, where not specified, stride (S) and padding (P) were set to $(1,1)$ and $(0,0)$, respectively.

Block	Layer name	Hyper-parameters	No. of trainable parameters	Output shape
	Input	$K_0 = 1$	0	(1,53,100)
ISS	*Sinc-Conv2D*	$K_0{}^{ISS} = 16$, $F_0{}^{\mathrm{ISS}} = (1,51)$, $P_0{}^{\mathrm{ISS}} = (0,25)$	32	(16,53,100)
	Depthwise-Conv2D	$D_1{}^{ISS} = 2$, $K_1{}^{ISS} = 32$, $F_1{}^{ISS} = (53,1)$, $c = 1$	1728	(32,1,100)
	ELU		0	(32,1,100)
	Dropout	$p = 0.5$	0	(32,1,100)
DST-large scale	*Separable (Depth.+Point.)-Conv2D*	$K_1{}^{DST} = 32$, $F_0{}^{DST} = (1,51)$, $D_0{}^{DST} = 1, P_0{}^{DST} = (0,25)$	2720 (1664 + 1056)	(32,1,100)
	ELU		0	(32,1,100)
	AvgPool2D	$F_p{}^{DST} = (1,10)$	0	(32,1,10)
	Dropout	$p = 0.5$	0	(32,1,10)
DST-short scale	*Separable (Depth.+Point.)-Conv2D*	$K_1{}^{DST} = 32$, $F_0{}^{DST} = (1,25)$, $D_0{}^{DST} = 1, P_0{}^{DST} = (0,12)$	1888 (832 + 1056)	(32,1,100)
	ELU		0	(32,1,100)
	AvgPool2D	$F_p{}^{DST} = S_p{}^{DST} = (1,10)$	0	(32,1,10)
	Dropout	$p = 0.5$	0	(32,1,10)
Regressor	*Concatenate*		0	(64,1,10)
	Flatten		0	(640)
	Fully-Connected	$N = K$, $c = 1$	2564	(4)
			8932	

$$\theta_{spat,l} = \left\{\theta_{l0}, \dots\theta_{lk}, \dots, \theta_{lD_1{}^{ISS}-1}\right\} \in \theta, 0 \le l \le K_0{}^{ISS} - 1, \tag{4}$$

indicating with θ_{lk} the k-th spatial filter ($0 \le k \le D_1{}^{ISS} - 1$) tied to the l-th band-pass filter.

This combination enables the design of a fully interpretable spectral-spatial feature extractor, as the parameters of these two first convolutional layers (Eq. (3) and Eq. (4)) directly provides the $K_0{}^{ISS}$ pair of cutoff frequencies of the band-pass filters and the associated $D_1{}^{ISS}$ combinations of electrodes exploited to decode the input EEG trial. Hence, the interpretable features are:

$$\begin{cases} \theta_{ISS} = \left\{\theta_{ISS,0}, \dots, \theta_{ISS,l}, \dots, \theta_{ISS,K_0{}^{ISS}-1}\right\} \\ \qquad \theta_{ISS,l} = \left(\theta_{spect,l}, \theta_{spat,l}\right) \end{cases}. \tag{5}$$

Then, the output of the ISS feature extractor is activated via an Exponential Linear Unit (ELU) non-linearity [57], i.e. $f(x) = x, x > 0$ and $f(x) = exp(x) - 1, x \le 0$, and dropout [58] is applied with dropout rate $p = 0.5$.

2.3.2. Dual-scale temporal (DST) feature extractor

This block is designed to learn temporal features at two time scales from the feature maps provided by the ISS block. Two different and parallel time scales, hereafter called 'large' and 'short' scales, are used, realizing a sub-network consisting of 2 branches. Separable convolutions are used in each branch to reduce the number of trainable parameters [59], thus, realizing a light dual-scale temporal feature extractor, as designed in Ref. [38].

At first, each parallel branch includes a temporal separable convolutional layer, defined by a temporal depthwise convolution followed by a pointwise convolution. The temporal depthwise convolutional layer learns one temporal pattern per input feature map (i.e., depth multiplier set to 1), unitary stride and zero-padding within each branch. However, it differs in the kernel size $F_0{}^{DST}$ across the two branches, to learn features on different time scales. In particular, $F_0{}^{DST} = (1,51)$ and $F_0{}^{DST} = (1,25)$, respectively in the large and short scales, corresponding to learning temporal features within windows of approximately 500 and 250 ms. Then, the pointwise convolutional layer learns $K_1{}^{ISS} = 32$ filters of size $(1,1)$ with unitary stride, within each branch. This layer optimally recombines the feature maps provided by the depthwise convolution within each scale, separately. That is, at each time scale, one temporal pattern is learned, separately, for each feature map provided by the ISS layer (see Fig. 2b), and afterwards the optimal combinations of these activations are learned.

Within each branch, the output provided by the temporal separable convolution is activated via ELU non-linearity, and average pooled with pool size and stride of $F_p{}^{DST} = (1,10)$ to reduce the number of time steps to be processed in the fully-connected layer of the following block (i.e., reducing from T_z to $T_z//10$, indicating with// the floor division operator). Lastly, dropout [58] is applied with dropout rate $p = 0.5$.

2.3.3. Regressor

This block transforms the feature maps at the output of the DST block into the predicted trajectory values. At first, the feature maps provided by the two parallel branches are concatenated together and reshaped as an array with a single dimension. Then, the flattened feature maps are given as input to a fully-connected layer with $N = K = 4$ units, establishing dense connections and constraining the weights of these connections to have a norm upper bounded by $c = 1$ (kernel max-norm constraint).

The total number of trainable parameters was 8932 (see Table 1). Crucially, the main network hyper-parameters such as the learning rate, number of band-pass filters ($K_0{}^{ISS}$), number of spatial filters (based on $D_1{}^{ISS}$), number of parallel time scales, inclusion of batch normalization [60], etc., were automatically searched in a preliminary analysis by performing hyper-parameter tuning via Bayesian optimization [61]. See Supplementary Section 2 for details about hyper-parameter tuning via Bayesian optimization. The ICNN structure previously described (from Section 2.3.1 to 2.3.3 and Table 1) was the optimal structure that was selected more frequently across the Bayesian-optimized models. Lastly, a sensitivity analysis (i.e., ablation test) on the main structural hyper-parameters was conducted (see Supplementary Section 2), by changing one hyper-parameter at a time and evaluating the performance change compared to the adopted Bayesian-optimized architecture, to understand to what extent hyper-parameters affect the performance, as done in Refs. [34,37,38].

2.4. Training strategies and performance evaluation

In this study, we trained the ICNN with 3 different training strategies, by differently defining the training sets or the initialization for the ICNN. However, it is crucial to notice that the definition of the test set was the same across training strategies, enabling a fair comparison between them. The training strategies were within-subject (WS), leave-one-subject-out (LOSO), and transfer learning (TL-WS).

2.4.1. Within-subject (WS)

Each subject-specific decoder was trained using the subject-specific training set consisting of the 50 trials of the calibration phase. The test set was defined as the test set belonging to the subject the ICNN was trained for, consisting of the 60 trials of the online phase. Overall, this strategy was conceived to simulate a use-case scenario where decoders are designed from scratch in a subject-specific manner (i.e., without exploiting any feature learned from other subjects), as generally performed in BCI calibration. Here, networks were randomly initialized before training [62].

2.4.2. Leave-one-subject-out (LOSO)

Each decoder was trained using a cross-subject training set. Specifically, for each subject s, named 'held-out subject', the training sets of all other subjects were aggregated together. Therefore, the training set comprised $12{\cdot}50 = 600$ training trials. Lastly, the test set was defined as the one belonging to the held-out subject (s-th subject), consisting of the 60 trials of the online phase. In this way we trained decoders that are cross-subject, because of the training set, and subject-agnostic, as the test set is relative to the subject held out from the training set (i.e., decoders are cross-subject and calibration free). This strategy was conceived to simulate a practical BCI scenario of calibration-free decoding, i.e., decoding on the new BCI user without performing any calibration on signals recorded from the new subject. Here, networks were randomly initialized before training [62].

2.4.3. Transfer learning on single subjects (TL-WS)

Transfer learning is inspired by the human ability to exploit the knowledge learned in a given domain/task to improve the performance and/or reduce the training time in a different but related domain/task [63]. In this strategy, the knowledge learned on other subjects was transferred to a new subject. As with the WS strategy, subject-specific training sets were used to train subject-specific decoders on each s-th subject, and, thus, the definition of the training and test sets was the same as in the WS strategy (see Section 2.4.1). However, differently from the WS strategy in which the ICNN was randomly initialized, in the TL-WS strategy the ICNN was initialized using the trainable parameters obtained during the LOSO strategy when the s-th subject was held-out. Therefore, the knowledge learned during the LOSO strategy, which incorporated inter-subject variability from all other subjects except the held-out one, was transferred on the held-out subject. That is, in this strategy a different initialization for trainable parameters was used, potentially representing a better initialization point in the parameter space than the random one, and possibly leading to an improvement in performance and/or to a reduction of the training trials needed to achieve high performance. To test the last point, the ICNN was trained with WS and TL-WS strategies both by using all 50 training trials available in the training set, and by using a subset of training trials of increasing size, i.e., 2, 4, 6, 8, 10, 20, 30, 40 trials, by randomly sampling 10 times the trials to be included in the reduced training set. The use of a reduced number of training trials was adopted to simulate practical BCI scenarios in which a new user approaches the BCI, and a limited number of training trials can be recorded. In this context, the TL-WS strategy might enable a short calibration on the new user still providing good performance.

Despite the training sets across all the different training strategies was different, the test set was kept unchanged corresponding to the 60 trials of the online phase for the s-th subject considered, to provide a fair comparison between the conducted experiments. Lastly, in each training strategy, a validation set was selected from the training set, by extracting the first 20% portion from each training trial, i.e., by extracting the first 20% of EEG chunks together with the corresponding kinematic values.

Each network was trained by using the mean squared error between the predicted and true trajectory values as loss function; this loss function was chosen as it is the most adopted one for regression problems [64]. Adaptive moment estimation (Adam) [65] was used as optimizer with learning rate $lr = 1e - 4$, mini-batch size $bs = 64$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for computing the running averages of the gradient and its square, and $\varepsilon = 10^{-8}$ to improve numerical stability. The maximum number of epochs was set to 250 and the training ended when the validation loss did not decrease for 50 consecutive epochs (early stopping). Besides early stopping, MS-Sinc-ShallowNet directly implemented in its structure methodologies devoted to improve generalization, such as dropout [58] and kernel max-norm constraint.

2.4.4. Performance evaluation

Once trained, the ICNN was evaluated on the EEG chunks belonging to the test set, obtaining the predicted trajectories of the 2-D position and velocity during each trial. The predicted trajectories were compared with the recorded kinematics by computing, for each subject, the Pearson's correlation coefficient (r), as was adopted in our previous study [26], and the root mean squared error (RMSE). The performance of our ICNN was compared to the one obtained with 7 other SOA decoders, including the best-performing SOA machine learning algorithm (PLS + UKF [18]), SOA DNNs proposed for EEG motor classification (Sinc-ShallowNet [37], ShallowConvNet [34], DeepConvNet [34]) and adapted here for EEG trajectory decoding, and SOA DNNs specifically proposed for EEG trajectory decoding (StackedLSTM [46], StackedGRU [46], and EEGNet-LSTM [49]). A description of these SOA decoders is reported in Appendix B. The same data preparation used for the presented ICNN (see Sections 2.1 and 2.2) was used for all other tested decoders. Comparisons with decoders were performed using a within-subject strategy, as i) the other strategies (leave-one-subject-out and transfer learning) are unfeasible with the SOA machine learning approach (PLS + UKF) and ii) this is the most adopted strategy to perform BCI calibration, thus it is the most representative training strategy to validate the proposed decoder. SOA DNNs were trained using the same training hyper-parameters (i.e., optimizer, learning rate, batch size, etc.) as those used for our ICNN, to provide a fair comparison. All DNNs were designed, trained and evaluated using the Python library PyTorch (version 1.12.1) [66]. Experiments were conducted on a workstation equipped with an AMD Threadripper 1900X, NVIDIA TITAN V (12 GB) and 48 GB of RAM.

