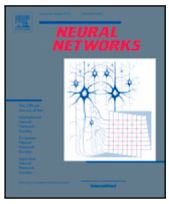
Neural Networks 182 (2025) 106847

Contents lists available at ScienceDirect

Neural Networks

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



Full Length Article

A protocol for trustworthy EEG decoding with neural networks

Davide Borra [a,*,1], Elisa Magosso [a], Mirco Ravanelli [b,c]

[a] Department of Electrical, Electronic and Information Engineering "Guglielmo Marconi" (DEI), University of Bologna, Cesena, Forlì-Cesena, Italy
[b] Department of Computer Science and Software Engineering, Concordia University, Montreal, Quebec, Canada
[c] Mila - Quebec AI Institute, Montreal, Quebec, Canada

ARTICLE INFO

Keywords:
Electroencephalography
Single-trial EEG decoding
Deep learning
Convolutional neural networks
Hyperparameter search
Brain–Computer Interfaces

ABSTRACT

Deep learning solutions have rapidly emerged for EEG decoding, achieving state-of-the-art performance on a variety of decoding tasks. Despite their high performance, existing solutions do not fully address the challenge posed by the introduction of many hyperparameters, defining data pre-processing, network architecture, network training, and data augmentation. Automatic hyperparameter search is rarely performed and limited to network-related hyperparameters. Moreover, pipelines are highly sensitive to performance fluctuations due to random initialization, hindering their reliability. Here, we design a comprehensive protocol for EEG decoding that explores the hyperparameters characterizing the entire pipeline and that includes multi-seed initialization for providing robust performance estimates. Our protocol is validated on 9 datasets about motor imagery, P300, SSVEP, including 204 participants and 26 recording sessions, and on different deep learning models. We accompany our protocol with extensive experiments on the main aspects influencing it, such as the number of participants used for hyperparameter search, the split into sequential simpler searches (multi-step search), the use of informed vs. non-informed search algorithms, and the number of random seeds for obtaining stable performance. The best protocol included 2-step hyperparameter search via an informed search algorithm, with the final training and evaluation performed using 10 random initializations. The optimal trade-off between performance and computational time was achieved by using a subset of 3–5 participants for hyperparameter search. Our protocol consistently outperformed baseline state-of-the-art pipelines, widely across datasets and models, and could represent a standard approach for neuroscientists for decoding EEG in a trustworthy and reliable way.

1. Introduction

Machine learning and deep learning approaches are widely used to process the brain activity, often encompassing data recorded via electroencephalographic (EEG) signals, mainly for designing neural decoders in Brain–Computer Interfaces (BCIs). In a BCI, neural signals are decoded into a desired set of brain states (e.g., event-related responses to visual stimuli) or behavioral states (e.g., hand posture) primarily by addressing classification tasks (Lotte et al., 2018; McFarland, Anderson, Muller, Schlogl, & Krusienski, 2006; Roy et al., 2019). Based on the prediction made by the decoder, an appropriate feedback is provided to the user, for example by enabling the control of an actuator (e.g., a robotic arm or a neuroprosthesis) or by providing an alternative way of communication in patients unable to communicate (Millán, 2010; Wolpaw, Birbaumer, McFarland, Pfurtscheller, & Vaughan, 2002). This way, a direct link between the user's brain and the environment is achieved, without the involvement of peripheral nerves or muscles. The main BCI paradigms are based on motor imagery (MI), P300, and steady-state visual evoked potential (SSVEP) (Abiri, Borhani, Sellers, Jiang, & Zhao, 2019).

Deep neural networks achieved state-of-the-art decoding performance compared to traditional machine learning for solving neural decoding problems (Roy et al., 2019). Among deep neural networks, convolutional neural networks (CNNs) are the most used ones (see Hossain, Islam, Hossain, Nijholt, and Ahad (2023), Roy et al. (2019) for a review) outperforming other solutions – e.g., recurrent neural networks, hybrid recurrent-convolutional networks, and multi-layer perceptrons – when decoding EEG for BCI purposes, also winning international scientific competitions (An, Chen, & Wu, 2023; Simões, Borra, Santamaría-Vázquez, Bittencourt-Villalpando, Krzemiński, Miladinović, Schmid, Zhao, Amaral, Direito, Henriques, Carvalho, & Castelo-Branco, 2020). Even though deep neural networks were used for designing accurate decoding pipelines, there are still critical issues that require

* Corresponding author.
E-mail address: davide.borra2@unibo.it (D. Borra).
[1] Work conducted at Mila - Quebec AI Institute.

https://doi.org/10.1016/j.neunet.2024.106847
Received 15 January 2024; Received in revised form 3 October 2024; Accepted 23 October 2024
Available online 2 November 2024

further investigation. Specifically, they introduce many architectural and training hyperparameters (i.e., parameters defining the functional form of the network and of the training process). Moreover, also other hyperparameters influence the overall decoding pipeline, such as the ones defining data pre-processing and data augmentation (i.e., the artificial generation of synthetic EEG signals for augmenting training data) (Lashgari, Liang, & Maoz, 2020).

Most of the deep learning-based decoding pipelines proposed in the state-of-the-art (see Roy et al. (2019) for a review) do not search for the optimal hyperparameters, resulting in the design of sub-optimal neural decoders (Roy et al., 2019). On the other hand, other EEG decoding studies (Borra, Fantozzi, & Magosso, 2021; Chowdhury, Muhammad, & Adeel, 2023; Ma et al., 2021; Schirrmeister, Springenberg, Fiederer, Glasstetter, Eggensperger, Tangermann, Hutter, Burgard, & Ball, 2017; Song, Zheng, Liu, & Gao, 2023) perform only post-hoc hyperparameter evaluations, analyzing the variation on the test set performance when using different hyperparameter values respect to baseline ones. Due to the high computational cost of performing hyperparameter search with deep neural networks (Roy et al., 2019) (especially on datasets with many participants, sessions, and EEG trials), only few studies consider pipelines that include automatic hyperparameter search by properly using a separate validation set (different from the test set) (Borra, Magosso, Castelo-Branco, & Simões, 2022; Borra, Mondini, Magosso, & Müller-Putz, 2023; de Oliveira & Rodrigues, 2023; Olivas-Padilla & Chacon-Murguia, 2019; Roy, Chowdhury, McCreadie, & Prasad, 2020), either using a non-informed search algorithm (e.g., grid search and random search) (de Oliveira & Rodrigues, 2023), or an informed algorithm (e.g., sequential model-based search) (Borra et al., 2022; Borra, Mondini, et al., 2023; Olivas-Padilla & Chacon-Murguia, 2019; Roy et al., 2020). The latter performs hyperparameter optimization by considering past evaluations to propose more promising hyperparameters in subsequent iterations, often resulting in improved performance (Bergstra, Bardenet, Bengio, & Kégl, 2011). Importantly, these prior EEG pipelines (Borra et al., 2022; Borra, Mondini, et al., 2023; de Oliveira & Rodrigues, 2023; Olivas-Padilla & Chacon-Murguia, 2019; Roy et al., 2020) tuned architectural (e.g., number of convolutional kernels) and training hyperparameters (e.g., learning rate) only, neglecting the exploration of optimal data pre-processing hyperparameters (e.g., cut-off frequencies for band-pass filtering) and data augmentation hyperparameters (e.g., parameters defining additive white Gaussian noise (Lashgari et al., 2020)). In these studies the search space was simplified by reducing the number of optimized hyperparameters to only 6, on average. Furthermore, pipelines including hyperparameter search were only applied at most on 2 datasets and for only one BCI paradigm (motor imagery) (de Oliveira & Rodrigues, 2023; Olivas-Padilla & Chacon-Murguia, 2019).

Due to these limitations, there is the urgent need for establishing a comprehensive decoding protocol (or equivalently, decoding workflow) – searching for the optimal hyperparameters defining data pre-processing, network architecture, network training, and data augmentation – and for a wider validation of the decoding protocol on many datasets and on different BCI paradigms.

Moreover, many aspects strongly influencing the decoding protocol are not considered in the literature of EEG decoding (Borra et al., 2021, 2022; Borra, Mondini, et al., 2023; Chowdhury et al., 2023; de Oliveira & Rodrigues, 2023; Ma et al., 2021; Olivas-Padilla & Chacon-Murguia, 2019; Roy et al., 2019, 2020; Schirrmeister et al., 2017; Song et al., 2023). These include: (i) the computational time required for hyperparameter search; (ii) the complexity of the search space, affecting the quality of the searched hyperparameters in case of high-dimensionality (e.g., space defined by more than 10–20 hyperparameters (Moriconi, Deisenroth, & Sesh Kumar, 2020)); (iii) the type of hyperparameter search algorithm (i.e., non-informed vs. informed search); (iv) the fluctuation of the final performance depending on the random initialization of the network (Bouthillier et al., 2021). All these factors hinder a fair comparison between different pipelines (e.g., defined with different models). For example, as regarding the point iv., in Borra et al. (2021, 2022), Chowdhury et al. (2023), de Oliveira and Rodrigues (2023), Ma et al. (2021), Olivas-Padilla and Chacon-Murguia (2019), Roy et al. (2020), Schirrmeister et al. (2017), Song et al. (2023) the models were trained adopting only one random initialization (i.e., setting only one random seed value) and then evaluated. It remains obscure how the performance is affected by using different random seeds for initialization, limiting the robustness of results. Moreover, the results presented in these studies lack in transparency regarding the choice for the random seed, specifically whether results arise from the best seed among a selection of randomly extracted seeds or only from one arbitrary selected seed. Therefore, the scientific community is still awaiting for a common protocol for trustworthy EEG decoding.

This study aims to mitigate these limitations by proposing a comprehensive protocol for decoding EEG signals. A key novel feature of our protocol is the exploration of hyperparameters across the entire decoding pipeline, encompassing data pre-processing, network architecture, network training, and data augmentation. Additionally, it incorporates multi-seed initialization during both the training and evaluation phases of the final model, i.e., the optimal model identified through hyperparameter search. Our protocol is validated on 9 multi-session BCI datasets (Aricò, Aloise, Schettini, Salinari, Mattia, & Cincotti, 2014; Faller, Vidaurre, Solis-Escalante, Neuper, & Scherer, 2012; Hoffmann, Vesin, Ebrahimi, & Diserens, 2008; Korczowski et al., 2019; Lee, Kwon, Kim, Kim, Lee, Williamson, Fazli, & Lee, 2019; Leeb, Lee, Keinrath, Scherer, Bischof, & Pfurtscheller, 2007; Tangermann et al., 2012; Zhou, Wu, Lv, Zhang, & Guo, 2016) publicly available for the offline definition of neural decoders for MI-based, P300-based and SSVEP-based BCIs, resulting into 204 participants recorded over 26 recording sessions, overall. To the best of our knowledge, this is the first attempt of proposing a complete and robust protocol for EEG decoding, validated on a large amount of participants, sessions, and EEG trials. Moreover, we address the main under-investigated aspects of the literature influencing the decoding protocol by performing a battery of dedicated experiments, specifically by:

i. Analyzing how hyperparameter search is influenced by the number of participants used (*number of participants used for searching hyperparameters*), i.e., the best trade-off between decoding performance and computational time.
ii. Splitting hyperparameter search into multiple sequential steps (e.g., 2 steps), each running on a low-dimensional sub-space sampled from the original search space, to analyze whether better results can be obtained on sequential searches performed on smaller sub-spaces (Malu, Dasarathy, & Spanias, 2021) (*multi-step hyperparameter search*).
iii. Studying the effect of using non-informed search (Bergstra & Bengio, 2012) vs. informed search (Bergstra et al., 2011) (*hyperparameter search algorithm*).
iv. Investigating how many times it is necessary to train and evaluate the final decoder using a different seed for random initialization, in order to get stable decoding performance (*variability of decoding performance due to the random initialization of trainable parameters*).

It is worth highlighting that the algorithms employed in our decoding protocol (multi-step hyperparameter search and multi-seed training and evaluation) have never been applied to EEG decoding, as these were previously theorized and applied in machine learning tasks other than EEG decoding (Malu et al., 2021; Ravanelli, Brakel, Omologo, & Bengio, 2018; Ravanelli et al., 2020) (e.g., speech classification). Therefore, even though these algorithms have been already proposed in the literature – applied to signals different from the EEG – their translation to EEG decoding represents an additional contribution of our study. Overall, in this study we do not propose a novel algorithm but rather we delineate an optimized and robust protocol for decoding EEG, exploiting previously validated methodologies. Indeed, designing

a novel processing method (e.g., a new data augmenter) or decoder (e.g., a new neural network) is not the only aspect that influences the quality of EEG decoding, but also the specific decoding protocol – that is, how the fundamental decoding stages are implemented for building the complete decoding pipeline (e.g., data pre-processing, data augmentation, network architecture and network training) – affects the results. Our proposed protocol – being validated on a large amount of EEG signals – could be used for a trustworthy EEG decoding. Additionally, we also provide recommendations for neuroscientists to reliably define, train, and use neural networks with EEG signals, thanks to the extensive battery of experiments conducted on the decoding protocol.

2. Background

2.1. BCI paradigms

The protocol we propose is validated across various BCI paradigms, based on motor imagery, P300, and SSVEP. In MI-based, P300-based, and SSVEP-based BCI paradigms the cue-locked EEG activity is decoded, specifically by analyzing the sensorimotor rhythms, the P300 response, and the SSVEP response, respectively. Each type of BCI paradigm is described in the following.

In MI-based BCIs, the user imagines a movement (Mulder, 2007); the imagination of the movement activates brain areas responsible for generating actual movements (Pfurtscheller & Neuper, 1997). The most used motor imagery paradigm is based on sensorimotor rhythms (Yuan & He, 2014). Here, the imagined movement is defined as the imagination of a kinesthetic movement of large body parts, such as hands, feet, and tongue, which are associated to different modulations of the brain activity. Specifically, these modulations occur in the frequency domain in upper alpha (10–13 Hz) and beta (13–30 Hz) bands – reflected onto event-related desynchronizations/synchronizations – both before and during movement imagination, and are mostly prominent at EEG electrode locations C3 and C4 (approximately above the sensorimotor cortex) (Pfurtscheller & Lopes da Silva, 1999). MI-based BCIs were successfully applied for driving an artificial limb (e.g., prosthetic hand) (Alonso-Valerdi, Salido-Ruiz, & Ramirez-Mendoza, 2015) or for rehabilitation, e.g., for restoring grasp in a patient with tetraplegia (Pfurtscheller, Müller, Pfurtscheller, Gerner, & Rupp, 2003).

Besides BCIs guided by internally elicited changes in the brain activity (as in case of motor imagery), the brain activity can be altered by external stimulations, e.g., mainly visual stimuli (Gao, Wang, Gao, & Hong, 2014) such as a flickering letter appearing on the screen, as exploited in visual P300-based (Fazel-Rezai, Allison, Guger, Sellers, Kleih, & Kübler, 2012) and SSVEP-based (Vialatte, Maurice, Dauwels, & Cichocki, 2010) BCIs.