2.5. Interpretation of the most relevant spectral and spatial features related to position and velocity

Interpreting the features learned by MS-Sinc-ShallowNet in the WS strategy ($\theta = \theta^{(s)}$, in WS strategy) can provide insights, for each subject, on the EEG features most relevant to trajectory decoding. The adopted ICNN structure provides interpretable parameters in the array $\theta_{ISS}{}^{(s)}$. As the ICNN processes the input EEG chunks, it filters out motor-unrelated spectral and spatial components while preserving only ones most relevant for the trajectory decoding problem. However, these features may have a different importance for the discrimination, meaning that a band-pass filtering in a peculiar frequency range and a subset of electrodes may be more relevant to predict positions and velocities. Therefore, an explanation technique (ET) was included to highlight the most relevant features ($\theta_{ISS}{}^{(s)}$) for decoding positions and velocities, within each subject. To ease the reading, the main steps implemented to interpret the network are qualitatively summarized in the following points, and reported in the scheme of Fig. 2c. The complete details with quantitative descriptions and motivation for each performed step are reported in Appendix C.

i. Relevance score computation (see Appendix C.1 for a complete description). By using an ET (here saliency maps [52]), for each 2-D feature map of the first temporal convolutional layer (overall 16 feature maps, see Fig. 2b), we computed the importance of each spatio-temporal sample of that feature map for predicting each decoded variable ($\{p_x, p_y, v_x, v_y\}$). Here, we applied saliency maps rather than other newer methods (e.g., layer-wise relevance propagation [67], gradient-weighted class activation mapping [68], shapley additive explanation [76]) since saliency maps have the advantage of keep the explanation process as simple as possible, without the introduction of factors and parameters whose chosen values may influence the obtained representations (e.g. ε rule and its parameters in layer-wise relevance propagation) [38]. Moreover, saliency maps are widely used for explaining networks used for EEG decoding [37,38,40,47,53,69]. The computation of saliency maps resulted in a 2-D relevance map associated to each 2-D feature map. Then, one *relevance score* (scalar value) for each feature map was derived by averaging the relevance map over time samples and electrodes. Since each feature map of the first temporal convolution layer contained the version of the input EEG filtered by one of the learned band-pass filters, this relevance score quantified the importance of the applied band-pass filter for predicting positions and velocities.
ii. Spectral relevance and EEG band relevance computations (see Appendix C.1 for a complete description). By knowing the cut-off frequencies of each filter (contained in $\theta_{spect,l}{}^{(s)}$) and the associated relevance score (computed in point i.), the relevance was expressed for each frequency bin, by weighting each frequency in the bandwidth of each filter with its relevance score and by averaging the result across all the 16 learned filters (*spectral relevance*, or, equivalently, relevance as function of frequency). As no differences were observed across x- and y-components (see Supplementary Section 3 and Appendix C.1), the spectral relevance was also averaged across components, thus resulting in one spectral relevance profile for position and one profile for velocity. Lastly, the relevance of each EEG band (*EEG band relevance*), namely delta (0.18–4 Hz), theta (4–8 Hz), alpha (8–13 Hz), beta (13–30 Hz), and low-gamma (30–40 Hz), was obtained by averaging the spectral relevance across all frequencies within each band. That is, the spectral relevance and EEG band relevance quantified the importance of each frequency and EEG band, respectively, for predicting positions and velocities.
iii. Spectral clustering and spatial relevance computation (see Appendix C.2 for a complete description). By considering the subject-specific EEG band relevance, clustering was performed via HDBSCAN algorithm [82] to reveal groups of subjects with similar features in the frequency domain. Then, separately for each cluster, the following procedure was applied to derive spatial relevance. Since for each band-pass filter a set of 2 spatial filters (contained in $\theta_{spat,l}{}^{(s)}$) were learned in the second convolutional layer, the *spatial relevance* of each EEG band was obtained by averaging the spatial filters (in their absolute value) associated to the band-pass filters that included, inside their bandwidth, the frequencies of the specific EEG band. This way, it was possible to compute the importance of each electrode site for decoding kinematics, specifically for each EEG band.

2.6. Chance level and statistical analyses

The chance level of the proposed ICNN was estimated empirically by evaluating the decoder after it was trained by randomly shuffling the association between input EEG chunks and kinematics (see Eq. (2)), as performed in Ref. [26]. Specifically, within-subject decoders were trained 100 times, randomly shuffling the input-output association in training data each time [26], and then evaluated on the test set (corresponding to the 60 trials of the online phase for the considered subject, see Section 2.4). Then, the upper bound confidence interval of the chance level (with a significance of $\alpha = 0.05$) was estimated as the 95th percentile of the performance metrics (taken in absolute value in case of correlations). Afterwards, the following statistical analyses were conducted.

i. The performance metrics (r and RMSE) scored by all tested decoders (8 decoders in total) were compared by adopting a Friedmann test [70], separately for positions and velocities. Then, as significant differences ($p < 0.001$) were found (see Section 3.1), post-hoc pairwise comparisons were performed comparing the performance metrics scored by the proposed ICNN vs. all other decoders. To do so, for each predicted kinematic variable (i.e., p_x,p_y,v_x,v_y), a pairwise comparison was performed between the performance obtained with MS-Sinc-ShallowNet and each other decoder, both trained using a WS strategy ($2 \cdot 7 \cdot 4 = 56$ total tests). This analysis was applied to evaluate significant differences in performance between the proposed decoder and the SOA decoders.
ii. Correlation scored by MS-Sinc-ShallowNet was compared across the adopted training strategies (WS, LOSO, TL-WS), using a Friedmann test [70], separately for positions and velocities, with WS and TL-WS both trained using all training trials (50 trials). Then, as significant differences ($p < 0.001$) were found (see Section 3.1), post-hoc pairwise comparisons were performed testing for differences between all combinations of training strategies, for each predicted kinematic variable (i.e., p_x,p_y,v_x,v_y). To do so, pairwise comparisons were performed between each combination of training strategies ($3 \cdot 4 = 12$ total tests). This analysis was applied to compare different training strategies, each reflecting a different practical scenario in which the decoder can be used.
iii. Correlation scored by MS-Sinc-ShallowNet was compared between WS and TL-WS strategies for progressively increasing numbers of training trials, to test the potential benefit in transferring the knowledge on a new subject from a network pretrained on other subjects. To this aim, for each predicted variable (i.e., p_x,p_y,v_x,v_y) and each number of training trials (2, 4, 6, 8, 10, 20, 30, 40, 50 training trials, see Section 2.4) a pairwise comparison between MS-Sinc-ShallowNet trained using WS strategy and using TL-WS strategy was performed ($9 \cdot 4 = 36$ total tests).
iv. EEG band relevance (see Section 2.5 and Appendix C) was compared across bands (delta, theta, alpha, beta, low-gamma) using a Friedmann test [70], separately for position and velocity. Then, as significant differences ($p < 0.001$) were found (see Section 3.3), post-hoc pairwise comparisons were performed testing all combinations (10 total tests), separately for position and velocity. This analysis was performed to evaluate which EEG band was the most relevant for decoding position and for decoding velocity.

In the analyses described in previous points, pairwise comparisons were performed using Wilcoxon signed-rank tests [71,72] and using false discovery rate correction at $\alpha = 0.05$ with the Benjamini–Hochberg procedure [73] to correct for multiple tests.

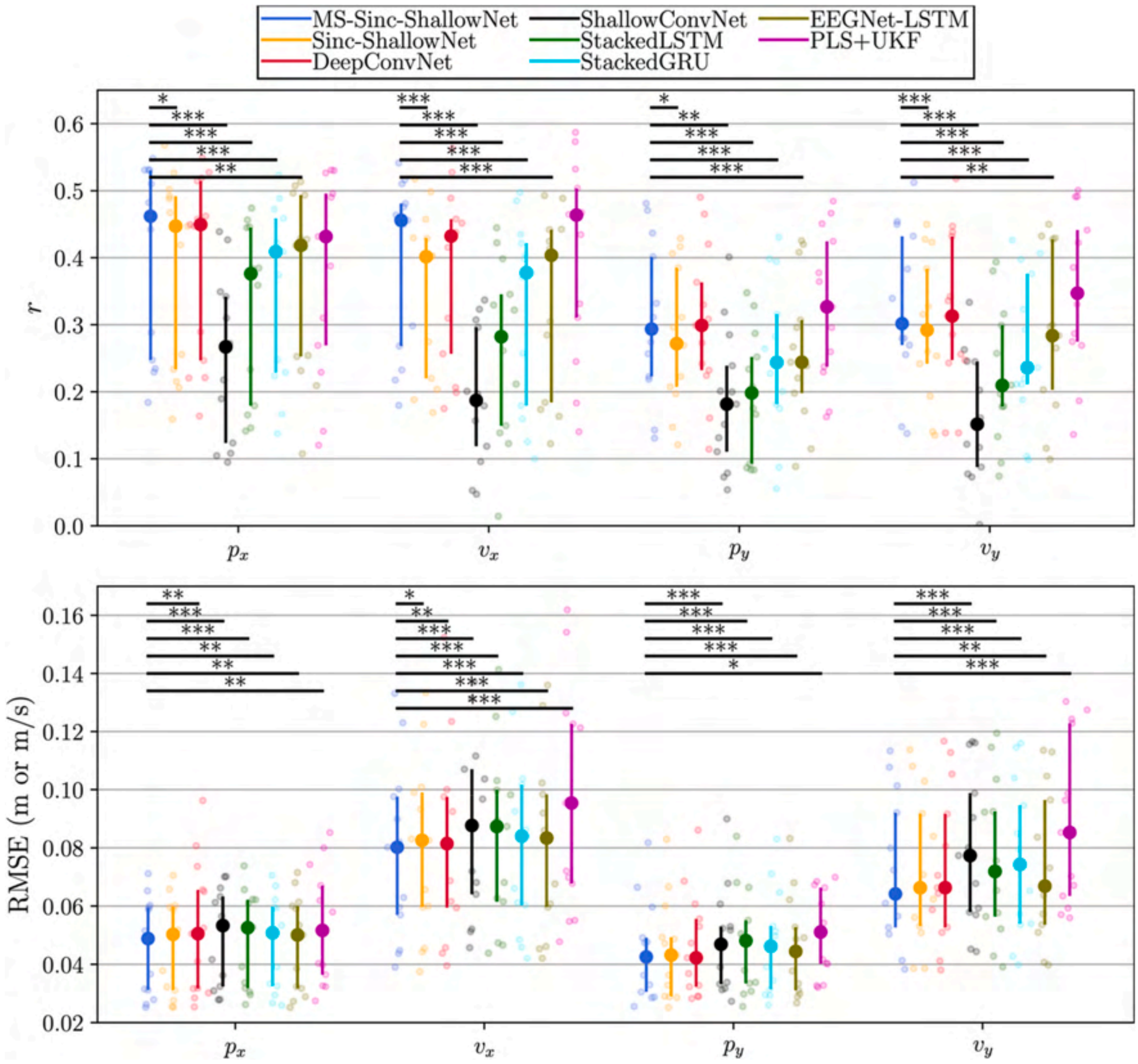

Fig. 3. Performance scored by MS-Sinc-ShallowNet and the tested state-of-the-art decoders for each decoded variable (p_x,p_y,v_x,v_y). Pearson's correlation coefficients (r) are reported in the top-panel, while RMSEs in the bottom-panel. Smaller dots represent the performance metric scored for each subject, while bigger dots represent the median of each distribution and whiskers represent the 25th and 75th percentile. Significant p-values corrected for multiple tests are reported (*p < 0.05, **p < 0.01, ***p < 0.001). Only significant comparisons are indicated.