The visual P300-based BCI represents the most popular EEG-based BCI (Abiri et al., 2019), and was first designed by Farwell and Donchin in 1988 (Farwell & Donchin, 1988), creating the so-called P300 speller (6x6 matrix of letters and numbers). This BCI is based on the detection of the P300 response. This response – first reported in the EEG by Sutton, Braren, Zubin, and John (1965) – is an attention-dependent event-related potential characterized by a positive deflection peaking between 250 and 500 ms after the stimulus onset, and is mostly distributed on the scalp approximately around the midline EEG electrodes (Fz, Cz, and Pz), increasing its strength from the frontal to the parietal sites (Polich, 2007). The P300 response can be elicited in the oddball paradigm (Farwell & Donchin, 1988), in which a sequence of stimuli is presented to the user, differing by their frequency of occurrence. Specifically, an infrequent deviant stimulus is immersed in a sequence of frequent standard stimuli (two-stimuli oddball paradigm), while the user is attending to it (e.g., by counting how many times infrequent stimuli are presented). The infrequent stimulus elicits the P300 response. By exploiting the P300 response, the user can guide a P300 speller, e.g., for patients with amyotrophic lateral sclerosis (Cipresso et al., 2012; Nijboer et al., 2008), and therapeutic BCIs, e.g., for improving social skills in autistic patients (Amaral et al., 2018).

Lastly, in visual SSVEP-based BCIs users attend a blinking or moving stimulus (e.g., provided by a flickering light-emitting diode) at a constant frequency (stimulus frequency), eliciting a SSVEP response. This response was first reported by Adrian and Matthews (1934a, 1934b) in 1934, and consists in a brain response to the flickering stimulus at the same frequency of the stimulus and even at its harmonics, occurring at parietal and occipital EEG electrode sites (Vialatte et al., 2010) (see also Norcia, Appelbaum, Ales, Cottereau, and Rossion (2015) for a review). The presentation of multiple flickering stimuli to the user, each with a different flickering frequency, elicits distinct responses in the EEG. By encoding each stimulus frequency to a specific command, the user can guide external devices (e.g., prosthesis) (Muller-Putz & Pfurtscheller, 2008) and SSVEP spellers (Yin, Zhou, Jiang, Yu, & Hu, 2015).

2.2. Convolutional neural networks for EEG decoding

CNNs gained popularity for decoding multi-variate neural time series for non-invasive recordings (e.g., EEG (Hossain et al., 2023; Roy et al., 2019)) and invasive recordings (e.g., single-neuron recordings (Borra, Filippini, Ursino, Fattori, & Magosso, 2023, 2024; Filippini, Borra, Ursino, Magosso, & Fattori, 2022) and electrocorticography recordings (Angrick et al., 2019; Xie, Schwartz, & Prasad, 2018)). Unlike traditional machine learning pipelines – separating the extraction of hand-crafted features of signals (feature extraction and selection) (McFarland et al., 2006) from classification (Lotte et al., 2018) – CNNs perform end-to-end classification, operating directly on neural time series. These networks automatically learn the most relevant neurophysiological features from the input EEG enabling the discrimination between distinct brain states (Borra, Bossi, Rivolta, & Magosso, 2023; Borra, Fantozzi, & Magosso, 2020b, 2020c; Borra et al., 2021; Borra & Magosso, 2021; Borra et al., 2022; Borra, Mondini, et al., 2023; Lawhern, Solon, Waytowich, Gordon, Hung, & Lance, 2018; Mayor-Torres, Ravanelli, Medina-DeVilliers, Lerner, & Riccardi, 2021; Schirrmeister et al., 2017; Song et al., 2023; Waytowich et al., 2018; Zhao, Tang, Si, & Feng, 2019). With CNNs, the EEG can be provided as input to the decoder as a 2-D matrix, with electrodes along one dimension and time samples along the other, preserving the original EEG representation. Notably, EEG signals can be only slightly pre-processed, mainly performing band-pass filtering, downsampling, and epoching into EEG trials. Moreover, during training, EEG trials can also be augmented for improving the CNN generalization, by artificially generating new examples starting from existing ones (data augmentation) (Lashgari et al., 2020), for example by adding white Gaussian noise (Lashgari et al., 2020) or by randomly combining portions of EEG signals belonging to different trials (Al-Saegh, Dawwd, & Abdul-Jabbar, 2021).

CNNs proved to significantly outperform traditional machine learning, widely across BCI paradigms, for both motor imagery (Lawhern et al., 2018; Schirrmeister et al., 2017), P300 (Borra et al., 2021; Lawhern et al., 2018; Simões et al., 2020), and SSVEP (Kwak, Müller, & Lee, 2017; Nguyen & Chung, 2019; Waytowich et al., 2018; Xu, Tang, Li, Zhang, & Feng, 2023) decoding. CNN architectures for EEG processing commonly involve temporal convolutions at first (learning the optimal filtering in time), and then spatial convolutions (learning the optimal combinations of electrode sites). Optionally, they may further filter the processed signals in time with deep temporal convolutions. Among architectures, EEGNet (Lawhern et al., 2018) – in its original version (Lawhern et al., 2018) and its variants (An et al., 2023; Borra, Bossi, et al., 2023; Borra et al., 2020b, 2020c, 2021; Borra & Magosso, 2021; Borra et al., 2022; Borra, Mondini, et al., 2023; Chen, Teng, Chen, Pan, & Geyer, 2024; Deng, Zhang, Yu, Liu, & Sun, 2021; Huang, Xue, Hu, & Liuli, 2020; Li, Su, Belkacem, Cheng, & Chen, 2022; Riyad, Khalil, & Adib, 2021; Simões et al., 2020; Vahid, Mückschel,

Stober, Stock, & Beste, 2020; Waytowich et al., 2018; Yao, Liu, Deng, Tang, & Yu, 2022) – represents the most used one, providing a good trade-off between model compactness, model size, training time, and decoding accuracy. EEGNet-like architectures reached state-of-the-art performance in various international scientific competitions (An et al., 2023; Simões et al., 2020). Among EEGNet-like variants, recent improvements regarded the adoption of multi-resolution temporal feature learning (Borra et al., 2021; Borra, Mondini, et al., 2023), by means of parallel convolutions, and the increase of the receptive fields in temporal convolutions (Chen et al., 2024), by means of dilated convolutions. In addition to these advancements, recent solutions introduced deeper changes in the network structure for improving the learning of convolutional features. Attention-based transformer models were introduced in CNNs to design convolutional transformers. These networks have the advantage of focusing not only on local features (learned by convolutional filters within local receptive fields), but also on long-term relationships (Ding et al., 2024; Liu et al., 2024; Song et al., 2023; Zhang, Li, Yang, & Han, 2023). Moreover, in addition to improving the temporal feature learning (by capturing also long-range dependencies), the spatial feature learning was recently improved too, by the adoption of convolutional graph neural networks. In these models, the spatial information is transformed from an Euclidean space (EEG signal data) to a non-Euclidean space (graph-structured data), which helps depicting the complex relationships among multiple electrodes (Klepl, Wu, & He, 2024; Tang et al., 2024; Xue, Song, Wu, Cheng, & Pan, 2024).

2.3. Single-trial EEG decoding: concepts and notations

Let us assume that EEG signals are recorded from each participant in different recording sessions (N_{sess} in total). For each participant and session, an EEG dataset is formed by trials obtained by epoching the continuous EEG recording with respect to an event of interest (e.g., the presentation onset of a stimulus in a P300 oddball paradigm, exploited in P300-based BCIs). Each trial is associated to a specific cognitive state (e.g., 'target' response: brain response to target stimuli in oddball paradigms, containing the P300 response) or motor state (e.g., 'left-hand imagined movement' response), among N_c possible states under investigation. By denoting with N_t the number of trials, the participant- and session-specific EEG dataset can be therefore formalized as the set $D^{(p,s)} = \left\{ \left(X_0, y_0\right), \ldots, \left(X_{N_t-1}, y_{N_t-1}\right) \right\}$, for the pth participant and sth session. In $D^{(p,s)}$, $X_i \in \mathbb{R}^{C \times T}$ represents the pre-processed EEG signals of the ith trial ($0 \leq i \leq N_t - 1$), collected from C scalp sites and T time samples, while $y_i \in L = \left\{ l_0, \ldots, l_{N_c-1} \right\}$ represents the label associated to X_i among the N_c possible brain states (e.g., l_0 = left-hand movement imagery).

A classifier f is trained using single-trial EEG signals with the goal of predicting the associated class. The classifier can be implemented using a deep neural network, such as a CNN. Formally, it is a parametric classifier $f\left(X_i; \theta, \phi\right) : \mathbb{R}^{C \times T} \rightarrow L$ parametrized in the trainable parameters (contained in θ) and in the hyperparameters (contained in ϕ). The values of the trainable parameters are set during the training process using a training set, while the values of the hyperparameters are obtained during the hyperparameter search process using a validation set. Finally, a test set is used to assess the performance of the model on previously unseen signals.

Due to high inter-subject EEG variability, classifiers are usually trained in a participant-specific manner to be accurately used in BCIs (Hossain et al., 2023; Roy et al., 2019; Saha et al., 2021; Wolpaw et al., 2002). This is even more relevant when applying BCI systems to patients, e.g., for rehabilitation purposes, in which the characteristics and severity of individual impairment must be considered (Saha et al., 2021). Specifically, a few initial recording sessions are devoted to calibrate (i.e., train) a participant-specific classifier, while in the last recording sessions the classifier is used online in the BCI system, providing real time feedback to the user. To emulate this use case during offline evaluations of neural decoders, for each participant the participant-specific signals recorded during one session are used as test set, and the remaining signals collected in the other sessions are used as training and validation sets. This corresponds to a leave-one-session-out strategy, which can be formalized as using $D_{test} = D^{(p,k)}$ as test set, and the remaining $D_{train,valid} = \cup_{j=0, j \neq k}^{N_{sess}-1} D^{(p,j)}$ as training and validation sets, for each kth held-out session. The training (D_{train}) and validation (D_{valid}) sets are finally designed by partitioning $D_{train,valid}$ with a 80%–20% ratio. This procedure is performed $\forall k \in \left[0, N_{sess} - 1\right]$, resulting formally in a session-level cross-validation scheme, where the number of cross-validation folds is equal to the number of total sessions (N_{sess}).

3. Materials and methods

3.1. Proposed decoding protocol

The two key features of our protocol are:

i. The inclusion of hyperparameter search for all the main aspects characterizing EEG decoding, such as data pre-processing, network architecture, network training, and data augmentation, with the corresponding hyperparameters contained in $\phi_{pre-proc}$, ϕ_{net}, $\phi_{training}$, and ϕ_{augm}, respectively.
ii. The adoption of multi-seed initialization for final training and evaluation, to provide robust estimates of decoding performance.

The protocol includes a within-participant leave-one-session-out training strategy for evaluating offline the decoders (see Section 2.3); this way, we employed a training strategy of neural decoders that is as close as possible to the same performed in online BCI recordings. Fig. 1 reports a high-level scheme of our decoding protocol, together with the main aspects investigated in the performed experiments (marked with red, see Section 3.5 for further details). The decoding protocol was implemented using the PyTorch (Paszke et al., 2019)-based library SpeechBrain (Ravanelli et al., 2021). To ensure the reproducibility of our results and an easy access to our decoding protocol, the codes and trained models are accessible at https://github.com/speechbrain/benchmarks/tree/main/benchmarks/MOABB, within our user-friendly Python library 'SpeechBrain-MOABB' (Borra, Paissan, & Ravanelli, 2024).

3.1.1. Hyperparameter search

Hyperparameter search aims at finding the best hyperparameter values for a decoding pipeline, optimizing the performance on a validation set. As no assumptions about the considered decoding problem are made, hyperparameter search is also referred as 'black-box optimization' (Golovin, Solnik, Moitra, Kochanski, Karro, & Sculley, 2017; Turner et al., 2021). In this study, the tuned hyperparameters were the ones in $\phi \in \boldsymbol{\Phi} = \boldsymbol{\Phi}_{pre-proc} \cup \boldsymbol{\Phi}_{net} \cup \boldsymbol{\Phi}_{training} \cup \boldsymbol{\Phi}_{augm}$, having indicated with $\boldsymbol{\Phi}$ the whole search space, and with $\boldsymbol{\Phi}_{pre-proc}$, $\boldsymbol{\Phi}_{net}$, $\boldsymbol{\Phi}_{training}$, and $\boldsymbol{\Phi}_{augm}$ the search spaces defined by data pre-processing ($\phi_{pre-proc}$), network architecture (ϕ_{net}), network training ($\phi_{training}$), and data augmentation (ϕ_{augm}) hyperparameters, respectively. Thus, we aim to find the optimal $\phi^* = argmin_{\phi \in \boldsymbol{\Phi}}\left(k\left(\phi\right)\right)$, where $k\left(\phi\right)$ denotes the objective score to minimize, as evaluated on the validation set (see Section 3.1.2 for the definition of k). It is worth highlighting that the proposed decoding protocol – searching also for the optimal hyperparameters for handling EEG signals $\phi_{pre-proc}$ and ϕ_{augm} (defining EEG pre-processing and augmentation) – incorporates a hyperparameter optimization strategy that is tailored to EEG characteristics. Indeed, depending on the type (e.g., motor imagery vs. oddball tasks) and properties (e.g., different motor imagery conditions) of the EEG recording paradigm, the data preparation and augmentation may require different settings (e.g., different cutoff frequencies in band-pass filtering) for finalizing the decoding problem with high performance. In other words, even though our protocol leverages on black-box hyperparameter optimization (Golovin et al., 2017; Turner et al., 2021), this is designed in a way that it indirectly considers the EEG characteristics specific of the