3. Results

3.1. *Performance of MS-Sinc-ShallowNet and comparison with SOA decoders*

The performance metrics obtained with MS-Sinc-ShallowNet and with other decoders trained using the WS strategy is displayed in Fig. 3, together with the results of the statistical analysis. Significant differences were found both in Pearson's correlation coefficients and RMSEs between decoders for all decoded variables ($p < 0.001$, Friedmann test). When comparing MS-Sinc-ShallowNet to PLS + UKF (the best-performing SOA machine learning algorithm for EEG trajectory decoding), the two decoders scored statistically comparable correlations for all the decoded trajectories, even though PLS + UKF performed slightly better for few variables (e.g., p_y, v_y, $p = 0.07$). However, the proposed ICNN scored significantly lower RMSEs than the PLS + UKF algorithm for all decoded trajectories, reflecting a better amplitude reconstruction, especially for the prediction of the velocity components. MS-Sinc-ShallowNet significantly outperformed ShallowConvNet, StackedLSTM, StackedGRU, and EEGNet-LSTM across all predicted variables both in terms of correlations and of RMSEs. This is an interesting result, as StackedGRU, and EEGNet-LSTM represent two well-performing SOA DNNs specifically released for EEG motor trajectory decoding. Furthermore, when comparing MS-Sinc-ShallowNet to Sinc-ShallowNet (i.e., the previous version of our ICNN, released for EEG motor classification), MS-Sinc-ShallowNet performed significantly better as to the correlation of all predicted variables, while the two ICCNs performed on par regarding RMSEs, except for the velocity in the x-axis, where the proposed ICNN significantly outperforms Sinc-ShallowNet. Lastly, MS-Sinc-ShallowNet performed on par with DeepConvNet for all predicted variables and performance measures, though significantly outperforming it as to RMSEs in the x-axis.

DNNs were also compared in terms of model size (expressed as the number of trainable parameters introduced) and training time (expressed as the time required to complete a training epoch, per training trial). This last measure was provided normalized by the number of training trials presented in each epoch, as different training strategies were generally characterized by a different number of training trials (e.g., 50 trials in WS vs. 600 trials in LOSO, see Section 2.4). In Fig. 4 each decoder is displayed as a dot in the model performance-

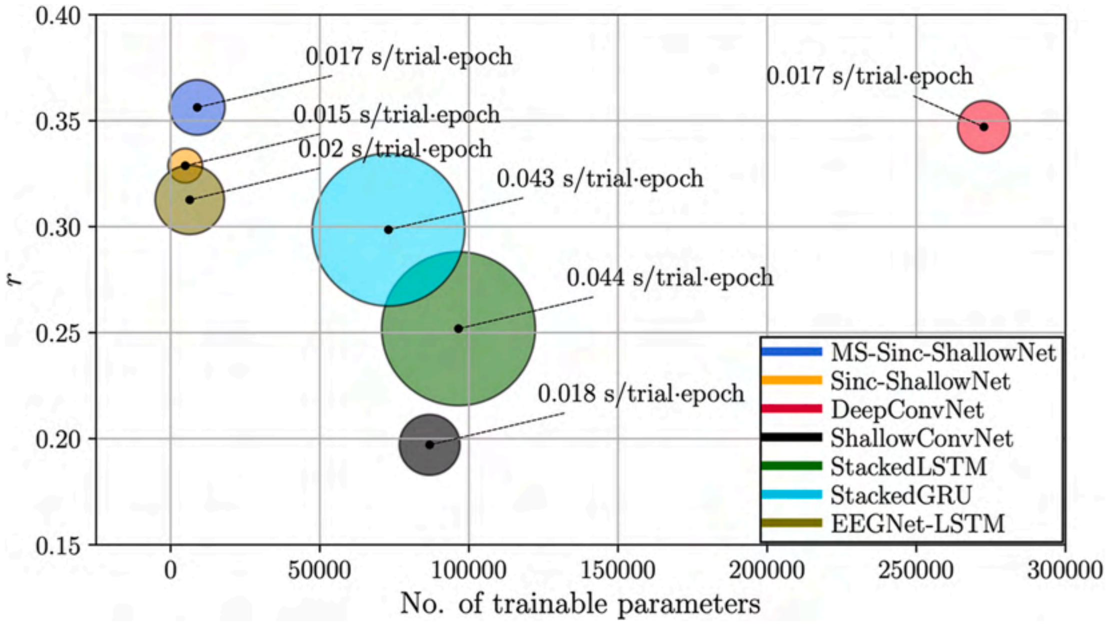

Fig. 4. Comparison among deep neural networks as to performance, size, and training time. Each neural network is represented in the model performance – model size plane as a dot, with the radius dot modulated depending on the model training time. The performance is expressed as the average correlation (r) across the predicted variables (p_x,p_y,v_x,v_y); the model size is quantified by the number of trainable parameters; the training time is measured as the time required to complete a training epoch per training trial.

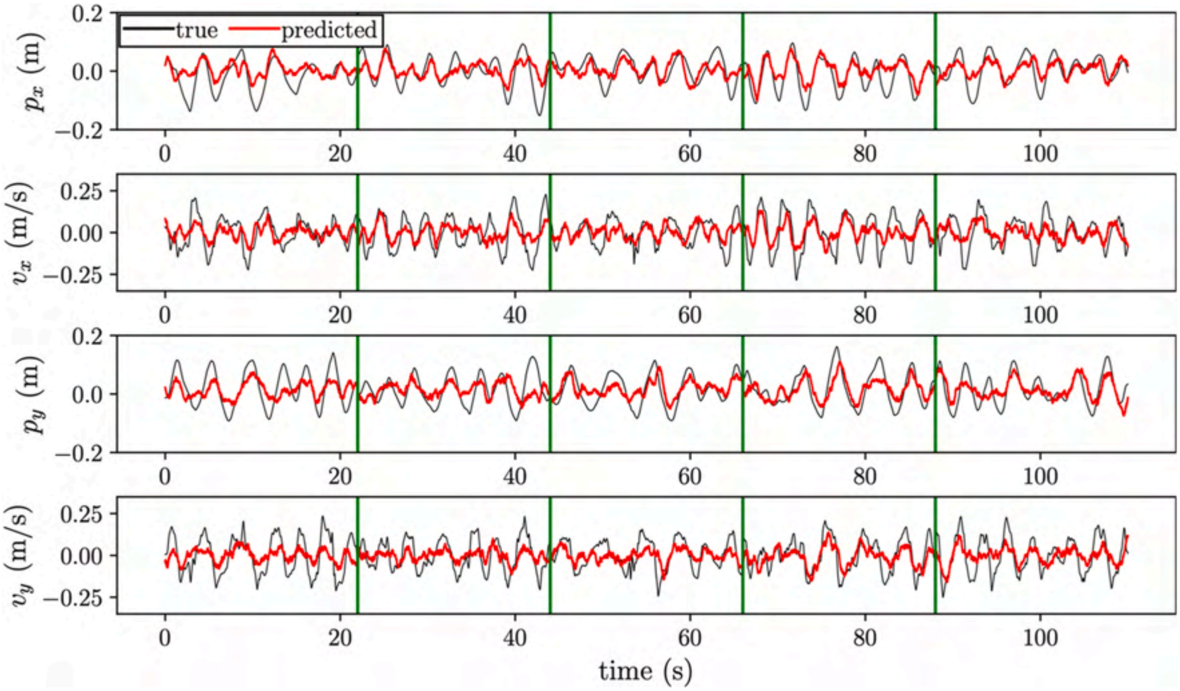

Fig. 5. Example of the trajectories predicted (red) by MS-Sinc-ShallowNet in a representative subject. True trajectories are reported too (black). Here, trials were concatenated together (only 5 trials are displayed for sake of readability); green vertical lines denote the separation between trials.

model size plane (the radius of the dot being proportional to the model training time), where the performance is resumed as the correlation averaged across the predicted variables (p_x, p_y, v_x, v_y). The proposed ICNN not only proved to significantly outperform most of the DNNs both in terms of correlations and RMSEs, but also showed a good compromise between model performance, size, and training time (especially when compared to DeepConvNet).

In addition to the previous performance evaluations, Fig. 5 reports also the predicted trajectories alongside with the true trajectories for a representative subject, as obtained with the proposed ICNN decoder, to provide a qualitative representation of the prediction of the proposed model.

3.2. *Performance when transferring the knowledge from other subjects*

The decoding performance of MS-Sinc-ShallowNet was further investigated by considering additional training strategies, that is the LOSO and TL-WS strategies. In Fig. 6 the performance metrics obtained

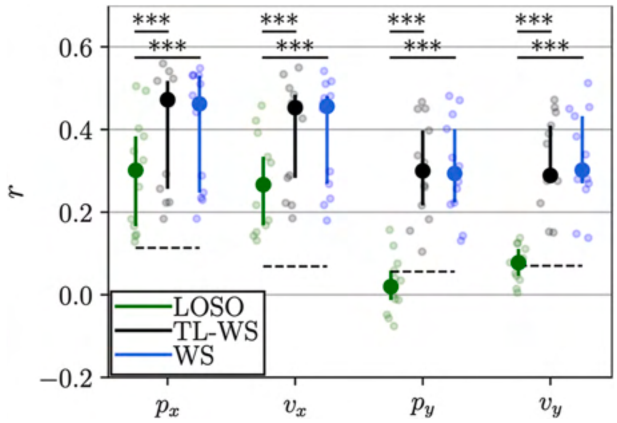

Fig. 6. Performance of MS-Sinc-ShallowNet with different training strategies. Pearson's correlation coefficient (r) is reported for each decoded variable (p_x,p_y, v_x, v_y) in case of the decoder trained with the leave-one-subject-out strategy (LOSO, green), transfer learning strategy (TL-WS, black) and within-subject strategy (WS, blue). Results of WS strategy are the same as in Fig. 3. Note that here the results of WS and TL-WS refer to training using the entire training set (all available 50 training trials) for each subject. Horizontal dashed lines denote the chance level. Significant p-values corrected for multiple tests are reported (*p $<$ 0.05, **p $<$ 0.01, ***p $<$ 0.001). Only significant comparisons are indicated.

with MS-Sinc-ShallowNet trained in WS, TL-WS and LOSO strategy are reported. Note that here the results of WS and TL-WS refer to training with all available 50 training trials for each subject. WS and TL-WS decoders performed significantly above the chance level for all subjects; conversely, LOSO decoders were significantly above chance for all subjects for position and velocity in the x-axis, and only for 4 subjects in the y-axis. This was expected, due to the high inter-subject EEG variability characterizing training distributions in the LOSO strategy. Moreover, significant differences in the performance metrics were found across training strategies for all decoded variables, with significantly lower correlations in LOSO compared to WS and compared to TL-WS strategies and no significant differences between WS and TL-WS strategies. However, despite their lower performance, LOSO models are useful to enable other training strategies such as TL-WS, where the knowledge learned from other subjects is used as initial point for training the network on a new subject and may lead to better performance than training from scratch (as in WS). The advantage of TL-WS over WS was not observed when using the entire training set (i.e., all 50 training trials), as denoted in Fig. 6 by the similar performance between WS and TL-WS, but it may emerge when less training trials are used, as presented in the following.

Fig. 7 reports the performance of the proposed ICNN while simulating scenarios with reduced training sets. Here, the performance is displayed not only using the entire training set (i.e., condition corresponding to 50 trials in the figure, same as the one reported in Fig. 6), but also using reduced training sets obtained by randomly sampling (10 times) 2, 4, 6, 8, 10, 20, 30, 40 trials from the entire training set (see Section 2.4). For each reduced training set, the performance of WS decoders is reported together with the ones of TL-WS decoders, to highlight the potential benefit of transferring the knowledge from other subjects (i.e., importing weights from pre-trained LOSO networks) compared to training from scratch (i.e., randomly initializing weights).

Transfer learning was found to be significantly beneficial across the decoded variables (p_x,p_y,v_x,v_y) in the low data regime (i.e., from 2 to 10

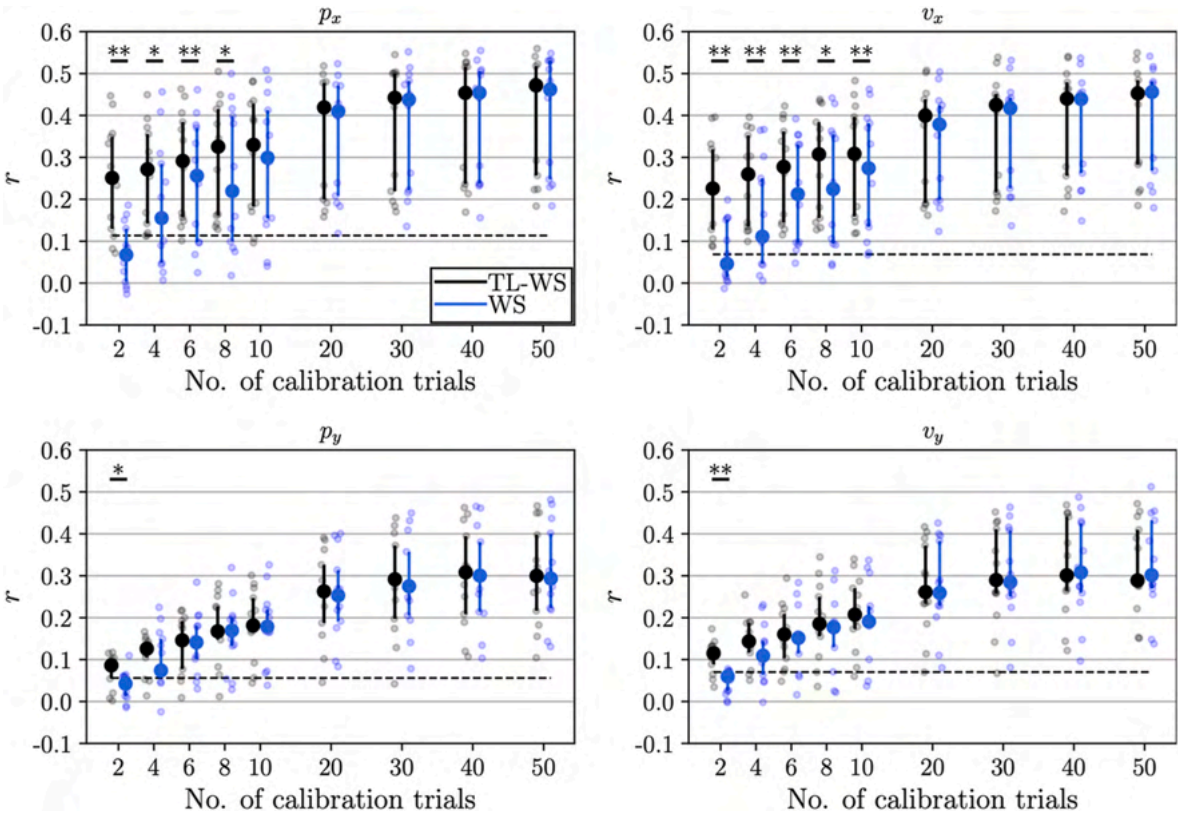

Fig. 7. Performance obtained using reduced training sets: effect of transferring the knowledge on new subjects. Pearson's correlation coefficient (r) is reported for each decoded variable (p_x,p_y,v_x,v_y) in case MS-Sinc-ShallowNet was trained using the entire training set (i.e., 50 trials) and more compact training sets (each with 2, 4, 6, 8, 10, 20, 30, 40 trials). For each condition, the network was trained using both within-subject strategy (WS, blue) and transfer learning strategy (TL-WS, black). Horizontal dashed lines denote the chance level. Significant p-values corrected for multiple tests are reported (*p $<$ 0.05, **p $<$ 0.01, ***p $<$ 0.001). Only significant comparisons are indicated.

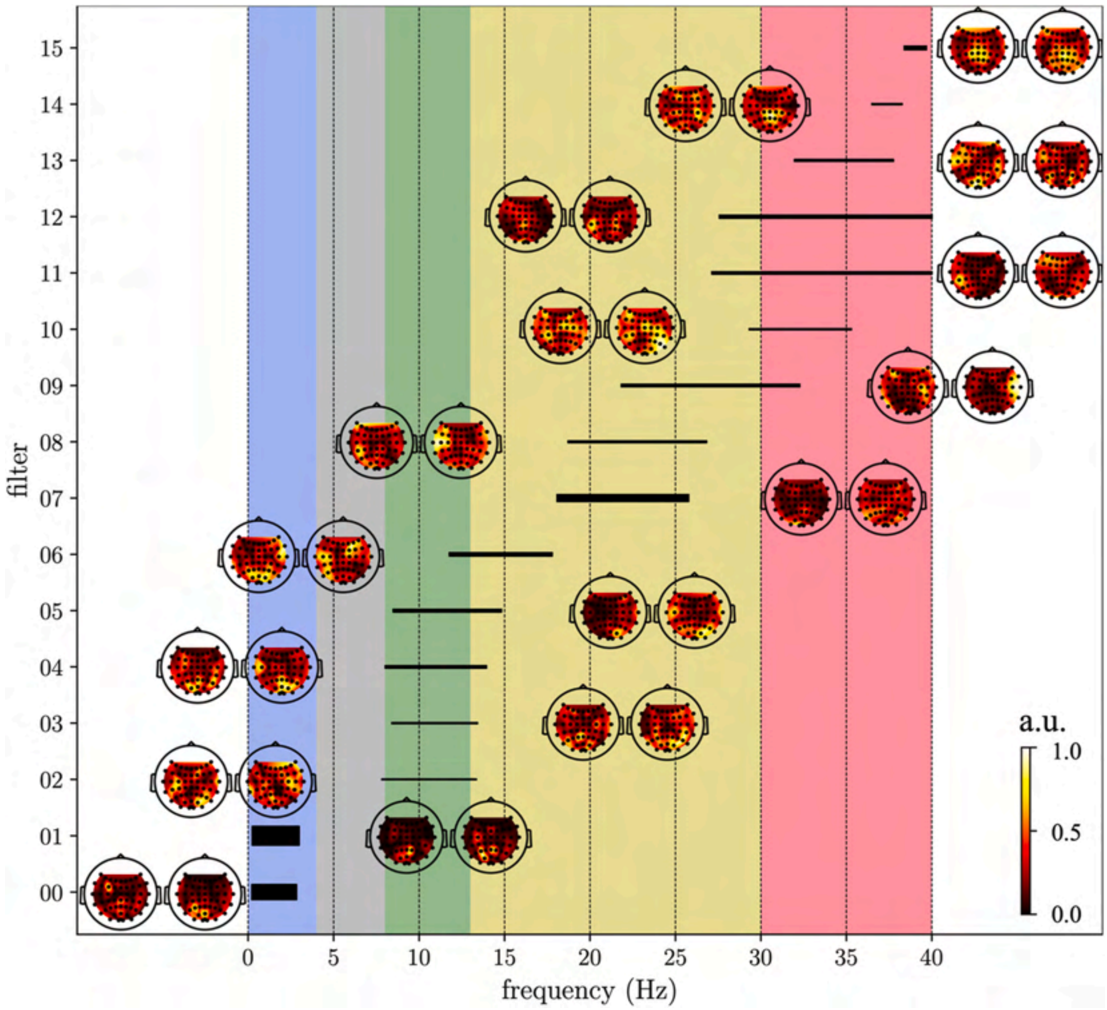

Fig. 8. Features learned by MS-Sinc-ShallowNet for a representative subject (same considered in Fig. 5). The distribution of the band-pass filters learned by the temporal sinc-convolutional layer is reported; each filter is represented as a horizontal black bar with endpoints corresponding to the cutoff frequencies. The bar width encodes the relevance of each band-pass filter for decoding both positions and velocities (the thicker the more important). Near to each band-pass filter, the associated set of spatial filters (in their absolute value) is displayed too. These were normalized from 0 to 1 (see colorbar). EEG bands are color-coded as: blue: delta, grey: theta, green: alpha, yellow: beta, red: low-gamma.

trials), especially in case of the smallest training set (e.g., with 2 training trials). Conversely, in higher data regime (i.e., from 20 to 50 trials), no significant improvements were observed in either variable. Furthermore, it is worth noticing that across the experiments performed in the low data regime (from 2 to 10 trials) the performance metrics were significantly above chance in a larger number of subjects in case of TL-WS than in case of WS (9 subjects vs. 4) across all decoded variables.

3.3. *Spectral and spatial relevance related to position and velocity*

By design, the proposed ICNN allows the learned features to be easily extracted and interpreted in the frequency and spatial domains (see Section 2.3.1, and 2.5). As an example, the spectral and spatial features learned by MS-Sinc-ShallowNet are reported in Fig. 8 for a representative subject (the same considered in Fig. 5). Here, the distribution of the interpretable band-pass filters learned in the first convolutional layer is reported together with the associated spatial filters learned in the second convolutional layer. Furthermore, the relevance of using a specific band-pass filter is reported too (modulating the bar widths); here, the relevance of each filter of the considered subject was measured by considering its relevance score (see Eq. C.1) associated to each decoded variable ($\{p_x, p_y, v_x, v_y\}$), and by averaging these four relevance scores. For the proposed decoder, filters falling within the delta-band were the most relevant ones to predict kinematics (filters no. 0 and no. 1); however, also filters at higher frequencies, e.g., beta-band (filter no. 7), were highly relevant, suggesting a role also for high frequencies in the prediction of kinematic variables at least for this analyzed subject. Lastly, the spatial filters associated to the most important band-pass filters (falling in the delta-band) were highly selective for few electrode sites only, mainly at central/centro-parietal electrodes.

Fig. 9 summarizes the results across all subjects as to the spectral relevance for kinematic decoding. Specifically, the figure shows the spectral relevance and the EEG band relevance, separately for position and velocity. The frequency components in the delta-band are the ones with highest relevance, for both position and velocity. This is further confirmed by the statistical analysis of the EEG band relevance: for both position and velocity, significant differences were found in the EEG band relevance across bands ($p < 0.001$, Friedmann test) and, in particular, between delta and each of the other bands (theta, alpha, beta, low-gamma, $p < 0.001$ post-hoc pairwise tests).

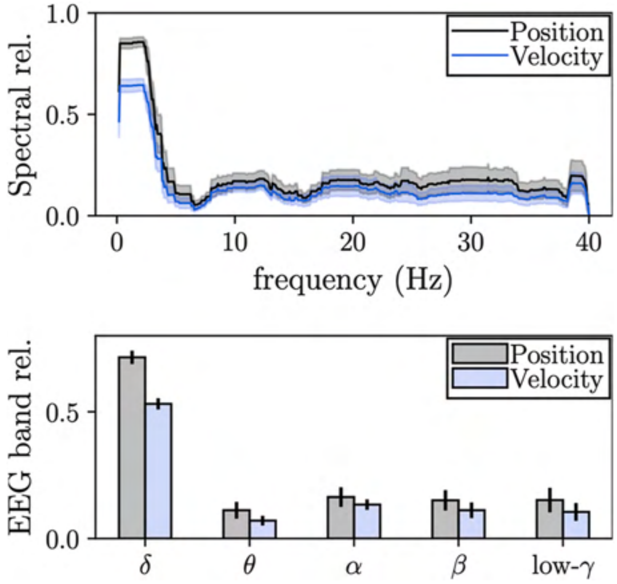

Fig. 9. Spectral relevance (top) and EEG band relevance (bottom) attributed by the ICNN for decoding position (black) and velocity (blue). Mean and standard error of the mean across subjects of the spectral relevance (thick line and shaded area) and of the EEG band relevance (height of the bars and error bar) are reported.

When clustering the subjects based on EEG band relevance, two clusters were obtained (cluster 0 with 6 subjects, cluster 1 with 7 subjects), and no outliers detected. The clusters gave information on the most common strategies picked by the ICNN in the frequency domain to decode position and velocity. From Fig. 10, it is evident that the delta-band had the highest relevance in both clusters, meaning that the delta-band was widely exploited across subjects. Nevertheless, subjects in cluster 0 additionally exhibited relevance of higher frequency ranges such as alpha, beta and low-gamma, for the decoding problem. Regarding the spatial relevance, this was investigated separately for each cluster. According to the results on the spectral relevance, we considered the spatial relevance in the delta range for both clusters and, in addition, in the alpha, beta and low-gamma ranges for cluster 0. In these examined EEG bands, the most relevant electrodes to decode hand position and velocity covered the contralateral central/centro-parietal and parietal/parieto-occipital sites.

4. Discussion

In this study, we used a light and interpretable CNN (MS-Sinc-ShallowNet) to reconstruct 2-D positions and velocities from the EEG during a pursuit tracking task. The ICNN was designed using network components previously validated [37,38] to learn interpretable spectral and spatial features in its first layers, and was adapted to learn deeper temporal features at multiple time scales in parallel. ICNN layers were designed to ensure a limited model size by adopting interpretable, depthwise and separable convolutions [37,38]. As main points of contribution of the present study to the field of EEG trajectory decoding, the proposed decoder was trained with different strategies including within-subject (WS), leave-one-subject-out (LOSO), and transfer learning (TL-WS), in order not only to test its potentialities for EEG trajectory decoding using subject-specific training (WS, as usually done in literature) but also to test the feasibility of calibration-free use (LOSO), and of transferring the knowledge from other subjects to a new one (TL-WS). Furthermore, the DNN adopted here for EEG trajectory decoding was interpretable in its nature, thus enabling an easy interpretation of the learned spectral and spatial features. The increased interpretability of the decoder was coupled with saliency maps (ICNN + ET combination) to illustrate how the most relevant EEG features learned for decoding the kinematics variables can be disclosed, and thus showing how a DNN, usually considered a black box, can transform into a (at least partially) glass box.