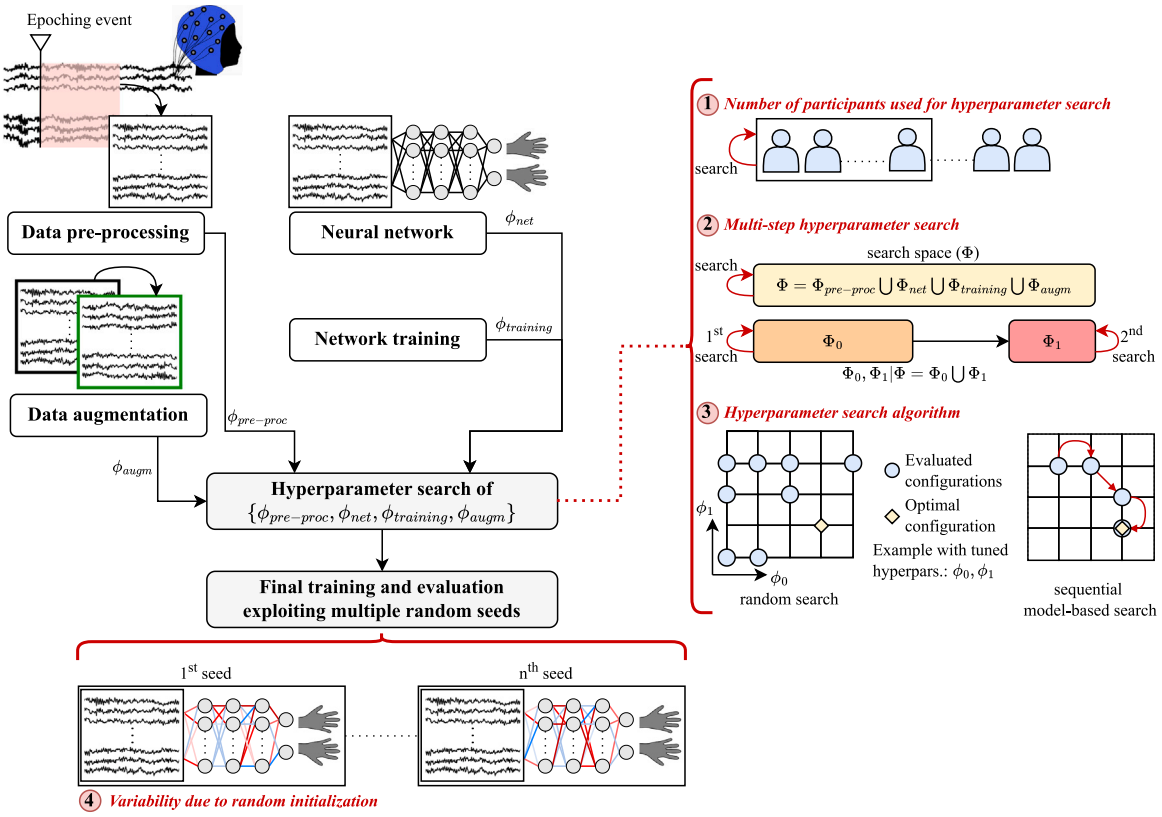


Fig. 1. Scheme of the proposed decoding protocol and of the performed experiments (marked with red).

recorded EEG paradigm, thanks to the inclusion of the hyperparameters of data pre-processing and data augmentation in the overall optimization framework. In our study, the optimization run for 100 iterations. At each iteration, a different hyperparameter configuration is sampled and used to train the decoder, and obtain the objective score k for that configuration. This process is expensive when dealing with deep neural networks, due to the high computational cost of these decoders. Several aspects affect the computational time of hyperparameter search and the quality of the found optimal hyperparameters (in terms of performance). These aspects are presented in the following.

In our case, participant-specific networks were trained using a leave-one-session-out strategy (see Section 2.3). For each participant and each session held out during leave-one-session-out cross-validation, a dedicated network was trained and evaluated. The objective score k was the average objective score across participants and cross-validation folds (i.e., held-out sessions). Thus, after hyperparameter search only one optimal set of hyperparameters ϕ^* was obtained, associated to the configuration that worked best for all the participants' and sessions' signals. By using all participants in the search, the found hyperparameters are the ones that minimize the objective score k on average on all participant-specific networks, potentially leading to a better performance. However, in this scenario, all possible participant-specific networks are trained, thus, requiring a high computational time for hyperparameter search. Addressing the search with a subset of participants could be beneficial for reducing the computational time required for obtaining the optimal hyperparameters. Thus, the number of participants used in hyperparameter search underwent investigation in our experiments (see point i. in Section 3.5).

Common search algorithms exploit a pre-defined rule to sample hyperparameters, e.g., sampling each possible hyperparameter configuration (grid search) or randomly sample a fixed number of hyperparameter configurations (random search (Bergstra & Bengio, 2012)). When dealing with high-dimensional search spaces, random search (Bergstra & Bengio, 2012) is preferable over grid search. This is because the computational budget is pre-defined and does not scale depending on the space dimension, often making it more effective than grid search (Yu & Zhu, 2020). In these approaches, past evaluations are not exploited to sample hyperparameters, that is, no memory is used to guide the search in the most promising hyperparameter sub-space, potentially wasting time on configurations with low validation performance ('non-informed search algorithms'). Sequential model-based search algorithms (Bergstra et al., 2011) address this limitation (often resulting in improved performance (Bergstra et al., 2011)) by maintaining a probabilistic model that incorporates past validation performance ('informed search algorithms'). This model serves as a surrogate for the objective function $k(\phi)$, and it is easier to optimize than the actual objective function: it maps hyperparameters to the probability of obtaining a specific value of k, and can be defined by using tree-structured Parzen estimator (TPE) (Bergstra et al., 2011).

Hyperparameter search can be conducted directly on the entire hyperparameter space $\boldsymbol{\Phi}$ (one-step search) or on hyperparameter sub-spaces (multi-step search). The latter is applied by performing multiple sequential searches (i.e., steps), each step searching for the optimal hyperparameters in a sub-space of the hyperparameter space (e.g., formed by only a subset of hyperparameters). In each step (except the first one), the search starts from the point in the hyperparameter space that resulted optimal in the previous step. Performing sequential searches on low-dimensional sub-spaces of $\boldsymbol{\Phi}$ instead of a single search on $\boldsymbol{\Phi}$, should prevent finding sub-optimal hyperparameter values due to the high-dimensionality of the entire search space $\boldsymbol{\Phi}$ (Malu et al., 2021). Of course, by sequentially searching hyperparameters in N_s sub-spaces of $\boldsymbol{\Phi}$, that is, in $\boldsymbol{\Phi}_k \subset \boldsymbol{\Phi}$, $k = 0, \dots, N_s - 1$, if $\boldsymbol{\Phi} = \boldsymbol{\Phi}_0 \cup \cdots \cup \boldsymbol{\Phi}_{N_s-1}$, after the multi-step search all the hyperparameters are tuned, exploring the entire search space $\boldsymbol{\Phi}$.

Notably, multi-step hyperparameter search (1-step vs. 2-step searches) and the family of hyperparameter search algorithm (non-informed vs. informed search algorithms), both affecting the quality of the found hyperparameters, underwent investigation in our experiments (see points ii. and iii. in Section 3.5).

Table 1
Hyperparameters, search space and sampling probability distributions. The search space is defined by using square brackets for denoting intervals, and curly brackets for denoting a set of admitted values. Uniform probability distributions were used, sampling floating point values (by default, not specified in the table) or integer values ('int' option) or a selection of admitted values ('choice' option).

	Hyperparameter	Search space	Probability distribution
Data pre-processing	Low cut-off frequency (f_0, Hz)	[0.1, 5]	uniform
	High cut-off frequency (f_1, Hz)	[20, 50]	uniform
	Trial upper limit (t_1, s)	[1, 4], if SSVEP or motor imagery, 0.8 if P300	uniform
	Spatial sampling distance (C_{step})	[1, $C_{step,max}$]	uniform, int
Architecture	No. of temporal conv. kernels (K_0)	[4, 64]	uniform, int
	Temporal conv. kernel size (F_0)	(1, [24, 62])	uniform, int
	Spatial conv. depth multiplier (D_1)	[1, 4]	uniform, int
	No. of temporal sep. conv. kernels (K_2)	[1, $2 \cdot K_0 \cdot D_1$]	uniform, int
	Temporal sep. conv. kernel size (F_2)	(1, [3, 24])	uniform, int
	Temporal pooling size (P_2)	(1, [1, 8])	uniform, int
	Dropout probability (p_{drop})	[0, 0.5]	uniform
Training	Learning rate (γ_{max})	{0.01, 0.005, 0.001, 0.0005, 0.0001}	uniform, choice
	Mini-batch size (N_{bs})	{16, 32, 64}	uniform, choice
	No. of epochs (N_{ep})	[250, 1000]	uniform
	No. of averaged models ($N_{ep,avg}$)	[1, 15]	uniform
Data augmentation	Max no. of CutCat segments ($N_{cutcat,max}$)	[2, 6]	uniform, int
	Max amplitude perturbation (δA_{max})	[0, 0.5]	uniform
	Max time shift (δt_{max}, s)	[0, 0.25]	uniform
	White Gaussian noise low-end SNR (SNR_{low}, dB)	[0, 15]	uniform
	White Gaussian noise high-end SNR (SNR_{high}, dB)	SNR_{low}+[5, 20]	uniform

To perform hyperparameter search, the Python library Oríon (Bouthillier et al., 2023) was used. The optimized hyperparameters, the search space, and the probability distributions for hyperparameter sampling during the search are summarized in Table 1. Please note that the tuned hyperparameters are presented in the following sections, describing data pre-processing, network architecture, network training, and data augmentation.

3.1.2. Final training and evaluation

Once hyperparameter search ended, the final training and evaluation was performed, by re-training the optimal model (using the optimal set of hyperparameters contained in ϕ^*) and by evaluating the model on the test set. This was performed multiple times, by using a different random seed for initializing network weights and biases each time (multi-seed training and evaluation), providing a more robust and trustworthy protocol. Indeed, as the networks are initialized randomly, the trained parameters θ^* depend on the initialization point in the parameter space. Changing the random initialization impacts the value of θ^* and consequently, leads to a change in the decoder's performance. The number of runs of final training and evaluation underwent investigation in the conducted experiments (see point iv. in Section 3.5).

During the final evaluation, the performance was assessed by computing the accuracy for motor imagery and SSVEP datasets, and F1 score for P300 datasets. These metrics were chosen as motor imagery and SSVEP datasets were class balanced, while P300 datasets were characterized by a strong class unbalancing (see Section 3.2.1 and Table 2). The same metrics were used to define the objective score minimized during hyperparameter search, that is, $k = 1 - accuracy$ for motor imagery and SSVEP datasets, and $k = 1 - F1$ for P300 datasets (denoting with $F1$ the F1 score).

3.2. Data

3.2.1. Datasets

In this study, we used 9 multi-session MOABB (Jayaram & Barachant, 2018) datasets publicly available, covering:

- Motor imagery-based BCIs: BNCI2014-001 (Tangermann et al., 2012), BNCI2014-004 (Leeb et al., 2007), BNCI2015-001 (Faller et al., 2012), Lee2019-MI (Lee et al., 2019), Zhou2016 (Zhou et al., 2016).
- P300-based BCIs: BNCI2014-009 (Aricò et al., 2014), EPFLP300 (Hoffmann et al., 2008), BI2015a (Korczowski et al., 2019).
- SSVEP-based BCIs: Lee2019-SSVEP (Lee et al., 2019).

For brevity, the main dataset properties are summarized in Tables 2 and 3, while a more detailed description can be found in Appendix A. Please refer to the original publications for further details about data recording paradigms and acquisition.

3.2.2. Data pre-processing

EEG signals were pre-processed as follows (see Roy et al. (2019) for a review), in the same way for all datasets:

i. Band-pass filtering. Signals were band-pass filtered within the range $[f_0, f_1]$ Hz, having indicated with f_0 and f_1 the low and high cut-off frequencies, respectively.
ii. Epoching. EEG trials were obtained by epoching the continuous EEG signals within the range $[t_0 = 0, t_1]$ s, having indicated with t_0 the lower limit of the trial (here corresponding to the onset of the reference event used during epoching), and with t_1 the upper limit of the trial. Note that $t_1 - t_0$ cannot exceed the maximum trial length reported in Table 2. The reference event was the movement imagery onset, the standard/deviant stimulus onset, and the SSVEP stimulus onset, respectively for the motor imagery, P300, and SSVEP recording paradigms. Therefore, after this procedure $T = (t_1 - t_0) \cdot f_s = t_1 \cdot f_s$ samples are included in X_i, having indicated with f_s the sampling frequency.
iii. Electrode set selection (i.e., spatial sampling). We performed a spatial sampling procedure to select the set of C channels of X_i from the entire set of channels available (see Table 2). Specifically, by starting from a seed channel (Cz, included in all datasets), the set of C channels included in X_i was formed by the channels adjacent to the seed channel within a distance of C_{step} channels (see Fig. 2a for an example of this procedure). $C_{step} \in [1, C_{step,max}]$, where $C_{step,max}$ denotes the distance value that returns the entire set of channels included in the dataset (see Table 2), and depends on the specific dataset considered (e.g., $C_{step,max} = 3$ for returning all channels in BNCI2014-001 dataset). In other words, after this spatial sampling procedure, the set of considered channels contained a small subset of channels (few neighbors of Cz, $C_{step} = 1$) up to the entire channel set ($C_{step} = C_{step,max}$), depending on the parameter C_{step}.

Table 2
Datasets: general properties. Signal characteristics (e.g., the maximum trial length, the sampling frequency, and the applied band-pass filtering) are defined by the authors releasing each dataset. Note that in case of P300 datasets, the class unbalance ratio corresponds to standard:target, see Appendix A.

Dataset	No. of participants	No. of sessions	No. of channels	Max. trial length (s)	Sampling freq. (Hz)	Band-pass filter (Hz)	No. of classes	Class unbalance
BNCI2014-001	9	2	22	4	250	0.5–100	4	No
BNCI2014-004	9	5	3	4.5	250	0.5–100	2	No
BNCI2015-001	12	2	13	5	512	0.5–100	2	No
Lee2019-MI	54	2	62	4	1000	None	2	No
Zhou2016	4	3	14	5	250	0.1–100	3	No
BNCI2014-009	10	3	16	0.8	256	0.1–20	2	5:1
EPFLP300	9	4	32	1	2048	None	2	5:1
BI2015a	43	3	32	1	512	None	2	5:1
Lee2019-SSVEP	54	2	62	4	1000	None	4	No

Table 3
Datasets: number of trials. For each dataset, this table lists the number of trials recorded during each recording session (on average across participants) and number of trials used in the performed leave-one-session-out experiments (see Section 2.3) for the training, validation and test sets (on average across participants and cross-validation folds).

Dataset	No. of trials					No. of training trials	No. of valid. trials	No. of test trials
	Session 1	*Session 2*	*Session 3*	*Session 4*	*Session 5*			
BNCI2014-001	288	288	–	–	–	232	56	288
BNCI2014-004	124.4	124.4	160	155.6	160	463.6	115.9	144.9
BNCI2015-001	200	200	–	–	–	160	40	200
Lee2019-MI	100	100	–	–	–	80	20	100
Zhou2016	153.5	146.5	150	–	–	240	60	150
BNCI2014-009	576	576	576	–	–	922	230	576
EPFLP300	813.5	827.9	826.4	821.8	–	1974	493.1	822.4
BI2015a	558	561.8	542.6	–	–	886.7	221.6	554.1
Lee2019-SSVEP	100	100	–	–	–	80	20	100

iv. Downsampling. To reduce the computational cost, signals were downsampled at 125 or 128 Hz, for datasets with sampling frequencies divisible by 5 or 2, respectively (see Table 2).