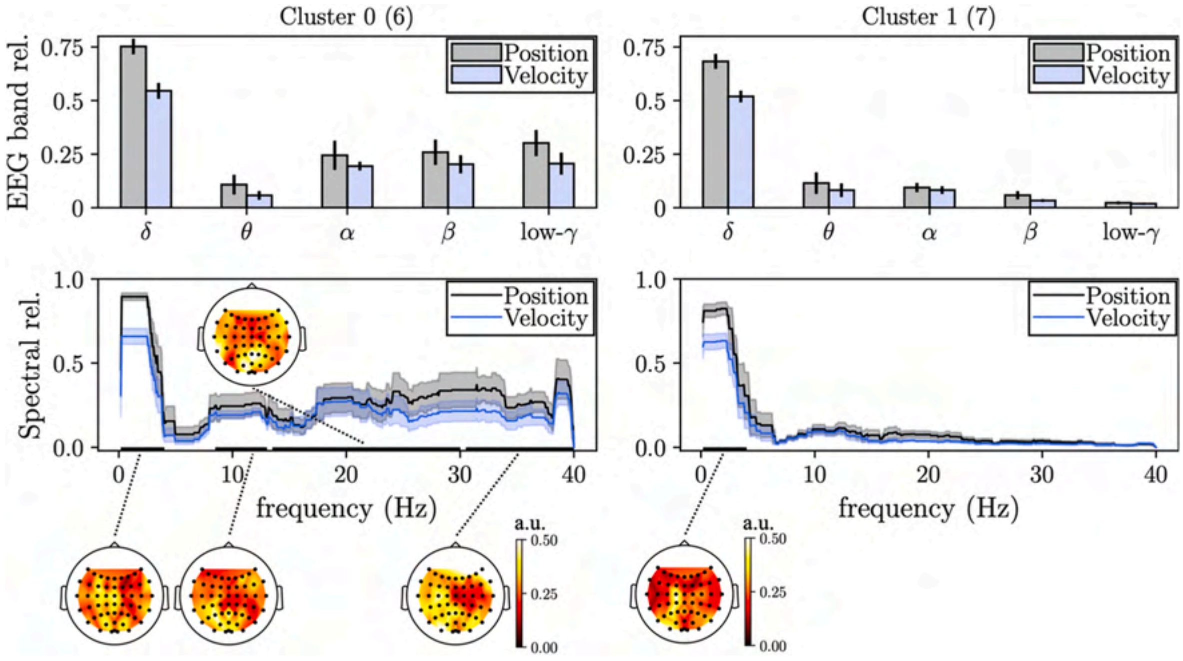

Fig. 10. Cluster analysis of the most relevant EEG spectral and spatial features related to position and velocity. The figure shows the features of the two identified clusters, i.e., cluster 0 (left, 6 subjects) and cluster 1 (right, 7 subjects). The number of subjects is reported within brackets. For each cluster, mean and standard error of the mean across subjects of the EEG band relevance (bar height and error bar, on top) and of the spectral relevance (thick line and shaded area, on bottom) are reported. The topological maps of the spatial relevance linked to the analyzed EEG bands (black lines on the x-axis) are displayed, too.

4.1. Performance of MS-Sinc-ShallowNet and comparison with SOA decoders

The correlations between the predicted and actual hand kinematics obtained via the proposed ICNN were comparable to those obtained via the traditional PLS + UKF decoder, but with a significantly better amplitude reconstruction, as denoted by the significantly lower RMSEs scored by the ICNN (see Fig. 3). This is of relevance as, despite the adopted PLS + UKF was designed to alleviate the amplitude mismatch problem in trajectory decoding [18], the proposed decoder provided a better estimation of both position and velocity amplitudes. The absence of an improvement of ICNN vs. PLS + UKF as to the correlation between predicted and actual values could be due to the nature of the adopted sliding window decoding approach (see Supplementary Fig. 1, Section 2.1, and Section 2.2). Indeed, the ICNN was forced to produce an output for each 1s-length EEG chunk provided as input, independently. During training, the loss function to be minimized is defined only by the mean squared error computed on each EEG chunk, without imposing any regularization term across chunks, e.g., a smoothness constraint across neighboring EEG chunks (i.e., across neighboring trajectory points). Therefore, this might have limited the correlation measure for CNN-based approaches and the effect of additional penalty terms in the loss function could be investigated in future studies.

MS-Sinc-ShallowNet significantly outperformed the tested DNNs that were specifically released for EEG trajectory decoding (StackedLSTM, StackedGRU, and EEGNet-LSTM), for all decoded variables. Furthermore, our approach performed on par or outperformed significantly also other DNNs proposed for EEG motor classification (Sinc-ShallowNet, ShallowConvNet, DeepConvNet) and adapted in this study for EEG trajectory decoding. In particular, when comparing MS-Sinc-ShallowNet with Sinc-ShallowNet, it turns out that the multi-scale temporal feature learning significantly increased correlations between predicted and true trajectories. While the proposed ICNN always outperformed ShallowConvNet, overall, it performed on par with DeepConvNet. However, when considering also other aspects of all the compared DNNs, such as model size and training time (see Fig. 4), the proposed ICNN represented a better compromise between model performance, size and training time, being slightly more accurate, ~30 times lighter (approx. 9 K vs. 273 K trainable parameters of DeepConvNet) and requiring the same time to be trained. Interestingly, both the tested ICNN models (Sinc-ShallowNet and MS-Sinc-ShallowNet) resulted in the best trade-off between model performance, size and training time compared to other DNNs, with MS-Sinc-ShallowNet being more accurate due to the used multi-scale feature extractor. This advantage of ICNNs was obtained even though, by incorporating interpretability into the model structure, the capacity of the decoder reduces. Indeed, in these models the interpretability increased by exploiting a re-parametrization that limited the model to explore only band-pass filters in the temporal domain. Overall, these results suggest that adopting an interpretable design could not only ease the interpretation of the learned features of DNNs but also might improve the quality of the decoding, limiting at the same time the model size and training time.

Lastly, it is worth noticing that for all decoders correlations were lower for trajectories predicted in the y-axis than in the x-axis, as found in a previous study on a subset of the adopted dataset [27]. This might be related to the experimental setup, where the screen was tilted towards the y-axis to facilitate the movements of the robot. Thus, the perception of movements of the moving object along the y-axis may be ambiguous in comparison with the ones along the x-axis [27].

4.2. Performance when transferring the knowledge from other subjects

MS-Sinc-ShallowNet was further evaluated by testing its ability to transfer the knowledge learned from other subjects to a held-out subject, with the aim of reducing the training time of the decoder. To perform transfer learning, an architecture pre-trained on other subjects (different from the held-out one) was used, corresponding to the LOSO model (see Fig. 6). Compared to training networks from scratch (i.e., randomly initialized, WS in Fig. 7), transfer learning (TL-WS in Fig. 7) led to a significant increase in decoding performance (up to a median increase of 0.18 in correlations) only in case of low data regime (from 2 to 10 training trials, especially with 2 training trials) also increasing the number of subjects decoded significantly above chance. Therefore, the LOSO model, by capturing relevant cross-subject features, represented a significantly better initialization point in the parameter space than the random one, for training the network on a new subject when a few training trials are available. This could have prospective implication for a practical usage of the decoder in BCI systems, thanks to the potentiality of transfer learning of reducing the number of training trials required to perform above chance, and thus promoting a reduction of training times during BCI sessions.

4.3. Spectral and spatial relevance related to position and velocity

We took advantage of the proposed ICNN, combined with an ET (ICNN + ET), to illustrate how the most relevant neural features for the decoding of positions and velocities, in both spectral and spatial domains, could be disclosed. For both kinematic variables, the delta-band resulted to be the most relevant (see Fig. 9), while higher frequency bands (e.g., alpha, beta and low-gamma) appeared to be relevant, in addition to delta, but with higher variability across subjects. Specifically, the contributions of higher frequency ranges emerged when the spectral relevance and the EEG band relevance were analyzed at the level of each cluster of subjects. Two clusters were automatically identified from the clustering analysis (see Fig. 10). While both clusters showed the highest relevance (with similar values across the two clusters) for the delta-band, one of the two clusters additionally showed higher relevance for alpha, beta and low-gamma ranges, compared to the other cluster. This result suggests that the ICNN widely exploited the delta-band across all subjects, while the contribution of higher frequency ranges to solve the decoding problem was relevant only in some cases.

The highest relevance found for the delta-band agrees with findings reported in literature, supporting the hypothesis that the low-frequency (<3 Hz) band of the EEG overall contains highly relevant information for the decoding of voluntary movement [19–21,23,24,26,27], and widely across subjects. However, our results further suggest that higher frequency ranges like beta (e.g., see the spectral features for the subject reported in Fig. 8) and low-gamma might also have a role and carry information about the movement, although the extent of their contribution is more variable across subjects (see Fig. 10). This is in line with a previous study [25], where circular arm movements were decoded from both low-frequency amplitude features and higher-frequency power features. In particular, while the trajectories could be reconstructed from all subjects when using the low-frequency features, they could not always be successfully estimated when using the higher-frequency components alone [25].

Previous decoding studies suggested how the kinematic information can be best decoded from amplitude features in the low-frequency range, however, power features should be used for higher frequencies [21,22,25], as they likely reflect the well-known modulation of sensorimotor rhythms with voluntary movement. Provided that in our approach the network takes as input the amplitude of the signal in a wider frequency range, it might appear like only the amplitude features are used, independently of the frequency content. It should, however, be noted that a CNN generally approximates non-linear functions, which may produce an equivalent effect of computing the power of the signal. Therefore, it might not be excluded that non-linear features extracted from the signal (equivalent, for example, to computing the power) are being exploited by the network at higher frequency components, with the advantage that the bandwidth of the filters is not to be determined 'a-priori', but is automatically learned by the network, according to the most relevant and subject-specific content to solve the decoding problem.

From the spatial relevance, we disclosed which electrodes were the most important inside the different EEG bands for decoding position and velocity. Across frequency bands, the most relevant electrodes for decoding kinematics were over the contra-lateral (i.e., left) primary sensorimotor areas (Fig. 10), therefore possibly reflecting the modulations of sensorimotor rhythms accompanying voluntary movement, and in line with the findings of previous studies [22,25,74]. Moreover, the ICNN appeared to rely also on parieto-occipital sites to decode the kinematics, similarly to the findings of [20,26,27] for the delta-band, and [21,22,25] for the beta-band. This could be explained by the nature of the task, which not only involved the hand movement, but also visual processing and eye-hand movement coordination [75].

4.4. Limitations of the current study

Overall, the results obtained with the proposed framework are well promising, both in terms of performance and quality of the spectral and spatial features learned by the network. However, the present study has some limitations that could be addressed in the future.

i. A relatively small dataset (13 participants) and only one motor paradigm (pursuit tracking task) were used to test our framework. These factors could have limited the validation of the network, both as to its decoding performance and its capability of analyzing EEG features. In particular, the low number of subjects could have reduced the quality of the learned cross-subject features in LOSO models, limiting in turn the performance scored during subject-to-subject transfer learning in the low data regime (up to approx. $r = 0.3$, across subjects) vs. using more training trials (up to approx. $r = 0.45$, across subjects). The performance analysis as well as the insights gained by DNNs about the EEG features require further investigations, by validating the robustness of the presented approach using larger datasets and even using datasets acquired in different experimental paradigms, e.g., involving reaching and/or reach-to-grasping.

ii. The ICNN feature analysis was conducted only in the frequency and spatial domains, without addressing the temporal domain; thus, the definition in the future of a complete analysis framework that analyzes relevant features also in time would be of high interest.

iii. The subjects used in this study were all healthy, while in real life users approaching BCIs are afflicted by motor impairments; thus, of course our results are to be intended as preliminary results obtained on healthy subjects and our approach needs to be validated in the future on patients.

5. Conclusion

In this study, we investigated the use of an ICNN for EEG trajectory decoding, specifically by exploiting a light architecture that learns interpretable spectral and spatial features, and deep temporal features at two time scales. The ICNN provided a significant better amplitude reconstruction of 2-D positions and velocities compared to a more traditional decoder based on PLS + UKF and widely outperformed other DNNs, providing a better trade-off between model performance, size, and training time, while at the same time enabling an easy interpretation of the learned spectral and spatial features. Thus, the proposed ICNN may have practical implications for designing solutions allowing a better reconstruction of kinematics from EEG and a more natural control of actuators in BCIs. Furthermore, transfer learning significantly improved the performance especially when using few training trials of the new user. Prospectively, this could lead to a significant reduction of calibration times and could contribute to the development of accurate and 'plug-and-play' decoders for trajectory decoding. Lastly, results on the most relevant spectral and spatial features related to kinematics highlighted by our ICNN + ET algorithm, although preliminary, are in line with previous studies analyzing event-related spectral perturbations, with the most relevant spectral features localized in the delta-band consistently across subjects (although also alpha, beta, and low-gamma appeared to have some relevance too but will less consistency), and spatial features mostly localized at sensorimotor and parieto-occipital sites. Thus, although further studies are necessary to obtain wider validation, an ICNN + ET algorithm appears capable of capturing features matching neurophysiological correlates in a data-driven fashion.

Credit author contributions

Davide Borra: Conceptualization, Methodology, Software, Formal analysis, Visualization, Writing - Original Draft, Writing - Review & Editing.

Valeria Mondini: Investigation, Data Curation, Writing - Review & Editing.

Elisa Magosso: Conceptualization, Validation, Supervision, Funding acquisition, Writing - Original Draft, Writing - Review & Editing.

Gernot R. Müller-Putz: Investigation, Data Curation, Funding acquisition, Writing - Review & Editing.

Declaration of competing interest

None Declared.

Acknowledgments

The authors declare no conflict of interest.

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the TITAN V used for this research. The provider was not involved in the study design, collection, analysis, interpretation of data, the writing of this article or the decision to submit it for publication.

This work was supported by #NEXTGENERATIONEU (NGEU) and funded by the Ministry of University and Research (MUR), National Recovery and Resilience Plan (NRRP), project MNESYS (PE0000006) — A Multiscale integrated approach to the study of the nervous system in health and disease (DN. 1553 11.10.2022).

Parts of the work were funded by the ERC project 'Feel Your Reach' (681231).

Appendix

A. Temporal sinc-convolutional layer

Denoting with k_l the l-th convolutional kernel, in a conventional convolutional layer each filter value (i.e. $k_l[0,n], n \in [0,50]$) has to be learned during the optimization process; conversely, in a sinc-convolutional layer, each filter value is defined by a parametrized function, forcing the overall filter distribution to belong to a specific subset of temporal filters (here only band-pass filters). Therefore, in a sinc-convolutional layer a re-parametrization of each kernel occurs:

$$k_l^{'}\left[0,n;\left\{f_{0,l},f_{1,l}\right\}\right] = 2f_{1,l}sinc\left(2\pi f_{1,l}n\right) - 2f_{0,l}sinc\left(2\pi f_{0,l}n\right), 0 \le l \le K_0^{ISS} - 1. \tag{A.1}$$

In Eq. A.1, $\{f_{0,l}, f_{1,l}\}$ is the set of trainable parameters related to the l-th kernel, including only the inferior ($f_{0,l}$) and superior ($f_{1,l}$) cutoff frequencies of the band-pass filter. In this way, for each temporal filter the number of trainable parameters reduces from 51 ($= F_0^{ISS}[0] \bullet F_0^{ISS}[1]$) to 2. Lastly, to alleviate the effects of the inevitable truncation of $k_l^{'}$ on the characteristics of each filter, the multiplication by a Hamming window is performed:

$$\begin{cases} k_{w,l}^{'}\left[0,n;\left\{f_{0,l},f_{1,l}\right\}\right] = k_l^{'}\left[0,n;\left\{f_{0,l},f_{1,l}\right\}\right] \bullet w[n] \\ w[n] = 0.54 - 0.46\cos\left(\dfrac{2\pi n}{F_0^{ISS}[1]-1}\right) \end{cases}. \tag{A.2}$$

Accordingly, the temporal sinc-convolution computes the convolution between the input and $k_{w,l}^{'}[0,n;\{f_{0,l},f_{1,l}\}]$, learning only the following 2 parameters for each kernel.

B. State-of-the-art decoders

The proposed ICNN was compared against other 7 SOA decoders, to provide a wide comparison with respect to the literature. The SOA decoders were:

i. The best-performing SOA machine learning algorithm proposed for EEG trajectory decoding, represented by the *PLS* + *UKF* decoder proposed in Kobler et al. [18], which was carefully designed to alleviate the amplitude mismatch problem characterizing linear decoders.

ii. DNNs proposed for EEG motor classification, modified here to reconstruct kinematics from the EEG. Among these, we included the single scale ICNN from which the MS-Sinc-ShallowNet originates from, i.e., *Sinc-ShallowNet* [37] (see Section 2.3 for a brief description of Sinc-ShallowNet) and CNNs, including *DeepConvNet* [34] (consisting of 5 convolutional layers and one fully-connected layer) and *ShallowConvNet* (consisting of 2 convolutional layers and one fully-connected layer) [34]. These three DNNs represents successful solutions designed specifically for sensorimotor rhythm classification (both for executed and imagined motor states): *Sinc-ShallowNet* denotes a recent CNN with interpretable components, and *DeepConvNet* and *ShallowConvNet* are among CNNs that were assessed for EEG trajectory decoding in previous research [49]. In these networks, we replaced the last layer (softmax activated fully-connected layer) with a linearly activated fully-connected layer with 4 output neurons as the one included in MS-Sinc-ShallowNet (see Section 2.3.3), in order to solve the objective regression problem instead of classification. In addition, kernels (both convolutional and pooling) operating in the temporal domain were scaled down in their size by a factor of 2, since these networks originally performed classification from EEG signals sampled at 250 Hz while here 100 Hz EEG signals were given as input. Except for these changes, these DNNs were used with their original architectural hyper-parameters.

iii. DNNs proposed for EEG trajectory decoding. To this aim, we included the best-performing algorithms resulting from recent studies [46,49]. Specifically, we included the RNN design described by Nakagome et al. [46], composed by multiple GRU layers (i.e., *StackedGRU*) by using the hyper-parameters suggested by the authors. Specifically, we stacked 3 GRU layers, each of 64 units, on top of a linearly activated fully-connected layer with 4 output neurons that finalize regression (as the fully-connected layer used in MS-Sinc-ShallowNet). Furthermore, adopting the same design of StackedGRU, we also tested LSTM cells (i.e., *StackedLSTM*) in place of GRU cells, as LSTM cells were considered as constitutive parts in the networks proposed in Chen et al. [49]. Moreover, in this way, we provide also here a comparison between GRU and LSTM cells in multi-layer RNNs, as performed in Nakagome et al. [46]. Lastly, we included the hybrid convolutional-recurrent network *EEGNet-LSTM* proposed by Chen et al. [49] (consisting of 3 convolutional layers, 2 LSTM layers, and one fully-connected layer with 4 neurons), adopting the same architectural hyper-parameters suggested by the authors.

C. Computation of the most relevant spectral and spatial features related to position and velocity

C.1. Spectral relevance computation

In a first stage, we computed the relevance of each spectral component to predict the positions and velocities, based on the K_0^{ISS} feature maps from the sinc-convolutional layer. These maps contain the input filtered by the learned band-pass filters. A schematization of the following steps is reported in Fig. C.1.

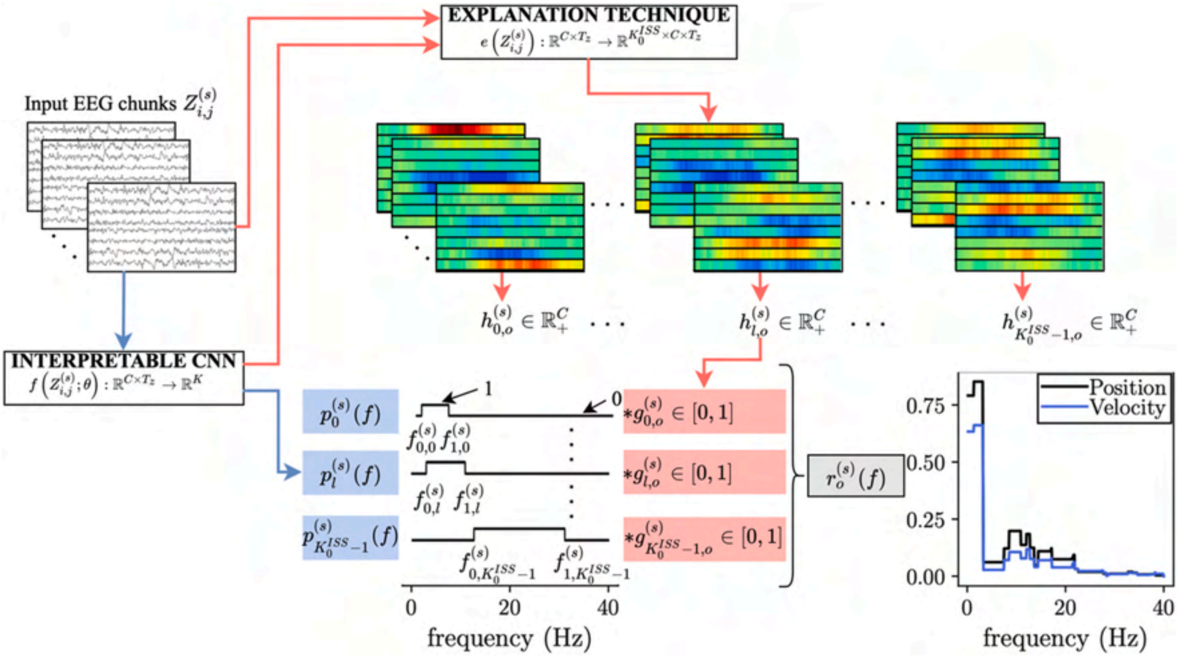
EXPLANATION TECHNIQUE
$e\left(Z^{(s)}_{i,j}\right): \mathbb{R}^{C\times T_z} \rightarrow \mathbb{R}^{K_0^{ISS}\times C\times T_z}$
Input EEG chunks $Z^{(s)}_{i,j}$
INTERPRETABLE CNN
$f\left(Z^{(s)}_{i,j};\theta\right): \mathbb{R}^{C\times T_z} \rightarrow \mathbb{R}^{K}$
$h^{(s)}_{0,o} \in \mathbb{R}^C_+$
$h^{(s)}_{l,o} \in \mathbb{R}^C_+$
$h^{(s)}_{K_0^{ISS}-1,o} \in \mathbb{R}^C_+$
$p^{(s)}_0(f)$
$p^{(s)}_l(f)$
$p^{(s)}_{K_0^{ISS}-1}(f)$
1
0
$f^{(s)}_{0,0}$ $f^{(s)}_{1,0}$
$f^{(s)}_{0,l}$ $f^{(s)}_{1,l}$
$f^{(s)}_{0,K_0^{ISS}-1}$ $f^{(s)}_{1,K_0^{ISS}-1}$
$*g^{(s)}_{0,o} \in [0,1]$
$*g^{(s)}_{l,o} \in [0,1]$
$*g^{(s)}_{K_0^{ISS}-1,o} \in [0,1]$
$r^{(s)}_o(f)$
0
20
40
frequency (Hz)
0.75
0.50
0.25
0.00
0
20
40
frequency (Hz)
Position
Velocity

Figure C.1. Scheme of the spectral relevance computation. The learned ICNN spectral features $\theta^{(s)}_{spect,l}, 0 \leq l \leq K_0{}^{ISS} - 1$ are extracted and $p^{(s)}_l(f)$ computed (blue boxes and lines). By combining the ICNN with an explanation technique, relevance scores related to each spectral feature are obtained $g^{(s)}_{l,o}, o \in \{p_x, p_y, v_x, v_y\}$ (red boxes and lines). Then, spectral features and relevance scores are combined to derive the subject-specific spectral relevance of each frequency bin, $r^{(s)}_o(f), o \in \{p, v\}$.