Overall, the performed EEG pre-processing affected the input EEG in the frequency, temporal, and spatial domains and was parametrized in the hyperparameters $\phi_{pre-proc} = \{f_0, f_1, t_1, C_{step}\}$. These parametrized steps are schematized in Fig. 2a for one representative dataset (pre-processing hyperparameters are marked with red in the figure). These were optimized together with other hyperparameters, enabling to find the optimal data preparation when decoding the EEG.

Of course, also other EEG pre-processing steps may be applied to enhance the task-related brain activity (Kim, 2018). However, when moving from EEG analysis to EEG decoding via deep neural networks, simpler pre-processing pipelines are exploited (Borra, Bossi, et al., 2023; Borra, Fantozzi, & Magosso, 2020a; Borra et al., 2020b, 2020c, 2021; Borra & Magosso, 2021; Borra et al., 2022; Borra, Mondini, et al., 2023; Farahat, Reichert, Sweeney-Reed, & Hinrichs, 2019; Lawhern et al., 2018; Schirrmeister et al., 2017; Song et al., 2023; Waytowich et al., 2018; Zhao et al., 2019), as the one used in our study. Our pre-processing pipeline takes inspiration from prior studies on deep learning-based EEG decoding (Borra, Bossi, et al., 2023; Borra et al., 2020a, 2020b, 2020c, 2021; Borra & Magosso, 2021; Borra et al., 2022; Borra, Mondini, et al., 2023; Farahat et al., 2019; Lawhern et al., 2018; Schirrmeister et al., 2017; Song et al., 2023; Waytowich et al., 2018; Zhao et al., 2019) (see also Roy et al. (2019) for a review). In those studies, the pre-processing is kept as light as possible, leaving the learning system able to automatically extract the most relevant features contained in the input EEG for accurately discriminate the target brain states (e.g., motor imagery conditions). We maintained the same light pre-processing pipeline, but differently from prior studies that used a priori fixed hyperparameters in the pre-processing methods, here we automatically optimized them within our hyperparameter search procedure.

3.2.3. Data augmentation

Each mini-batch of EEG trials provided as input to the neural network was augmented by applying different data augmenters in parallel (see Lashgari et al. (2020) for a review), doubling the number of training trials compared to the values reported in Table 3. This way, $2N_{bs}$ EEG trials were contained in the concatenated mini-batch, having indicated with N_{bs} the mini-batch size defined by the original EEG trials only. The following data augmenters were applied in our protocol, which have been validated for augmenting EEG signals in prior studies (Al-Saegh et al., 2021; George, Smith, Madiraju, Yahyasoltani, & Ahamed, 2022; Lashgari et al., 2020; Mohsenvand, Izadi, & Maes, 2020; Rommel, Paillard, Moreau, & Gramfort, 2022; Sadik, Saraoglu, Canbaz Kabay, Tosun, & Akdag, 2021).

i. CutCat (Al-Saegh et al., 2021; George et al., 2022). CutCat augmentation method consists in generating new EEG trials by cutting existing trials into N_{cutcat} segments and by randomly concatenating together segments belonging to different EEG trials. The number of segments N_{cutcat} was randomly sampled from a uniform distribution in the interval $[2, N_{cutcat,max}]$ for each mini-batch, having indicated with $N_{cutcat,max}$ the maximum number of CutCat segments.
ii. Random amplitude perturbation (Lashgari et al., 2020; Mohsenvand et al., 2020; Sadik et al., 2021). This method consists in applying a perturbation to the amplitude of EEG signals, by increasing or decreasing the amplitude of existing trials (A) by a factor δA. That is, the amplitude of augmented EEG signals (A^*) was $A^* = (1 + \delta A)\, A$. The EEG amplitude increased or decreased when $\delta A > 0$ or $\delta A < 0$, respectively. The amplitude perturbation factor δA was randomly sampled from a uniform distribution in the interval $[-\delta A_{max}, \delta A_{max}]$ for each processed mini-batch, having indicated with δA_{max} the maximum value of δA (in its absolute value).

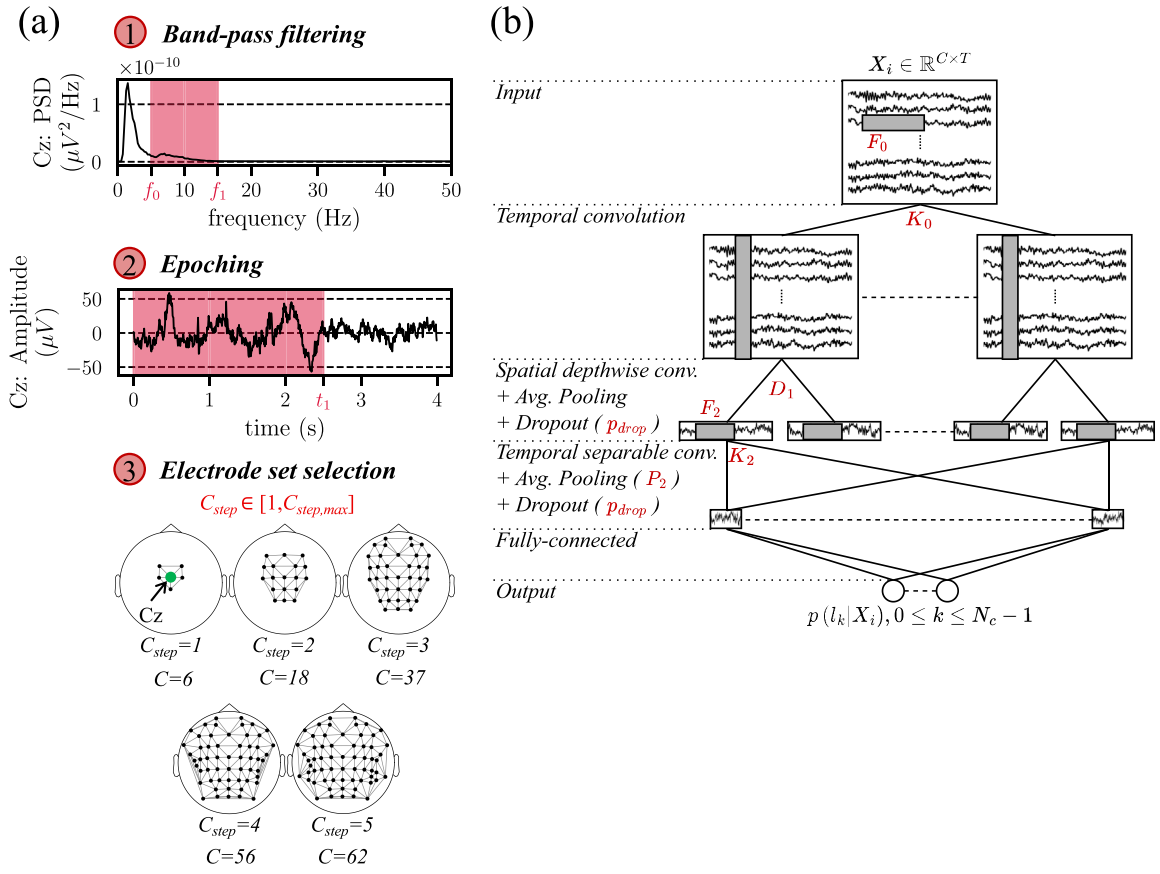


Fig. 2. Panel a — Parametrized data pre-processing. On the top and middle panels, the power spectral density (PSD) and the amplitude over time of Cz are reported, respectively. The bottom panel displays the sets of channels generated by the spatial sampling procedure. For these representations, Lee2019-MI (Lee et al., 2019) dataset was used. The main data pre-processing hyperparameters are displayed in red. Panel b — Network architecture. The main layers are listed on the left. Boxes represent the output feature maps, and gray rectangles represent the convolutional kernels. The main network hyperparameters are displayed in red.

iii. Random time shift (Lashgari et al., 2020; Mohsenvand et al., 2020). This method consists in shifting EEG trials forward or backward in time by a quantity δt. The shift was performed forward in time or backward in time when $\delta t > 0$ or $\delta t < 0$, respectively. The shift factor δt was randomly sampled from a uniform distribution in the interval $\left[-\delta t_{max}, \delta t_{max}\right]$ for each processed mini-batch, having indicated with δt_{max} the maximum value of δt (in its absolute value).

iv. Additive white Gaussian noise (Lashgari et al., 2020; Mohsenvand et al., 2020; Rommel et al., 2022; Sadik et al., 2021). This method inject a white Gaussian noise into the original EEG time series, with an assigned signal-to-noise-ratio (SNR). The SNR was randomly sampled from a uniform distribution in the interval $\left[SNR_{low}, SNR_{high}\right]$ for each processed mini-batch, where SNR_{low} and SNR_{high} denote the low- and high-end of the mixing ratios.

The data augmentation hyperparameters optimized during hyperparameter search were $\phi_{augm} = \left\{N_{cutcat,max}, \delta A_{max}, \delta t_{max}, SNR_{low}, SNR_{high}\right\}$. These were optimized together with the other hyperparameters, enabling to find the optimal data augmentation strategy when decoding EEG.

It is important to underline that using data pre-processing and/or data augmentation pipelines specific of the EEG paradigm (i.e., one different pipeline for each investigated paradigm) was deliberately avoided in this study. Indeed, the aim of the current study is to propose a standardized and versatile protocol for properly defining an entire EEG decoding pipeline based on deep neural networks (from data pre-processing and augmentation to network architecture and training), which can be applied by neuroscientists to a variety of EEG paradigms and EEG decoding problems. Therefore, we did not include components of the decoding pipeline that are a priori tailored to a specific EEG paradigm and decoding problem. From the data handling procedures described in Sections 3.2.2 and 3.2.3, our decoding protocol has the advantage of automatically learning the more appropriate data preparation and augmentation strategy specific of the analyzed EEG dataset – as resulting from our comprehensive hyperparameter search – for accurately decoding the output brain states (e.g., motor imagery conditions). Indeed, it automatically and optimally adapts the data handling procedure to the underlying EEG signal characteristics. Thus, the designed parametrized data pre-processing and data augmentation increase the versatility of our protocol, allowing its use for validating offline EEG decoders in various BCI recording paradigms. In other words, as the proposed hyperparameter search optimizes also the hyperparameters of the parametrized data pre-processing and data augmentation steps, our protocol is able to automatically and optimally prepare the EEG signals, such to enhance the most useful EEG characteristics for solving the specific decoding problem.

3.3. Network architecture

EEGNet (Lawhern et al., 2018) was adopted as neural decoder. A peculiarity of EEGNet is its compactness, and the use of depthwise and

separable convolutions to keep limited the number of trainable parameters. A schematic representation of the network is reported in Fig. 2b. First, the network processes the input neural activity X_i using temporal convolution, to learn how to optimally filter each EEG channel. Here, K_0 temporal kernels with size F_0 are applied, using unitary stride and zero-padding such that the local output shape matches the input shape. Then, spatial convolution is applied separately for each filtered version of the input (spatial depthwise convolution) to learn how to optimally combine the information across the EEG electrode sites. Here, a set of D_1 different spatial kernels with size $F_1 = (C, 1)$ is learned for each filtered version of the input, using unitary stride and no padding. Thus, in total $K_0 \cdot D_1$ spatial kernels are learned. Neurons are activated using a Exponential Linear Unit (ELU) non-linear activation function, and neuron activations are downsampled in time by applying average pooling (pooling size and stride of $P_1 = (1, 4)$). Neuron activations are further processed in time, by using temporal separable convolution. Here, K_2 temporal kernels with size F_2 are applied, using unitary stride and zero-padding such that the local output shape matches the input shape. Neurons are activated using ELU non-linear activation function, and neuron activations are downsampled in time by applying average pooling (pooling size and stride of P_2). All the previous operators (convolution and pooling) are performed in 2-D in the spatio-temporal domain of the input EEG ($X_i \in \mathbb{R}^{C \times T}$). To improve generalization, batch normalization is applied after each convolutional layer, and dropout is included after the two pooling layers, dropping out neurons during training with a probability of p_{drop}. The output from the previous convolutional processing is provided to a fully-connected layer with N_c output neurons, one per class. The output class scores are then converted to conditional probabilities by using the softmax activation function. Therefore, the network output provided the probability that the input EEG trial X_i contained a neural response to a specific brain state (e.g., left-hand imagined movement), i.e., $p\left(l_k | X_i\right), 0 \leq k \leq N_c - 1$.

The network architectural hyperparameters of EEGNet optimized during hyperparameter search were $\phi_{net} = \left\{K_0, F_0, D_1, K_2, F_2, P_2, p_{drop}\right\}$ (marked with red in Fig. 2b).

Crucially, it should be noted that for a more robust validation of the proposed decoding protocol, we applied it also with other CNNs, including one of the first successful CNNs proposed for decoding motor imagery from EEG (ShallowConvNet (Schirrmeister et al., 2017)), and a recent convolutional transformer (EEGConformer (Song et al., 2023)), both applied to BNCI2014-001 dataset (Tangermann et al., 2012) in the original publications. A brief description of ShallowConvNet and EEGConformer is reported in Appendix B (see also Section 3.5).

3.4. Network training

The neural network was randomly initialized, by using a Xavier uniform initialization strategy (Glorot & Bengio, 2010), and biases were initialized to zero. The network was trained using the categorical cross-entropy as loss function $j(\theta)$, by finding the optimal $\theta^* = argmin_\theta (j(\theta))$. Parameters were optimized via mini-batch stochastic gradient descent for N_{ep} epochs, computing gradients using backpropagation and using Adaptive moment estimation (Adam) (Kingma & Ba, 2014) as optimizer, with a learning rate γ and a mini-batch size of N_{bs}. Parameter updates were weighted depending on the class occurrences in the dataset, thus, by weighting more under-represented classes than over-represented classes (crucial for P300 datasets, showing a strong class unbalancing, see Table 2). Cyclic learning rate annealing was performed (Smith, 2015), by changing the learning rate using a triangular function from $\gamma_{min} = 1e-8$ to γ_{max}. The number of training iterations at which the learning rate gradually reached the maximum value (γ_{max}) from the starting value (γ_{min}) was set at 5 times the number of iterations within an epoch, as recommended in Smith (2015). Lastly, once trained the network for N_{ep} epochs, the Polyak averaging (temporal averaging) procedure (Polyak & Juditsky, 1992) was applied. Specifically, weights and biases were averaged across the last $N_{ep,avg}$ epochs to address convergence problems due to the noisy optimization process (characterizing mini-batch stochastic gradient descent), thus, to have a more stable solution.

The network training hyperparameters optimized during hyperparameter search were $\phi_{training} = \left\{\gamma_{max}, N_{bs}, N_{ep}, N_{ep,avg}\right\}$.

3.5. Experiments and statistical analyses

Many aspects of hyperparameter search, final training and evaluation may affect the quality of the decoding on the test set. These aspects are described in the following and were addressed in this study by performing a dedicated battery of experiments for each dataset, aiming at defining a complete and robust protocol for decoding EEG signals.

i. Number of participants used for searching hyperparameters. Here we considered how the decoding performance changes by varying the number of participants used during hyperparameter search. In addition to using all participants for hyperparameter search, we used a subset of 1, 3, 5, 10 participants (5 and 10 participants only for the largest datasets). For each dataset, the performance on the test set (on all participants and leave-one-session-out cross-validation folds) obtained when optimizing hyperparameters using all participants was compared with the one obtained when using less participants. Pairwise Wilcoxon signed-rank tests were performed (Wilcoxon, 1945), and the Benjamini–Hochberg procedure (Benjamini & Hochberg, 1995) was used to correct for multiple tests (30 tests in total).

ii. Multi-step hyperparameter search. Here we tested 1-step search, by running the search on the entire space Φ, vs. 2-step hyperparameter search, by running the first step for 50 iterations on the subset $\Phi_0 = \Phi_{pre-proc} \cup \Phi_{net} \cup \Phi_{training}$, and the second step for other 50 iterations on the subset $\Phi_1 = \Phi_{augm}$. Therefore, in the first step hyperparameters for data pre-processing, network architecture, and network training were tuned while turning off data augmentation; conversely, in the second step data augmentation was turned on and only its hyperparameters were tuned, by keeping fixed the values of the other hyperparameters ($\phi \in \Phi_0$). For each dataset, the performance on the test set (on all participants and leave-one-session-out cross-validation folds) obtained when performing 1-step hyperparameter search was compared with the one obtained with 2-step hyperparameter search. Pairwise Wilcoxon signed-rank tests were performed (Wilcoxon, 1945), and the Benjamini–Hochberg procedure (Benjamini & Hochberg, 1995) was used to correct for multiple tests (9 tests in total).

iii. Hyperparameter search algorithm. Here we compared a non-informed search algorithm (random search) vs. an informed search algorithm (sequential model-based search). Sequential model-based search employed TPE as surrogate model and the search began with 20 iterations of random hyperparameter space sampling (i.e., random search) to initialize the surrogate model. For each dataset, the performance on the test set (on all participants and leave-one-session-out cross-validation folds) obtained with random search was compared with the one obtained with sequential model-based search (TPE-based). Pairwise Wilcoxon signed-rank tests were performed (Wilcoxon, 1945), and the Benjamini–Hochberg procedure (Benjamini & Hochberg, 1995) was used to correct for multiple tests (9 tests in total).

iv. Variability of decoding performance due to the random initialization of trainable parameters. To analyze the performance variability due to random initialization, we performed the final training and evaluation 100 times, each time by randomly initializing networks using a different random seed. Then, a performance variability curve was obtained for each dataset, in the following way. We randomly sampled a fixed number of initializations (ranging from 1 to 20) 50 times out of the total

of 100 initializations. For each dataset, we averaged the performance values across the sampled initializations and computed the standard deviation across the 50 random extractions. The analysis of this curve enabled to understand to which extent the average performance computed across different randomly initialized networks may provide stable decoding results, as a function of the number of initializations used. In particular, we checked when the performance variability curve was less than 1% across datasets. To do so, a non-parametric cluster-level paired t-test (Maris & Oostenveld, 2007) with threshold-free cluster enhancement (Smith & Nichols, 2009) was performed.

Finally, we performed further tests to increase robustness of our protocol validation. Specifically, we compared our protocol applied to EEGNet with a baseline state-of-the-art protocol based on the same decoder. The EEGNet baseline protocol corresponds to the protocol (data pre-processing, network design, network training) used in the reference article by Lawhern et al. (2018), that originally proposed this decoder. For this baseline protocol we used the same hyperparameters adopted by the authors, as resulting from an empirical manual selection of hyperparameters. See Appendix C for further details about the baseline protocol based on EEGNet. For each dataset, the performance on the test set (on all participants and leave-one-session-out cross-validation folds) obtained with the proposed decoding protocol was compared with the one of the baseline protocol. Pairwise Wilcoxon signed-rank tests were performed (Wilcoxon, 1945), and the Benjamini–Hochberg procedure (Benjamini & Hochberg, 1995) was used to correct for multiple tests (9 tests in total). In addition, we applied our protocol to ShallowConvNet (Schirrmeister et al., 2017) and EEGConformer (Song et al., 2023) and the resulting performance distributions were compared to those obtained by the baseline protocols of the same decoders. Here, the baseline protocols were those used in the reference publications originally proposing the decoders, i.e., Schirrmeister et al. (2017) for ShallowConvNet and Song et al. (2023) for EEGConformer, see again Appendix C for baseline protocols of these two decoders.

4. Results

4.1. Number of participants used for searching hyperparameters

Fig. 3 reports the difference between the performance obtained with a subset of participants and with all participants when performing hyperparameter search. Two-step sequential model-based search (TPE-based) using 10 random seeds during final training and evaluation was used for these experiments. Except for Lee2019-MI and Lee2019-SSVEP datasets (the largest datasets in terms of number of total participants) – showing a gradual increase in performance as the number of participants used for searching hyperparameters increases – the other datasets showed approximately a constant performance trend as the number of participants increases. Across all the considered participant subset sizes, using 3 or 5 participants for searching hyperparameters led to models with mostly comparable ($p > 0.05$) or even slightly more accurate decoders (e.g., BNCI2014-001, BNCI2015-001, Lee2019-SSVEP) than using all participants. It is worth highlighting, however, that this hold for 8 out of 9 datasets, as in Lee2019-MI the performance was always significantly lower when using less participants.

Fig. 4 displays the time required for performing hyperparameter search with all participants and with 3 or 5 (Lee2019-MI and Lee2019-SSVEP) participants. The figure reports the wall-clock time needed to complete hyperparameter search on a NVIDIA V100 GPU, providing a quantitative measure of the computational effort of running hyperparameter search in two different setups, differing from the amount of signals exploited during the search. Considering the previous results, using less participants during hyperparameter search not only led to mainly comparable performance for most of the analyzed datasets (8 out of 9 datasets) but also to a significantly lower computational time ($p = 0.0039$, Wilcoxon signed-rank test) for performing the search, i.e., from 321.8 ± 131.0 h (mean $\pm$ standard error of the mean) to 55.9 ± 20.2 h. Of course, this is an expected result, as hyperparameter optimization was run on a subset of participants. However, this information is of relevance for neuroscientists to understand how much the computational time scales down by reducing the number of subjects used in the hyperparameter optimization.

4.2. Multi-step hyperparameter search

Regarding multi-step search investigations, Fig. 5 (left panel) reports the performance difference when using 2-step vs. 1-step hyperparameter search. For these experiments, sequential model-based search (TPE-based) using 10 random seeds during final training and evaluation was used. Two-step hyperparameter search provided significantly higher performance than one-step search for 4 out of 9 datasets (covering MI, P300, and SSVEP paradigms), and comparable performance ($p > 0.05$) for the other datasets. Specifically, an improve in performance of +2.3% (on average across datasets) was observed.

4.3. Hyperparameter search algorithm

Additionally, results on the used search algorithm are reported in Fig. 5 (right panel), displaying the performance difference when using sequential model-based (TPE-based) vs. random search. For these experiments, 2-step searches using 10 random seeds during final training and evaluation were performed. Sequential model-based hyperparameter search led to significantly higher performance than random search for 6 out of 9 datasets (covering MI, P300, and SSVEP paradigms), and comparable performance ($p > 0.05$) for the other datasets. Specifically, an improve in performance of +2.2% (on average across datasets) was observed.

Fig. 6 illustrates a representative example (on BNCI2014-001 dataset) showing the evolution of the validation performance during hyperparameter search (i.e., over the 100 iterations performed). It depicts the dynamics obtained with sequential model-based search (blue line) vs. random search (black line). It turned out that sequential model-based search reached higher validation performance than random search in both steps of the performed 2-step search, with an increase of +4.8% and +2.4% during step 1 and 2, respectively. It is worth noticing that a higher performance was achieved at the end of the second step (devoted to searching data augmentation hyperparameters only) compared to the first step (devoted to searching all other hyperparameters), specifically with an increase of +3.6% in random search, and of +1.2% in sequential model-based search, proving that data augmentation was able to improve the decoder performance.

4.4. Variability of decoding performance due to random initialization

Fig. 7 shows the performance variability curve due to the random initialization of the trainable parameters (top panel), along with the results from the performed statistical analysis (bottom panel). As the number of initializations used to average the performance increases, the variability reduces from approximately 2% (on average across datasets) with only 1 initialization (i.e., without performing any averaging across different initializations) to approximately 0.5% after 10 initializations, remaining stable at this variability level for the following conditions tested (from 10 to 20 initializations). Moreover, after 10 initializations the performance variability was also significantly lower than 1%. It is worth noticing that even with 7 random initializations the variability was at the significance level ($p = 0.05$), but only from 10 to 20 initializations remained stable under the significance level ($p < 0.05$). Among the performance variability curves (blue thin lines) of Fig. 7, few curves emerged with higher variability values, even up to 4.97%. Notably, we observed that the highest performance variability values were obtained with the largest datasets (in terms of number of participants), Lee2019-MI (4.97%) and Lee2019-SSVEP (2.21%), being accompanied by a

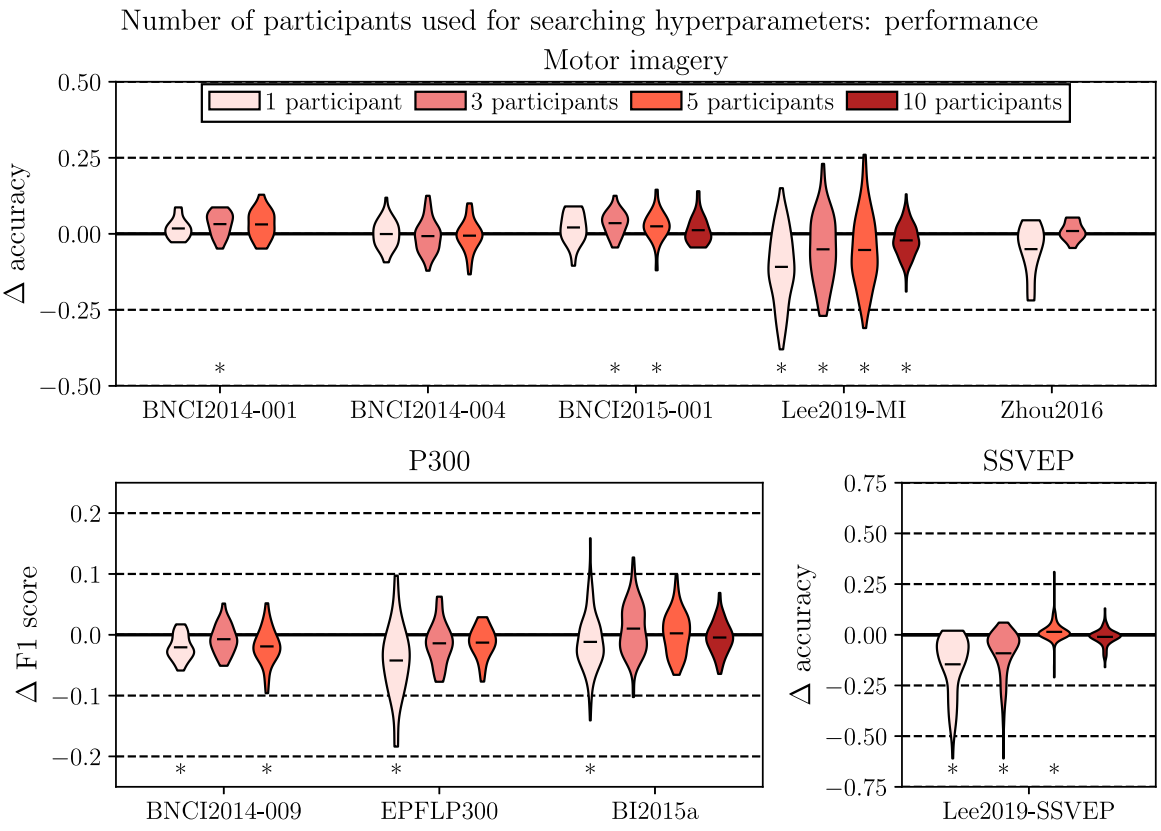


Fig. 3. Number of participants used for searching hyperparameters: performance. For each dataset, each distribution reports the difference between the performance obtained with a subset of participants and with all participants when performing hyperparameter search, as a function of the number of participants used. Performance distributions are displayed as violin plots (horizontal black line: mean value). Results from the statistical analyses are reported too ($*$ $p < 0.05$), see Section 3.5.

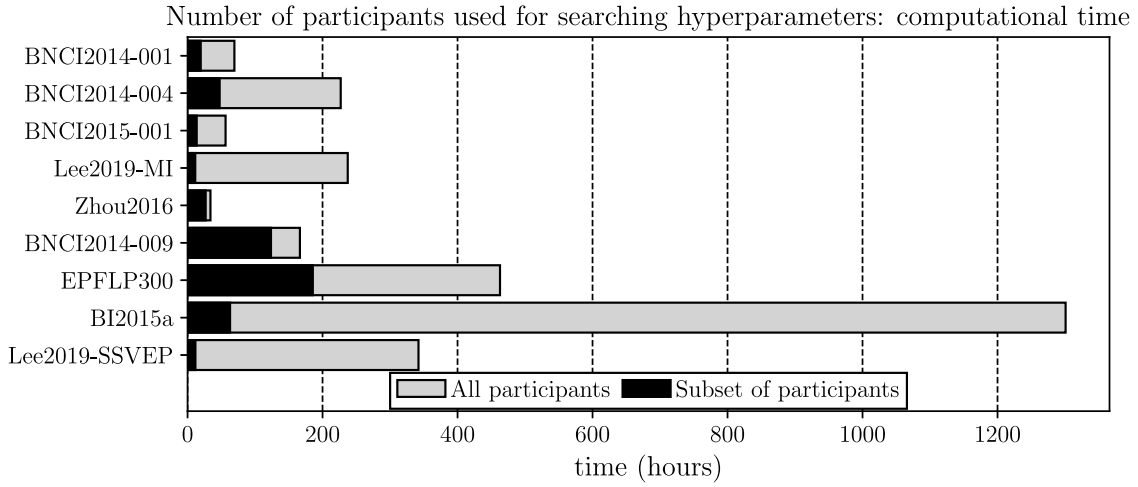


Fig. 4. Number of participants used for searching hyperparameters: computational time. For each dataset, the figure displays the time required for performing the hyperparameter search for the models used in Fig. 3, in case of using signals recorded from all participants (gray bars) or from a subset of participants (black bars). For the latter we considered a subset of 3 participants for all datasets except for the biggest datasets (Lee2019-MI and Lee2019-SSVEP) where 5 participants were used.

high inter-participant variability in the input EEG signals, while other datasets were associated to variability values inferior than 2%.