For the EEG chunks of the test set, we evaluated the relevance of each spatio-temporal sample in the feature map to decode the 2-D positions and velocities. To do this, we computed saliency maps [52] to quantify, by using gradients, how much a spatio-temporal sample in each filtered input affects the prediction of each kinematic variable (p_x,p_y,v_x,v_y). Therefore, we obtained, for each output variable, one saliency map for each feature map of the first convolutional layer, i.e., $e(Z^{(s)}_{i,j}) : \mathbb{R}^{C\times T_z} \rightarrow \mathbb{R}^{K_0{}^{ISS}\times C\times T_z}$. When explaining the decision of networks applied to EEG, saliency maps are widely used [37,38,40,40,47,53,69], with the advantage of requiring the sole computation of gradients via backpropagation. Of course, more advanced and recent techniques, such as layerwise relevance propagation (LRP) [67], gradient-weighted class activation mapping (Grad-CAM) [68], and shapley additive explanation (SHAP) [76] can represent valid alternatives to saliency maps, but these were only used in few studies with EEG [77–80]. Indeed, saliency representations are generally preferred to keep the explanation process as simple as possible [38], without introducing too many factors that could influence the obtained representations, e.g., the type of propagation rule used (e.g., ε rule) in LRP. Thus, we adopted saliency maps in this study; however, it should be noted that the same framework presented in our study could be easily used with any explanation technique by replacing saliency maps by any other technique (e.g., LRP), as it is an explanation technique-independent framework.

The so computed saliency maps were averaged across trials ($\forall i$), chunks ($\forall j$), and in the temporal domain ($\forall t, 0 \leq t \leq T_z - 1$). By finally computing the absolute value, the vector quantities $h^{(s)}_{l,o} \in \mathbb{R}^C_+, 0 \leq l \leq K_0{}^{ISS} - 1, o \in \{p_x, p_y, v_x, v_y\}$ can be obtained, with o indicating the output kinematic variable. Finally, the relevance score $g^{(s)}_{l,o}$ was computed as:

$$g^{(s)}_{l,o} = \underset{c}{avg}\left(h^{(s)}_{l,o}\right) \Big/ \max_{l}\left(\underset{c}{avg}\left(h^{(s)}_{l,o}\right)\right), 0 \leq c \leq C - 1, \tag{C.1}$$

with $g^{(s)}_{l,o}$ being a scalar quantity $\in [0,1]$ summarizing the importance of the l-th band-pass filter for the o-th predicted variable.

Subsequently, the frequencies belonging to the passband of the filter associated to the l-th feature map and defined by $\theta_{spect,l}{}^{(s)}$ (see Eq. (3)) were assigned the corresponding relevance score $g^{(s)}_{l,o}$:

$$\begin{cases} p^{(s)}_l(f) = \begin{cases} 1, if\ f^{(s)}_{0,l} \leq f \leq f^{(s)}_{1,l} \\ 0, elsewhere \end{cases}, \\ q^{(s)}_{l,o}(f) = g^{(s)}_{l,o} \bullet p^{(s)}_l(f) \end{cases} \tag{C.2}$$

where $p^{(s)}_l(f)$ indicates the probability of a frequency f to be included in the passband of the l-th band-pass filter. Finally, the spectral relevance $q^{(s)}_o(f)$, quantifying the relevance of each frequency bin for the o-th kinematic variable, was obtained as:

$$q^{(s)}_o(f) = \underset{l}{\text{avg}}\ q^{(s)}_{l,o}(f). \tag{C.3}$$

From a preliminary analysis, the spectral relevance $q^{(s)}_o(f)$ resulted to be comparable across x- and y-axes for both position and velocity (permutation cluster test with threshold-free cluster enhancement [81], see Supplementary Section 3). Therefore, the $q^{(s)}_o(f)$ was averaged along the axes, thus, obtaining only one average spectral relevance profile $r_o^{(s)}(f)$ for the position and one for the velocity, where the index o hereafter denotes the kinematic variable, i.e., $o \in \{p, v\}$:

$$\begin{cases} r_p^{(s)}(f) = \dfrac{q_{px}^{(s)}(f) + q_{py}^{(s)}(f)}{2} \\ r_v^{(s)}(f) = \dfrac{q_{vx}^{(s)}(f) + q_{vy}^{(s)}(f)}{2} \end{cases}. \tag{C.4}$$

Finally, the spectral relevance for each kinematic variable was averaged within EEG bands (hereafter named 'EEG band relevance'), in the delta (0.18–4 Hz), theta (4–8 Hz), alpha (8–13 Hz), beta (13–30 Hz), and low-gamma (30–40 Hz) bands, so to identify the most relevant spectral features predicting the position or velocity.

C.2. Spectral clustering and spatial relevance computation

In a second stage, we performed automatic clustering to reveal whether certain groups of subjects were sharing common EEG features in the frequency domain, i.e., sharing similar patterns of relevance in the EEG rhythms. To do so, the EEG band relevance of both position and velocity was clustered using Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [82] (by adopting the Python library hdbscan, version 0.8.27), and using the correlation between observations as distance metric. This clustering algorithm was used instead of other solutions (e.g., partitioning clustering algorithms such as K-means [83]) as it does not require to specify the number of clusters a priori, it can identify cluster with arbitrary size and shape, it is suitable for data with arbitrary shape and size, and it can handle noise in data, enabling an easy detection and removal of outliers from clusters [84]. Therefore, by using HDBSCAN the optimal number of clusters according to correlation is automatically learned from the observations. The EEG band relevance was chosen as it summarizes the features of the spectral relevance profile $r_o^{(s)}(f)$ in a compact way (i.e., 2 × 5 features per subject, instead of 2 × frequency bins per subject), being the clustering applied to a limited number of subjects (13 in this study). This procedure automatically divided the 13 subjects into clusters based on their similarity as to the EEG band relevance.

For each cluster, the spectral relevance was first averaged across subjects in the cluster, obtaining an average profile of spectral relevance for that cluster. Then, for each cluster we also computed an average spatial relevance, associated to each EEG band, according to the following procedure.

Let us denote with $[f_{0,r}, f_{1,r}]$, $0 \le r \le 4$ the r-th frequency range defining each of the 5 EEG bands (see end of Appendix C.1). For the s-th subject and for the r-th frequency range we considered the subset of the ICNN band-pass filters, denoted as $S_r^{(s)}$, containing in their passband the frequency bins belonging to $[f_{0,r}, f_{1,r}]$, and we extracted the spatial filters associated to this subset of band-pass filters, i.e., ${\theta_{spat,l}}^{(s)} = \{\theta_{lk}^{(s)}\}$ (see Eq. (4)), $l \in S_r^{(s)}, 0 \le k \le D_1^{ISS} - 1$. Spatial filters were considered in their absolute value, as done in Refs. [37,85]. Subsequently, the absolute spatial features were averaged together, electrode per electrode ($\forall c, 0 \le c \le C - 1$), and normalized to the maximum across electrodes, obtaining the spatial relevance associated to the r-th band for the s-th subject:

$$\sigma_r^{(s)} = \underset{l \in S^{(s)}, k}{\text{avg}}\; abs\left(\overset{(s)}{\underset{lk}{\theta}}\right) \Big/ \max_c \left(\underset{l \in S^{(s)}, k}{\text{avg}}\; abs\left(\theta_{lk}^{(s)}\right)\right). \tag{C.5}$$

Lastly, $\sigma_r^{(s)}$ was averaged across subjects in the cluster.

Appendix D. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compbiomed.2023.107323.

References

[1] J.R. Wolpaw, N. Birbaumer, D.J. McFarland, G. Pfurtscheller, T.M. Vaughan, Brain–computer interfaces for communication and control, Clin. Neurophysiol. 113 (2002) 767–791.

[2] J.D.R. Millán, Combining brain-computer interfaces and assistive technologies: state-of-the-art and challenges, Front. Neurosci. 1 (2010).

[3] G.R. Müller-Putz, R.J. Kobler, J. Pereira, C. Lopes-Dias, L. Hehenberger, V. Mondini, V. Martínez-Cagigal, N. Srisrisawang, H. Pulferer, L. Batistić, A. I. Sburlea, Feel Your reach: an EEG-based framework to continuously detect goal-directed movements and error processing to gate kinesthetic feedback informed artificial arm control, Front. Hum. Neurosci. 16 (2022), 841312.

[4] C.E. Bouton, A. Shaikhouni, N.V. Annetta, M.A. Bockbrader, D.A. Friedenberg, D. M. Nielson, G. Sharma, P.B. Sederberg, B.C. Glenn, W.J. Mysiw, A.G. Morgan, M. Deogaonkar, A.R. Rezai, Restoring cortical control of functional movement in a human with quadriplegia, Nature 533 (2016) 247–250.

[5] J.L. Collinger, B. Wodlinger, J.E. Downey, W. Wang, E.C. Tyler-Kabara, D.J. Weber, A.J. McMorland, M. Velliste, M.L. Boninger, A.B. Schwartz, High-performance neuroprosthetic control by an individual with tetraplegia, Lancet 381 (2013) 557–564.

[6] P. Ofner, G.R. Muller-Putz, Using a noninvasive decoding method to classify rhythmic movement imaginations of the arm in two planes, IEEE Trans. Biomed. Eng. 62 (2015) 972–981.

[7] P. Ofner, G.R. Muller-Putz, Decoding of velocities and positions of 3D arm movement from EEG, in: 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE, San Diego, CA, 2012, pp. 6406–6409.

[8] M. Tangermann, K.-R. Müller, A. Aertsen, N. Birbaumer, C. Braun, C. Brunner, R. Leeb, C. Mehring, K.J. Miller, G.R. Müller-Putz, G. Nolte, G. Pfurtscheller, H. Preissl, G. Schalk, A. Schlögl, C. Vidaurre, S. Waldert, B. Blankertz, Review of the BCI competition IV, Front. Neurosci. 6 (2012).

[9] Y. Chu, X. Zhao, Y. Zou, W. Xu, G. Song, J. Han, Y. Zhao, Decoding multiclass motor imagery EEG from the same upper limb by combining Riemannian geometry features and partial least squares regression, J. Neural Eng. 17 (2020), 046029.

[10] T. Pistohl, T. Ball, A. Schulze-Bonhage, A. Aertsen, C. Mehring, Prediction of arm movement trajectories from ECoG-recordings in humans, J. Neurosci. Methods 167 (2008) 105–114.

[11] G. Schalk, J. Kubánek, K.J. Miller, N.R. Anderson, E.C. Leuthardt, J.G. Ojemann, D. Limbrick, D. Moran, L.A. Gerhardt, J.R. Wolpaw, Decoding two-dimensional movement trajectories using electrocorticographic signals in humans, J. Neural Eng. 4 (2007) 264–275.

[12] L.R. Hochberg, D. Bacher, B. Jarosiewicz, N.Y. Masse, J.D. Simeral, J. Vogel, S. Haddadin, J. Liu, S.S. Cash, P. van der Smagt, J.P. Donoghue, Reach and grasp by people with tetraplegia using a neurally controlled robotic arm, Nature 485 (2012) 372–375.

[13] M. Filippini, D. Borra, M. Ursino, E. Magosso, P. Fattori, Decoding sensorimotor information from superior parietal lobule of macaque via Convolutional Neural Networks, Neural Network. 151 (2022) 276–294.

[14] H.G. Yeom, J.S. Kim, C.K. Chung, Estimation of the velocity and trajectory of three-dimensional reaching movements from non-invasive magnetoencephalography signals, J. Neural Eng. 10 (2013), 026006.

[15] S. Waldert, H. Preissl, E. Demandt, C. Braun, N. Birbaumer, A. Aertsen, C. Mehring, Hand movement direction decoded from MEG and EEG, J. Neurosci. 28 (2008) 1000–1008.

[16] A.P. Georgopoulos, F.J.P. Langheim, A.C. Leuthold, A.N. Merkle, Magnetoencephalographic signals predict movement trajectory in space, Exp. Brain Res. 167 (2005) 132–135.

[17] T.J. Bradberry, F. Rong, J.L. Contreras-Vidal, Decoding center-out hand velocity from MEG signals during visuomotor adaptation, Neuroimage 47 (2009) 1691–1700.

[18] R.J. Kobler, A.I. Sburlea, V. Mondini, M. Hirata, G.R. Müller-Putz, Distance- and speed-informed kinematics decoding improves M/EEG based upper-limb movement decoder accuracy, J. Neural Eng. 17 (2020), 056027.