4.5. Comparison with the state-of-the-art

By leveraging the insights provided by the performed battery of experiments, in the following experiments we set our protocol with 2-step sequential model-based hyperparameter search (TPE-based), with the final training and evaluation performed using 10 random initializations for providing a stable decoding performance result. All participants were used for searching hyperparameters, as from our results using a subset of participants provided only a clear benefit in reducing computational time for all datasets, while the performance was mostly comparable performance compared to using all participants.

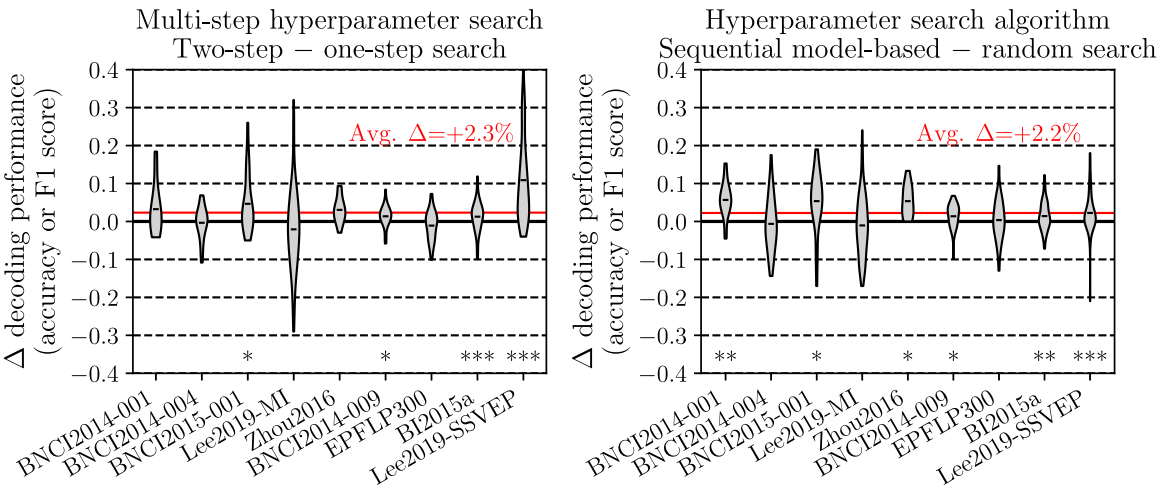


Fig. 5. Multi-step hyperparameter search (left panel) and hyperparameter search algorithm (right panel). The difference between the performance obtained with 2-step and 1-step hyperparameter search is reported in the left panel, and the same between the performance obtained with sequential model-based search (TPE-based) and random search is reported in the right panel, separately for each dataset. Performance distributions are displayed as violin plots (horizontal black line: mean value). Results from the statistical analyses are reported too (∗ $p < 0.05$, ∗∗ $p < 0.01$, ∗∗∗ $p < 0.001$), see Section 3.5. The horizontal red line denotes the average value of the performance difference between the contrasted conditions across all datasets (Δ).

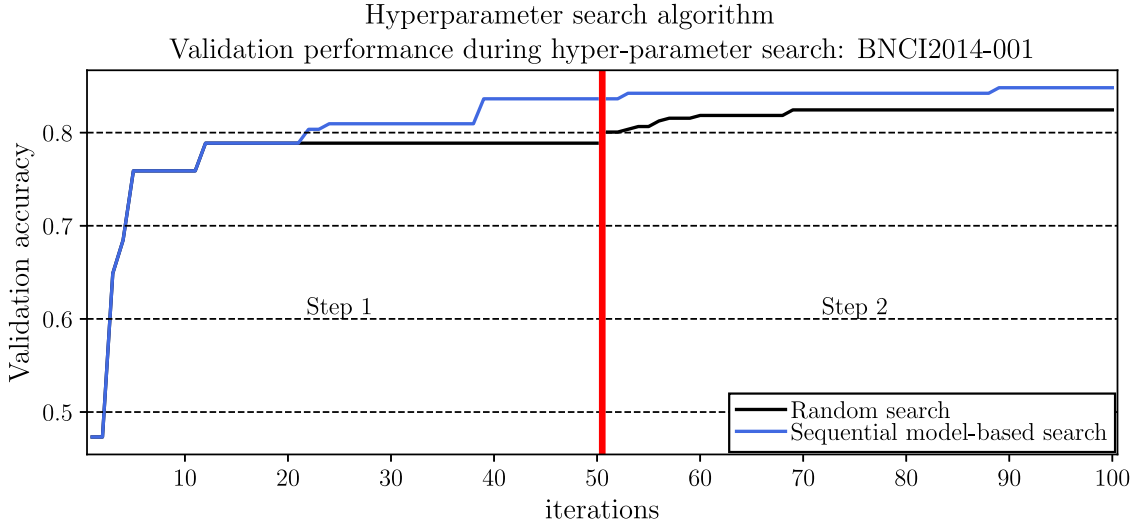


Fig. 6. Representative example of the validation performance dynamic during hyperparameter search: sequential model-based search (TPE-based, blue line) vs. random search (black line). For each iteration, the best validation performance reached up to that iteration is reported. Here, the BNCI2014-001 dataset (Tangermann et al., 2012) used. This example was sampled from the models used in Fig. 5, thus, refer to 2-step searches.

Fig. 8 shows the decoding performance obtained with the proposed protocol and with the baseline state-of-the-art protocol, both based on EEGNet (see Section 3.5 and Appendix C for the description of the baseline protocol based on EEGNet). Notably, our proposed protocol significantly outperformed the baseline protocol for all datasets (9 out of 9 datasets), with performance improvements spanning from +2.9% to +56.6%, on average across participants and leave-one-session-out cross-validation folds. Table 4 reports the main details of the optimal network architectures obtained after hyperparameter search. Specifically, the number of trainable parameters, the model size, the number of multiply–accumulate operations (MACs), and the training time are reported. The latter is expressed as the time to train each participant and each leave-one-session-out cross-validation fold (on average) on a NVIDIA V100 GPU. The optimal values of the searched hyperparameters are reported in Appendix D.

As a final contribution, we further validated our protocol with respect to the state-of-the-art by applying it to other CNNs (ShallowConvNet and EEGConformer). The results reported in Appendix E, showed that also for each of these two decoders, our protocol outperformed the corresponding baseline protocol (see Appendix C for the baseline protocol of ShallowConvNet and EEGConformer). Finally, we summarize in Appendix F the benchmark results obtained in this study by applying our protocol to the different BCI datasets and the different models (EEGNet, ShallowConvNet, and EEGConformer) of high relevance in the scientific community. Please note that the same performance values reported in Appendix F for EEGNet were used to generate Fig. 8.

5. Discussion

5.1. Experiments and suggestions for neuroscientists

From the performed experiment on the number of participants used during hyperparameter search, mostly comparable performance metrics (for 8 out of 9 datasets) were achieved by searching hyperparameters on few participants (3 or 5 participants) with respect to using all the

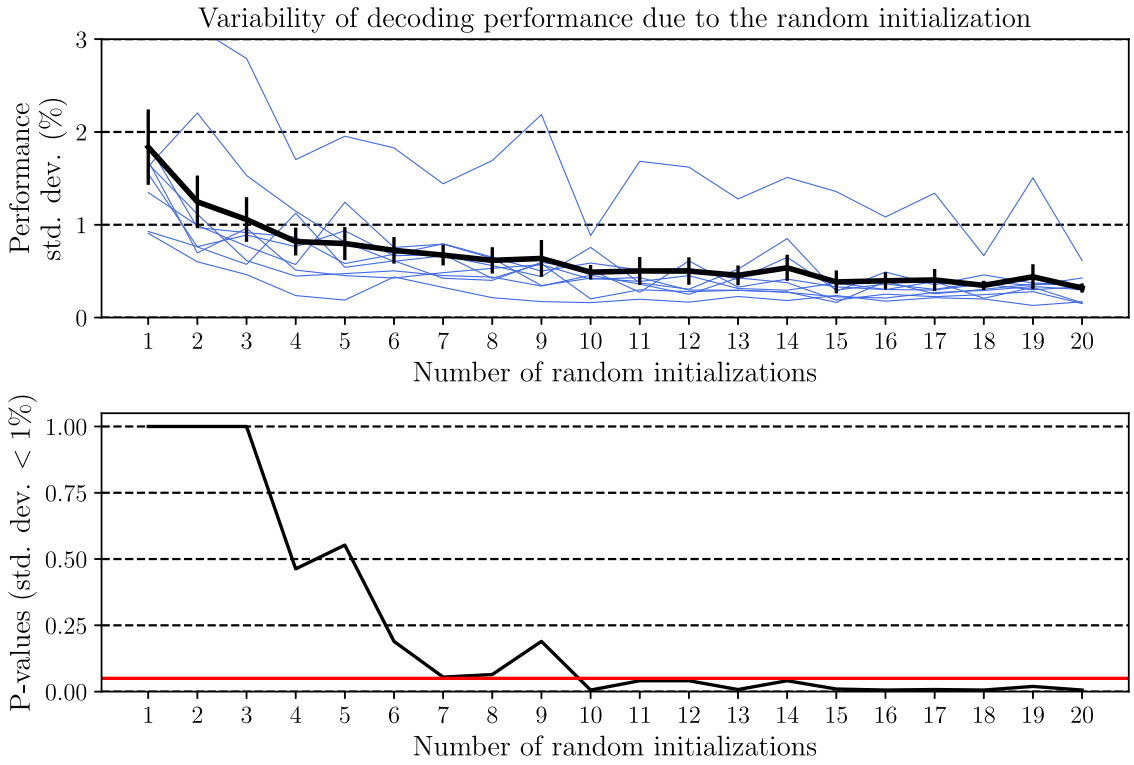


Fig. 7. Variability of decoding performance due to the random initialization of trainable parameters. In the top panel, the performance variability curve is reported for each dataset (thin blue lines), and also averaged across datasets (thick black line, with error bars denoting the standard deviation). In the bottom panel, the p-values obtained while searching for performance variability significantly below 1% are reported (see Section 3.5 for the performed statistical analysis). The horizontal red line denotes the significance level ($p = 0.05$).

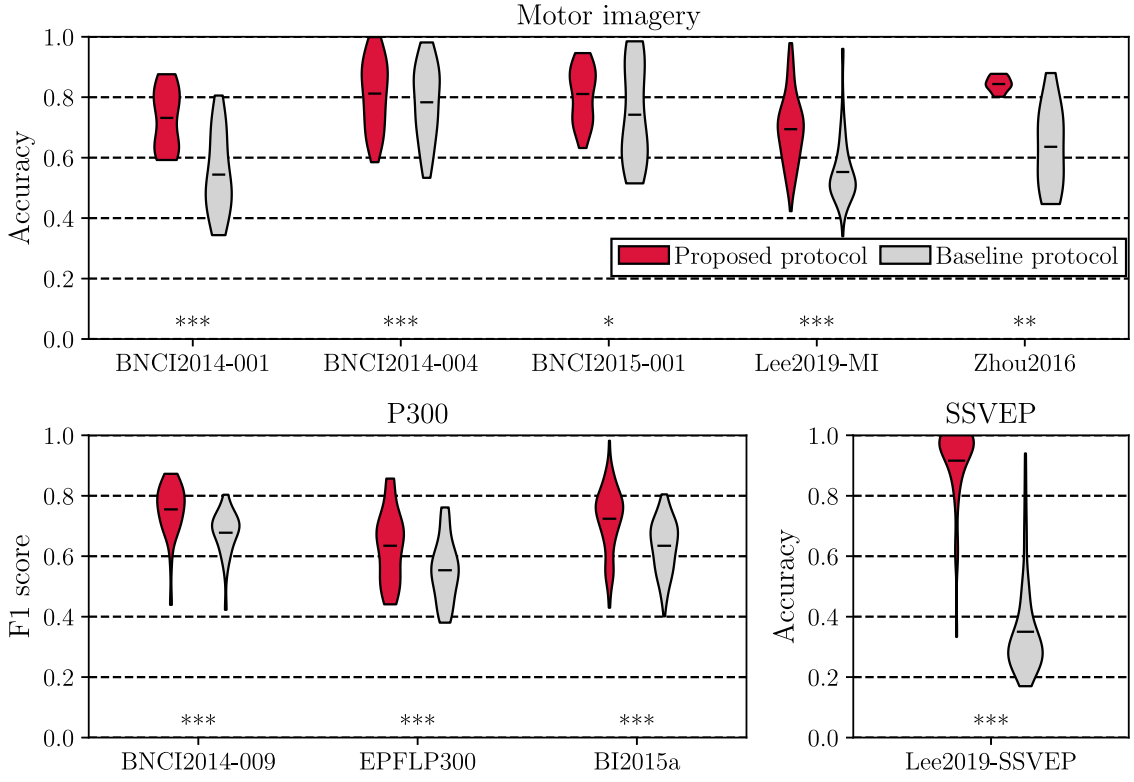


Fig. 8. Comparison with the state-of-the-art. Here, the decoding performance obtained with the proposed decoding protocol was compared with the one of the state-of-the-art baseline protocol, both based on EEGNet (Lawhern et al., 2018). Performance distributions are displayed as violin plots (horizontal black line: mean value). Results from the statistical analyses are reported too (* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$), see Section 3.5.

Table 4
Neural network properties of the optimal architectures as identified during hyperparameter search. The number of trainable parameters, total size in memory (input size of one EEG trial, forward/backward pass size, and trainable parameters size, expressed in MB), multiply–accumulate operations (MACs), and training time are reported.

Dataset	Trainable parameters	Model size (MB)	MACs (M)	Training time (min)
BNCI2014–001	145.9k	11.96	42.06	6.17
BNCI2014–004	9.8k	1.25	2.10	3.27
BNCI2015–001	159.8k	3.77	11.06	1.67
Lee2019-MI	196.4k	5.65	10.45	0.75
Zhou2016	59.1k	7.98	26.98	1.93
BNCI2014–009	49.8k	2.13	5.29	4.31
EPFLP300	13.5k	3.44	6.23	7.67
BI2015a	33.7k	2.50	4.76	4.21
Lee2019-SSVEP	86.2k	12.22	24.96	1.75

available participants. This produced in turn a significant reduction of the computational time required for searching hyperparameters. Therefore, our results suggest that the best trade-off between performance and computational time can be achieved by performing hyperparameter search on a subset of the available dataset. This is crucial as for some datasets (e.g., BI2015a in Fig. 4) the computational time required for hyperparameter search reached more than 1300 h using the signals of all available participants (unfeasible in practical applications), while it was less than 100 h using only a subset of participants. In addition, the gradual increase of the number of participants used for searching hyperparameters (increasing the signal variability in turn) appears to not improve the quality of the searched hyperparameters (in terms of associated performance), see Fig. 3. This was observed especially for the datasets including less participants and more EEG trials in the validation set (see Tables 2 and 3), while for the datasets with the highest number of participants and the lowest number of validation trials (Lee2019-MI and Lee2019-SSVEP), a progressive performance increase was observed with the number of participants used in the hyperparameters search. Indeed, the objective score k (see Section 3.1.1) – being minimized during hyperparameter search – depends on the number of participant-specific networks trained and evaluated during hyperparameter search (i.e., the number of participants used during the search) and on the number of validation EEG trials used. It is fair highlighting however, that due to the high computational cost required by running hyperparameter search also when using a small subset of participants (see Fig. 4), the experiments in this case were performed by sampling the subsets of participants only once. By changing the chosen selection of participants, the found optimal hyperparameters could change too. Therefore, our results using less participants should require further validation in the future, by analyzing the variability of the found hyperparameters across different subsets of participants. Given these results, neuroscientists should use the entire dataset for searching hyperparameters in case enough computational resources are available; however, practitioners are also encouraged to use less participants to speed up this computational demanding process when resources are not available, as from our results only for 1 out of 9 datasets the performance was significantly worsened in this scenario.

Of course, by increasing the number of hyperparameters to search, the complexity of the search space increases too, e.g., including 20 total hyperparameters in our case (see Table 1). As expected, however, splitting the hyperparameter search into two sequential distinct searches on spaces with a lower dimensionality than the entire search space, provided more promising hyperparameter values, which was reflected onto higher decoding performance (Fig. 5 left panel). Moreover, the results obtained by contrasting different search algorithms (Fig. 5 right panel) suggest that an informed algorithm (sequential model-based search), by exploiting past evaluations to suggest new hyperparameter configurations, is able to return more optimal hyperparameter configurations than a non-informed algorithm. Therefore, neuroscientists should consider splitting hyperparameter search into multiple searches when dealing with large hyperparameter spaces (10–20 hyperparameters (Moriconi et al., 2020)), and should keep investigating the use of informed algorithms (sequential model-based search) – as already implemented in Borra et al. (2022), Borra, Mondini, et al. (2023), Olivas-Padilla and Chacon-Murguia (2019), Roy et al. (2020) – without using non-informed algorithms, as done in de Oliveira and Rodrigues (2023).

Another important but under-investigated aspect in the literature regards model variability due to random initializations. Researchers performing simulations on deep neural networks indeed, could get results biased towards a desirable direction (e.g., significant improvements compared to a state-of-the-art algorithm) just by changing the random seed of the random initialization until the desired effect is achieved. Of course, this has to be avoided. One way to solve this issue is to average the performance across different performance evaluations, each obtained once trained the network with a different random initialization, as we performed in this study (multi-seed training and evaluation). Crucially, in the revised literature on deep learning-based EEG decoding (An et al., 2023; Borra, Bossi, et al., 2023; Borra et al., 2020b, 2020c, 2021; Borra & Magosso, 2021; Borra et al., 2022; Borra, Mondini, et al., 2023; Chowdhury et al., 2023; de Oliveira & Rodrigues, 2023; Deng et al., 2021; Huang et al., 2020; Kwak et al., 2017; Lawhern et al., 2018; Li et al., 2022; Ma et al., 2021; Nguyen & Chung, 2019; Olivas-Padilla & Chacon-Murguia, 2019; Riyad et al., 2021; Roy et al., 2020; Schirrmeister et al., 2017; Simões et al., 2020; Song et al., 2023; Vahid et al., 2020; Waytowich et al., 2018; Xu et al., 2023; Yao et al., 2022) this issue was not addressed, and the reported results were obtained only using one random initialization, thus providing a weak performance evaluation and comparison with the state-of-the-art. From our experiments (see Fig. 7), the fluctuations of the decoding performance due to initialization gradually decreased as the number of used initializations increased, starting from approximately 2% (on average, across datasets) of performance variability when using 1 random initialization, to values significantly and permanently below 1% of variability (approximately of 0.5%, on average) from 10 to 20 random initializations. Thus, neuroscientists should train and evaluate neural networks at least 10 times, each time by randomly initializing the network differently (e.g., with a different random seed), and consider the average performance across these networks to properly interpret the results in a robust and trustworthy way.

This battery of experiments enabled to identify a specific configuration for the proposed decoding protocol. The most promising solution indeed, consisted of using 2-step sequential model-based search (TPE-based) exploiting the entire dataset (or equivalently, few participants in case of limited computational resources), and of performing the final training and evaluation using 10 different random initializations. Performance-wise, the so-identified decoding protocol significantly outperformed a baseline state-of-the-art decoding protocol defined on the same neural network (i.e., EEGNet Lawhern et al. (2018)), consistently across datasets. On average, the neural networks defined with the found optimal hyperparameters introduced 48.2k trainable

parameters, and took approximately 3.5 min to train for each participant and cross-validation fold. Crucially, to further validate our decoding protocol, we applied it also to other CNNs (namely ShallowConvNet (Schirrmeister et al., 2017) and EEGConformer (Song et al., 2023)) proposed in the literature for motor imagery decoding, see Appendices B, C and E for further details. Notably, we found a significant improvement in performance when using our decoding protocol also for these other CNNs (+9.9%, +7.3%, on average across participants and cross-validation folds, respectively for ShallowConvNet and EEGConformer) compared with the respective baseline protocols state-of-the-art defined as in Schirrmeister et al. (2017), Song et al. (2023). These results suggest that the proposed approach could also be CNN architecture-independent, and that could be easily used by practitioners in the future also with other CNNs.

5.2. Impact

Our study provides a unique collection of robust and trustworthy EEG decoding results. Indeed, the validation of our protocol and of each aspect that characterizes it have been conducted on 9 datasets comprising 204 participants and 26 recording sessions, spanning across motor imagery, P300 and SSVEP tasks. Remarkably, our experiments have run for a total of 7.61 months (wall-clock time of execution), across all the experiments conducted with different EEG datasets and neural networks.

Neuroscientists could benefit from exploiting our protocol for accurately decoding new datasets, by performing hyperparameter search from scratch on new data. A strength of the proposed protocol is that, even though the hyperparameter search is by construction data-agnostic, in our study it indirectly takes into account the EEG characteristics. Indeed, in our hyperparameter search we parametrized data pre-processing and data augmentation, such that the preparation and the augmentation of EEG signals were defined by parameters (i.e., hyperparameters), for example the cutoff frequencies in band-pass filtering and the range for applying a random amplitude perturbation in the random amplitude data augmenter. Such hyperparameters were optimized jointly with the hyperparameters of the network architecture and training. In this way, the settings of the decoding pipeline steps that handle the input EEG signals (data pre-processing and augmentation) were automatically optimized during the hyperparameter tuning, and therefore, resulted to be tailored to the EEG characteristics.

Alternatively, depending on the addressed decoding problem (e.g., motor imagery decoding), practitioners could leverage the optimal hyperparameters that we found for that problem (among motor imagery, P300, and SSVEP decoding) available in Appendix D, and only perform the final training and evaluation on their own data, to further speed up computations. For example, as concerning data pre-processing for motor imagery applications (see Table D.1) the band-pass filtering could be optimally designed from approximately 1 Hz to 37.5 Hz (on average across datasets), thus including all EEG bands up to low-γ band. This result matches with the known literature on motor correlates, reporting that also EEG frequencies other than α (8–12 Hz) and β (12–30 Hz) (Pfurtscheller & Lopes da Silva, 1999), such as δ (up to 4 Hz) and low-γ (30–50 Hz), encode movement-related information both in motor execution and motor imagery (Borra et al., 2020c; Borra, Mondini, et al., 2023; Kim, Biessmann, & Lee, 2015; Korik, Sosnik, Siddique, & Coyle, 2018; Ofner & Muller-Putz, 2015). Moreover, the optimal channel sets were formed by few channels (from 3 to 18) around Cz covering mainly the sensorimotor cortex, e.g., $C_{step} = 2$ corresponding to using $C = 18$ electrodes out of the total 62 electrodes in Lee2019-MI (Lee et al., 2019) (see Tables 2 and D.1, but also Fig. 2a showing the spatial sampling procedure for Lee2019-MI dataset). Our results could be useful for designing more parsimonious EEG recording setups for BCI paradigms, crucial for reducing EEG preparation times in recording sessions, pointing towards more convenient and compact setups for translating the use of EEG from laboratory to real-life scenarios. Moreover, the results of the optimal data augmentation procedure may provide interesting insights regarding the augmentation method more appropriate for each EEG paradigm. As an example, the detection of the P300 response is known to be highly driven by EEG amplitude-related characteristics (see McFarland et al. (2006), and Lotte et al. (2018) for a review), as the P300 component is manifested as a positive voltage deflection between 250–500 ms after the stimulus onset (see Section 2.1, for further details). For the P300 decoding problem, the hyperparameter defining the optimal maximum amplitude perturbation (δA_{max}) was automatically set at 1.96% on average across P300 datasets (see the optimal hyperparameters reported in Table D.2 in the Appendix D). Therefore, our decoding protocol was automatically able to neglect (almost switch off) the random amplitude perturbation augmentation when addressing a decoding problem mainly driven by amplitude-related features, in order to not disrupt the signal characteristics physiologically meaningful and relevant for the discrimination.

Additionally, when designing new decoding pipelines, neuroscientists could exploit our suggestions about the key aspect affecting a decoding protocol (e.g., the hyperparameter search algorithm to use).

5.3. Future directions

Given the promising results obtained here in the context of EEG decoding for cue-based BCI recordings, in the future our protocol will be extended as follows. First, even though our decoding protocol was investigated in a comprehensive battery of experiments, covering 9 datasets across diverse BCI paradigms, further evidence is needed on other EEG datasets and on other application scenarios. Here we exploited MOABB datasets for validating our decoding protocol, as these are the most used datasets for validating new EEG decoding pipelines (Jayaram & Barachant, 2018), and as we were interested into validating our protocol with easy-to-replicate experiments. Indeed, MOABB datasets cover the main BCI recording paradigms, that is, motor imagery, P300, and SSVEP (Abiri et al., 2019). Additionally, datasets are public, with no need to sign data sharing agreements. This way, not only an easy replication of our approach is possible (via the release of our protocol within the toolkit SpeechBrain-MOABB (Borra, Paissan, & Ravanelli, 2024)), but also an easy replication of the results. However, the adopted datasets might not be fully representative of the addressed BCI paradigms, and larger datasets could be used to increase the robustness of the results. Moreover, further evidence is needed on other recording paradigms, reflecting other application scenarios, for example seizure classification or continuous kinematic decoding (continuous EEG recordings), even acquired in real-world conditions and with mobile EEG devices. Second, the generalization of our protocol across different subjects should also be addressed, validating both the cross-subject knowledge of decoders (i.e., trainable parameters), and the found optimal hyperparameters. As concerning the last aspect, however, we speculate that our protocol, by searching for the hyperparameters that maximize the validation performance on average across subject-specific decoders, was able to identify the hyperparameter configuration that worked best across subjects; however, further investigations are needed to support this hypothesis.

5.4. Concluding remarks

In conclusion, we proposed a comprehensive protocol for decoding EEG signals and we validated it on 9 publicly available multi-session datasets. The protocol was developed in the effort to overcome the limitations of existing state-of-the-art practices, by not only exploring hyperparameters associated with network architecture and training but those characterizing the entire decoding pipeline. Moreover, the incorporation of multi-seed initialization ensures transparency and reliability in decoding results. We conducted a series of experiments to validate the key aspects influencing the proposed protocol. These

included examining the impact of the number of participants used in hyperparameter search, the implementation of multi-step search, the choice between non-informed and informed search algorithms, and the variability in performance resulting from the random initialization of neural networks.

Although more experimental evidence is needed to further support the proposed protocol, it could represent a standardized tool for EEG decoding that achieves state-of-the-art performance in a trustworthy and reliable way. Moreover, the performed experiments can provide useful suggestions to neuroscientists on how to properly set up a deep learning-based decoding protocol for EEG signals, thus providing a common foundation for developing new decoding pipelines.

CRediT authorship contribution statement

Davide Borra: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Elisa Magosso:** Writing – review & editing, Validation, Funding acquisition, Formal analysis. **Mirco Ravanelli:** Writing – review & editing, Software, Resources, Funding acquisition.

Declaration of competing interest

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

Acknowledgments

The authors declare no conflict of interest.

We gratefully acknowledge the support of Digital Research Alliance of Canada (alliancecan.ca) and of Mila - Quebec AI Institute for the use of their clusters while performing the computations. Providers were 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).

This research was co-funded by the Italian Complementary National Plan PNC-I.1 "Research initiatives for innovative technologies and pathways in the health and welfare sector" D.D. 931 of 06/06/2022, "DARE - DigitAl lifelong pRevEntion" initiative, code PNC0000002, CUP: B53C22006450001.

Appendix A. Datasets: additional details

A.1. Motor imagery

i. BNCI2014-001 (Tangermann et al., 2012). This dataset consists of 22-channel EEG recorded from 9 healthy participants across 2 recording sessions. Electrodes were placed according to 10–10 international system at Fz, FC3, FC1, FCz, FC2, FC4, C5, C3, C1, Cz, C2, C4, C6, CP3, CP1, CPz, CP2, CP4, P1, Pz, P2, POz. The task consisted in the imagination of four movements for 4 s: left hand, right hand, feet, and tongue. For each participant and each session, 288 trials were recorded, balanced across classes. EEG signals were sampled at 250 Hz and band-pass filtered between 0.5 and 100 Hz. This dataset is also known as 'dataset IIa' from BCI competition IV (Tangermann et al., 2012).

ii. BNCI2014-004 (Leeb et al., 2007). This dataset consists of 3-channel EEG recorded from 9 healthy participants across 5 recording sessions. Electrodes were placed according to 10–20 international system at C3, Cz, C4. The task consisted in the imagination of two movements for 4.5 s: left hand, and right hand. For each participant and each session, approximately 145 trials were recorded (on average across participants and sessions), balanced across classes. EEG signals were sampled at 250 Hz and band-pass filtered between 0.5 and 100 Hz. This dataset is also known as 'dataset IIb' from BCI competition IV (Tangermann et al., 2012).

iii. BNCI2015-001 (Faller et al., 2012). This dataset consists of 13-channel EEG recorded from 12 healthy participants across 2 recording sessions. Electrodes were placed according to 10–10 international system in correspondence of three Laplacian derivations at C3 (FC3, C5, CP3, C1), Cz (FCz, C1, CPz, C2), and C4 (FC4, C2, CP4, C6). The task consisted in the imagination of two movements for 5 s: right hand, and feet. For each participant and each session, 200 trials were recorded, balanced across classes. EEG signals were sampled at 250 Hz and band-pass filtered between 0.5 and 100 Hz.

iv. Lee2019-MI (Lee et al., 2019). This dataset consists of 62-channel EEG recorded from 54 healthy participants across 2 recording sessions. Electrodes were placed according to 10–10 international system. The task consisted in the imagination of two movements for 4 s: left hand-grasping, and right hand-grasping. For each participant and each session, 100 trials were recorded, balanced across classes. EEG signals were sampled at 1000 Hz.

v. Zhou2016 (Zhou et al., 2016). This dataset consists of 14-channel EEG recorded from 4 healthy participants across 3 recording sessions. Electrodes were placed according to 10–10 international system at Fp1, Fp2, FC3, FCz, FC4, C3, Cz, C4, CP3, CPz, CP4, O1, Oz, O2. The task consisted in the imagination of three movements for 5 s: left hand, right hand, and feet. For each participant and each session, approximately 150 trials were recorded (on average across participants and sessions), balanced across classes. EEG signals were sampled at 250 Hz and band-pass filtered between 0.1 and 100 Hz.