[19] T.J. Bradberry, R.J. Gentili, J.L. Contreras-Vidal, Reconstructing three-dimensional hand movements from noninvasive electroencephalographic signals, J. Neurosci. 30 (2010) 3432–3437.

[20] R.J. Kobler, A.I. Sburlea, G.R. Müller-Putz, Tuning characteristics of low-frequency EEG to positions and velocities in visuomotor and oculomotor tracking tasks, Sci. Rep. 8 (2018), 17713.

[21] J. Lv, Y. Li, Z. Gu, Decoding hand movement velocity from electroencephalogram signals during a drawing task, Biomed. Eng. Online 9 (2010) 64.

[22] A. Korik, R. Sosnik, N. Siddique, D. Coyle, Decoding imagined 3D hand movement trajectories from EEG: evidence to support the use of Mu, beta, and low gamma oscillations, Front. Neurosci. 12 (2018) 130.

[23] A. Úbeda, J.M. Azorín, R. Chavarriaga, J. del R. Millán, Classification of upper limb center-out reaching tasks by means of EEG-based continuous decoding techniques, J. NeuroEng. Rehabil. 14 (2017) 9.

[24] A. Úbeda, E. Hortal, E. Iáñez, C. Perez-Vidal, J.M. Azorín, Assessing movement factors in upper limb kinematics decoding from EEG signals, PLoS One 10 (2015), e0128456.

[25] R.J. Kobler, I. Almeida, A.I. Sburlea, G.R. Müller-Putz, Using machine learning to reveal the population vector from EEG signals, J. Neural Eng. 17 (2020), 026002.

[26] V. Mondini, R.J. Kobler, A.I. Sburlea, G.R. Müller-Putz, Continuous low-frequency EEG decoding of arm movement for closed-loop, natural control of a robotic arm, J. Neural Eng. 17 (2020), 046031.

[27] V. Martinez-Cagigal, R.J. Kobler, V. Mondini, R. Hornero, G.R. Muller-Putz, Non-linear online low-frequency EEG decoding of arm movements during a pursuit tracking task, in: 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), IEEE, Montreal, QC, Canada, 2020, pp. 2981–2985.

[28] N. Robinson, A.P. Vinod, Noninvasive brain-computer interface: decoding arm movement kinematics and motor control, IEEE Syst. Man Cybern. Mag. 2 (2016) 4–16.

[29] J.-H. Kim, F. Biessmann, S.-W. Lee, Decoding three-dimensional trajectory of executed and imagined arm movements from electroencephalogram signals, IEEE Trans. Neural Syst. Rehabil. Eng. 23 (2015) 867–876.

[30] G. Lindsay, Convolutional neural networks as a model of the visual system: past, present, and future, J. Cognit. Neurosci. (2020) 1–15.

[31] Y. Yu, X. Si, C. Hu, J. Zhang, A review of recurrent neural networks: LSTM cells and network architectures, Neural Comput. 31 (2019) 1235–1270.

[32] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T.H. Falk, J. Faubert, Deep learning-based electroencephalography analysis: a systematic review, J. Neural Eng. 16 (2019), 051001.

[33] A. Craik, Y. He, J.L. Contreras-Vidal, Deep learning for electroencephalogram (EEG) classification tasks: a review, J. Neural Eng. 16 (2019), 031001.

[34] R.T. Schirrmeister, J.T. Springenberg, L.D.J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, T. Ball, Deep learning with convolutional neural networks for EEG decoding and visualization, Hum. Brain Mapp. 38 (2017) 5391–5420.

[35] V.J. Lawhern, A.J. Solon, N.R. Waytowich, S.M. Gordon, C.P. Hung, B.J. Lance, EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces, J. Neural Eng. 15 (2018), 056013.

[36] M. Simões, D. Borra, E. Santamaría-Vázquez, Gbt-Upm, M. Bittencourt-Villalpando, D. Krzemiński, A. Miladinović, Neural_Engineering_Group, T. Schmid, H. Zhao, C. Amaral, B. Direito, J. Henriques, P. Carvalho, M. Castelo-Branco, BCIAUT-P300: a multi-session and multi-subject benchmark dataset on autism for P300-based brain-computer-interfaces, Front. Neurosci. 14 (2020), 568104.

[37] D. Borra, S. Fantozzi, E. Magosso, Interpretable and lightweight convolutional neural network for EEG decoding: application to movement execution and imagination, Neural Network. 129 (2020) 55–74.

[38] D. Borra, S. Fantozzi, E. Magosso, A lightweight multi-scale convolutional neural network for P300 decoding: analysis of training strategies and uncovering of network decision, Front. Hum. Neurosci. 15 (2021), 655840.

[39] D. Borra, S. Fantozzi, E. Magosso, 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, Springer International Publishing, Cham, 2020, pp. 1837–1843.

[40] A. Farahat, C. Reichert, C. Sweeney-Reed, H. Hinrichs, Convolutional neural networks for decoding of covert attention focus and saliency maps for EEG feature visualization, J. Neural Eng. 16 (2019), 066010.

[41] G. Bressan, G. Cisotto, G.R. Müller-Putz, S.C. Wriessnegger, Deep learning-based classification of fine hand movements from low frequency EEG, Future Internet 13 (2021) 103.

[42] C. Zhang, Y.-K. Kim, A. Eskandarian, EEG-inception: an accurate and robust end-to-end neural network for EEG-based motor imagery classification, J. Neural Eng. 18 (2021), 046014.

[43] X. Zhao, H. Zhang, G. Zhu, F. You, S. Kuang, L. Sun, A multi-branch 3D convolutional neural network for EEG-based motor imagery classification, IEEE Trans. Neural Syst. Rehabil. Eng. 27 (2019) 2164–2177.

[44] A.M. Roy, An efficient multi-scale CNN model with intrinsic feature integration for motor imagery EEG subject classification in brain-machine interfaces, Biomed. Signal Process Control 74 (2022), 103496.

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

[46] S. Nakagome, T.P. Luu, Y. He, A.S. Ravindran, J.L. Contreras-Vidal, An empirical comparison of neural networks and machine learning algorithms for EEG gait decoding, Sci. Rep. 10 (2020) 4372.

[47] D. Borra, E. Magosso, M. Castelo-Branco, M. Simoes, A Bayesian-optimized design for an interpretable convolutional neural network to decode and analyze the P300 response in autism, J. Neural Eng. 19 (2022), 046010.

[48] D. Borra, M. Filippini, M. Ursino, P. Fattori, E. Magosso, Motor decoding from the posterior parietal cortex using deep neural networks, J. Neural Eng. 20 (2023) 036016.

[49] Y.-F. Chen, R. Fu, J. Wu, J. Song, R. Ma, Y.-C. Jiang, M. Zhang, Continuous bimanual trajectory decoding of coordinated movement from EEG signals, IEEE J. Biomed. Health Inform. 26 (2022) 6012–6023.

[50] J. Bradbury, S. Merity, C. Xiong, R. Socher, Quasi-recurrent neural networks, 2016, arXiv preprint arXiv:1611.01576 [Cs].

[51] G. Montavon, W. Samek, K.-R. Müller, Methods for interpreting and understanding deep neural networks, Digit. Signal Process. 73 (2018) 1–15.

[52] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, 2014, arXiv preprint arXiv:1312.6034 [Cs].

[53] A. Vahid, M. Mückschel, S. Stober, A.-K. Stock, C. Beste, Applying deep learning to single-trial EEG data provides evidence for complementary theories on action control, Commun. Biol. 3 (2020) 112.

[54] R. Kobler, A.I. Sburlea, G. Müller-Putz, A comparison of ocular artifact removal methods for block design based electroencephalography experiments, in: G. Müller-Putz, D. Steyrl, S. Wrissnegger, R. Scherer (Eds.), Proceedings of the 7th Graz Brain-Computer Interface Conference 2017, Verlag der Technischen Universität Graz, 2017, pp. 236–241.

[55] M. Ravanelli, Y. Bengio, Speaker recognition from raw waveform with SincNet, in: 2018 IEEE Spoken Language Technology Workshop (SLT), 2018, pp. 1021–1028.

[56] D. Borra, S. Fantozzi, E. Magosso, 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, Springer International Publishing, Cham, 2020, pp. 1113–1122.

[57] D.-A. Clevert, T. Unterthiner, S. Hochreiter, Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs), 2015, arXiv preprint arXiv:1511.07289 [Cs].

[58] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958.

[59] F. Chollet, Xception: deep learning with depthwise separable convolutions, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1800–1807.

[60] S. Ioffe, C. Szegedy, 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, PMLR, Lille, France, 2015, pp. 448–456.

[61] J. Snoek, H. Larochelle, R.P. Adams, Practical bayesian optimization of machine learning algorithms, in: F. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems, vol. 25, Curran Associates, Inc., 2012, pp. 2951–2959.

[62] M.V. Narkhede, P.P. Bartakke, M.S. Sutaone, A review on weight initialization strategies for neural networks, Artif. Intell. Rev. 55 (2022) 291–322.

[63] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning, 2020, arXiv preprint arXiv:1911.02685 [Cs, Stat].

[64] Q. Wang, Y. Ma, K. Zhao, Y. Tian, A comprehensive survey of loss functions in machine learning, Ann. Data. Sci. 9 (2022) 187–212.

[65] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, 2017, arXiv preprint arXiv:1412.6980 [Cs].

[66] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, in: Automatic Differentiation in PyTorch, NIPS-W, 2017.

[67] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PLoS One 10 (2015), e0130140.

[68] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-Cam: Visual explanations from deep networks via gradient-based localization, Int. J. Comput. Vis. 128 (2020) 336–359.

[69] D. Borra, E. Magosso, Deep learning-based EEG analysis: investigating P3 ERP components, J. Integr. Neurosci. 20 (2021) 791–811.

[70] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, J. Am. Stat. Assoc. 32 (1937) 675–701.

[71] W.J. Conover, Practical Nonparametric Statistics, third ed., Wiley, New York, 1999.

[72] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics Bull. 1 (1945) 80.

[73] Y. Benjamini, Y. Hochberg, Controlling the false discovery rate: a practical and powerful approach to multiple testing, J. Roy. Stat. Soc. B 57 (1995) 289–300.

[74] H. Yuan, C. Perdoni, B. He, Relationship between speed and EEG activity during imagined and executed hand movements, J. Neural Eng. 7 (2010), 026001.

[75] F. Filimon, J.D. Nelson, R.-S. Huang, M.I. Sereno, Multiple parietal reach regions in humans: cortical representations for visual and proprioceptive feedback during on-line reaching, J. Neurosci. 29 (2009) 2961–2971.

[76] S. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, 2017, arXiv preprint arXiv:1705.07874 [Cs, Stat].

[77] S.K. Khare, U.R. Acharya, An explainable and interpretable model for attention deficit hyperactivity disorder in children using EEG signals, Comput. Biol. Med. 155 (2023), 106676.

[78] D. Borra, F. Bossi, D. Rivolta, E. Magosso, Deep learning applied to EEG source-data reveals both ventral and dorsal visual stream involvement in holistic processing of social stimuli, Sci. Rep. 13 (2023) 7365.

[79] I. Sturm, S. Lapuschkin, W. Samek, K.-R. Müller, Interpretable deep neural networks for single-trial EEG classification, J. Neurosci. Methods 274 (2016) 141–145.

[80] F.C. Morabito, C. Ieracitano, N. Mammone, An explainable Artificial Intelligence approach to study MCI to AD conversion via HD-EEG processing, Clin. EEG Neurosci. 54 (2023) 51–60.

[81] S. Smith, T. Nichols, Threshold-free cluster enhancement: addressing problems of smoothing, threshold dependence and localisation in cluster inference, Neuroimage 44 (2009) 83–98.

[82] L. McInnes, J. Healy, S. Astels, hdbscan: Hierarchical density based clustering, JOSS 2 (2017) 205.

[83] J. Han, M. Kamber, Data Mining: Concepts and Techniques, third ed., Elsevier, Burlington, MA, 2012.

[84] R.J.G.B. Campello, P. Kröger, J. Sander, A. Zimek, Density-based clustering, WIREs Data Mining Knowl Discov 10 (2020).

[85] H. Cecotti, A. Graser, Convolutional neural networks for P300 detection with application to brain-computer interfaces, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2011) 433–445.