A.2. P300

i. BNCI2014-009 (Aricò et al., 2014). This dataset consists of 16-channel EEG recorded from 10 healthy participants across 3 recording sessions. Electrodes were placed according to 10–10 international system at Fz, FCz, Cz, CPz, Pz, Oz, F3, F4, C3, C4, CP3, CP4, P3, P4, PO7, PO8. During the task, participants used a P300 speller (6x6 matrix, 36 possible characters in total) organized as in Farwell and Donchin (1988). After the presentation of a stimulus (standard or target stimulus), the post-stimulus response was recorded for 0.8 s. For each participant and each session, 576 trials were recorded, with a strong class unbalance between standard and target stimuli (5:1, having indicated standard:target), by construction of the P300 speller paradigm. EEG signals were sampled at 256 Hz and band-pass filtered between 0.1 and 20 Hz.

ii. EPFLP300 (Hoffmann et al., 2008). This dataset consists of 32-channel EEG recorded from 9 participants across 4 recording sessions. Five out of nine participants were disabled (suffering from amyotrophic lateral sclerosis), while the remaining four participants were healthy. Electrodes were placed according to 10–10 international system. During the task, participants used an interface to choose one out of six images. In this six-choice P300 paradigm, the six different images flashed in a random order, eliciting the P300 response on the target image. After the presentation of a stimulus (standard or target stimulus), the post-stimulus response was recorded for 1 s. For each participant and each session, approximately 822 trials were recorded (on average across participants and sessions), with a strong class unbalance between standard and target stimuli (5:1, having indicated standard:target), by construction of the P300 recording paradigm. EEG signals were sampled at 2048 Hz.

iii. BI2015a (Korczowski et al., 2019). This dataset consists of 32-channel EEG recorded from 43 healthy participants across 3 recording sessions. Electrodes were placed according to 10–10 international system. During the task, participants played a videogame (Brain Invaders) by using a P300-based BCI. The interface exploited the oddball paradigm using a grid of 36 symbols (organized in a 6x6 matrix), randomly flashing during the task to elicit the P300 response. For each participant and each session, approximately 554 trials were recorded (on average across participants and sessions), with a strong class unbalance between standard and target stimuli (5:1, having indicated standard:target), by construction of the P300 recording paradigm. EEG signals were sampled at 512 Hz.

A.3. SSVEP

For SSVEP-based BCIs, we considered Lee2019-SSVEP (Lee et al., 2019). This dataset consists of 62-channel EEG recorded from 54 healthy participants across 2 recording sessions. Electrodes were placed according to 10–10 international system. The paradigm was designed following a conventional SSVEP-based control of BCI systems that require four-direction movements. SSVEP stimuli were presented to the user for 4 s at four different positions (down, right, left, up), flickering at four different frequencies: 5.45, 6.67, 8.57, 12 Hz. For each participant and each session, 100 trials were recorded, balanced across classes. EEG signals were sampled at 1000 Hz.

Appendix B. Other state-of-the-art CNN architectures

B.1. ShallowConvNet

ShallowConvNet (Schirrmeister et al., 2017) is one of the first CNN proposed for motor imagery decoding. This network is designed specifically for oscillatory signal classification, by learning features related to log band-power via squaring non-linearity, average pooling, and log non-linearity after convolutions.

First, ShallowConvNet performs temporal convolution on the input X_i using K_0 kernels with size F_0. Then, spatial convolution is applied, learning $K_1 = K_0$ kernels with size $F_1 = (C, 1)$. After this sequence of temporal and spatial convolutions, feature maps are squared, average pooled (using a pooling size P_1 and pooling stride S_1), and then passed through a log non-linearity. To improve generalization, batch normalization and dropout (with a dropout probability of p_{drop}) are applied, respectively after the spatial convolution and after the log non-linearity. Finally, feature maps are provided to a fully-connected layer with N_c output neurons activated via softmax activation function.

The main architectural hyperparameters are $\phi_{net} = \{K_0, F_0, P_1, S_1, p_{drop}\}$.

B.2. EEGConformer

EEGConformer (Song et al., 2023) is a convolutional transformer that consists of three modules: a convolution module, a self-attention module, and a fully-connected module.

Similar to ShallowConvNet, in the convolution module the convolutions are performed first along the temporal dimension, by learning K_0 kernels with size F_0. Then, spatial convolution is applied, learning $K_1 = K_0$ kernels with size $F_1 = (C, 1)$, and feature maps were activated via ELU non-linearity, average pooled (using a pooling size P_1 and pooling stride S_1). Batch normalization and dropout (with a dropout probability of p_{drop}) are applied, respectively after the spatial convolution and after the pooling layer. In the self-attention module, global temporal dependencies are learned from the EEG feature maps by introducing a self-attention mechanism in the network, adopting a multi-head attention strategy. Self-attention computation was repeated N_{depth} times and included N_{heads} heads. Please note that the values indicating the learned convolutional features (i.e., K_0, K_1) by the convolutional module, due to the multi-head nature of the self-attention module, are to be intended as per-head features, thus, the overall number of convolutional features should be multiplied by N_{heads}. Lastly, in the fully-connected module a fully-connected layer with N_c output neurons activated via softmax activation function provided the output probability distribution.

The main architectural hyperparameters are $\phi_{net} = \{K_0, F_0, P_1, S_1, p_{drop}, N_{depth}, N_{heads}\}$.

Appendix C. Baseline state-of-the-art decoding protocol

Baseline protocols were defined by exploiting hyperparameters defining data pre-processing, network architecture and training as in the pipelines adopted in the reference publications that first proposed EEGNet (Lawhern et al., 2018), ShallowConvNet (Schirrmeister et al., 2017), and EEGConformer (Song et al., 2023). These baseline protocols served for performing the validation of our decoding protocol based on the same networks. The baseline state-of-the-art protocols are resumed in the following.

i. Data pre-processing. The pre-processing steps are the same performed in our decoding pipeline (see Section 3.2.2), except for the spatial sampling procedure. Indeed, in the reference publications of the considered CNNs (Lawhern et al., 2018; Schirrmeister et al., 2017; Song et al., 2023), all electrode sites were considered, i.e., no spatial sampling was performed, $C_{step} = C_{step,max}$. The data pre-processing hyperparameter values are listed in Table C.1.
ii. Network architecture. EEGNet hyperparameters were set as $K_0 = 8$, $F_0 = (1, 62)$, $D_1 = 2$, $K_2 = 16$, $F_2 = (1, 16)$, $P_2 = (1, 8)$, $p_{drop} = 0.5$. ShallowConvNet hyperparameters were defined as $K_0 = 40$, $F_0 = (1, 13)$, $P_1 = (1, 36)$, $S_1 = (1, 8)$, $p_{drop} = 0.5$. Lastly, EEGConformer hyperparameters were set as $K_0 = 8$, $F_0 = (1, 13)$, $P_1 = (1, 36)$, $S_1 = (1, 8)$, $N_{depth} = 5$, $N_{heads} = 5$, $p_{drop} = 0.5$. Note that hyperparameters defined in the temporal domain of ShallowConvNet (Schirrmeister et al., 2017) and EEGConformer (Song et al., 2023) (being used for processing EEG sampled at 250 Hz) were halved as in this study we used signals sampled at 125 or 128 Hz, as also done in past studies when adapting CNNs to other sampling frequencies (Borra et al., 2020c; Borra, Mondini, et al., 2023).
iii. Network training. Similar to our decoding pipeline, Adam (Kingma & Ba, 2014) was used as optimizer, with a learning rate γ and a mini-batch size of N_{bs}. At a variance with our protocol, the learning rate was not annealed, and Polyak averaging (Polyak & Juditsky, 1992) was not applied (i.e., $N_{ep,avg} = 1$). Furthermore, early stopping was applied for selecting the number of training epochs N_{ep} by training networks up to 1000 epochs and returning the network associated to the minimum validation loss. The network training hyperparameter values are listed in Table C.1.

Appendix D. Optimal hyperparameters

Tables D.1, D.2, and D.3 report the optimal hyperparameters (contained in $\phi_{pre-proc}$, ϕ_{net}, $\phi_{training}$, and ϕ_{augm}), as obtained with the proposed protocol (only for EEGNet-based protocol, for brevity), respectively for motor imagery, P300, and SSVEP datasets.

Table C.1
Hyperparameters defining data pre-processing and network training used in the decoding pipelines in the reference publications that first proposed EEGNet (Lawhern et al., 2018), ShallowConvNet (Schirrmeister et al., 2017), and EEGConformer (Song et al., 2023).

	Hyperparameter	Baseline values		
		EEGNet	*ShallowConvNet*	*EEGConformer*
Data pre-proc.	Low cut-off frequency (f_0, Hz)	4	0.5	4
	High cut-off frequency (f_1, Hz)	40	38	40
	Trial upper limit (t_1, s)	2.5	4	4
	Spatial sampling distance (C_{step})	$C_{step} = C_{step,max}$ (all channels considered)		
Training	Learning rate (γ)	0.0001	0.0001	0.0002
	Mini-batch size (N_{bs})	64	64	64
	No. of epochs (N_{ep})	Defined via early stopping (over max. 1000 epochs)		
	No. of averaged models ($N_{ep,avg}$)	1	1	1

Table D.1
Optimal hyperparameters as identified during hyperparameter search: motor imagery.

	Hyperparameter	Motor imagery					
		BNCI2014-001	*BNCI2014-004*	*BNCI2015-001*	*Lee2019-MI*	*Zhou2016*	*Average*
Data pre-proc.	Low cut-off frequency (f_0, Hz)	0.13	0.97	3.4	0.23	0.58	1.062
	High cut-off frequency (f_1, Hz)	46.0	25.6	49.6	21.1	45.3	37.5
	Trial upper limit (t_1, s)	4.0	3.0	4.0	3.2	3.7	3.6
	Spatial sampling distance (C_{step})	2 ($C = 17$)	1 ($C = 3$)	3 ($C = 13$)	2 ($C = 18$)	2 ($C = 14$)	2
Architecture	No. of temporal conv. kernels (K_0)	61	30	26	41	61	44
	Temporal conv. kernel size (F_0)	(1, 51)	(1, 42)	(1, 54)	(1, 29)	(1, 58)	(1, 47)
	Spatial conv. depth multiplier (D_1)	4	3	3	2	2	3
	No. of temporal sep. conv. kernels (K_2)	428	46	99	145	215	187
	Temporal sep. conv. kernel size (F_2)	(1, 15)	(1, 24)	(1, 24)	(1, 13)	(1, 19)	(1, 19)
	Temporal pooling size (P_2)	(1, 7)	(1, 5)	(1, 7)	(1, 8)	(1, 3)	(1, 6)
	Dropout (p_{drop})	0.0085	0.3609	0.2184	0.0120	0.3694	0.1938
Training	Learning rate (γ_{max})	0.0001	0.0001	0.001	0.001	0.001	0.0006
	Mini-batch size (N_{bs})	16	16	16	64	16	26
	No. of epochs (N_{ep})	862	796	760	821	741	796
	No. of averaged models ($N_{ep,avg}$)	10	2	12	15	10	10
Data augm.	Max no. of CutCat segments ($N_{cutcat,max}$)	3	5	2	3	5	4
	Max amplitude perturbation (δA_{max})	0.0174	0.4683	0.0577	0.3492	0.4625	0.2710
	Max time shift (δt_{max}, s)	0.01	0.16	0.0	0.03	0.1	0.06
	White Gaussian noise low-end SNR (SNR_{low}, dB)	15.0	12.0	5.4	14.0	14.0	12.1
	White Gaussian noise high-end SNR (SNR_{high}, dB)	34.1	31.9	22.2	21.0	26.1	27.1

Table D.2
Optimal hyperparameters as identified during hyperparameter search: P300.

	Hyperparameter	P300			
		BNCI2014-009	*EPFLP300*	*BI2015a*	*Average*
Data pre-proc.	Low cut-off frequency (f_0, Hz)	0.12	0.17	0.29	0.19
	High cut-off frequency (f_1, Hz)	43.3	41.4	47.4	44.0
	Spatial sampling distance (C_{step})	3 ($C = 16$)	3 ($C = 32$)	3 ($C = 32$)	3
Architecture	No. of temporal conv. kernels (K_0)	58	61	39	53
	Temporal conv. kernel size (F_0)	(1, 42)	(1, 29)	(1, 29)	(1, 33)
	Spatial conv. depth multiplier (D_1)	3	1	3	2
	No. of temporal sep. conv. kernels (K_2)	219	108	206	178
	Temporal sep. conv. kernel size (F_2)	(1, 17)	(1, 24)	(1, 13)	(1, 18)
	Temporal pooling size (P_2)	(1, 4)	(1, 4)	(1, 4)	(1, 4)
	Dropout (p_{drop})	0.3903	0.3831	0.2011	0.3248
Training	Learning rate (γ_{max})	0.0001	0.0001	0.0001	0.0001
	Mini-batch size (N_{bs})	64	16	16	32
	No. of epochs (N_{ep})	894	508	510	637
	No. of averaged models ($N_{ep,avg}$)	11	2	11	8
Data augm.	Max no. of CutCat segments ($N_{cutcat,max}$)	2	2	3	2
	Max amplitude perturbation (δA_{max})	0.0578	0.0004	0.0007	0.0196
	Max time shift (δt_{max}, s)	0.0	0.25	0.01	0.09
	White Gaussian noise low-end SNR (SNR_{low}, dB)	5.4	12.0	9.7	9.0
	White Gaussian noise high-end SNR (SNR_{high}, dB)	22.2	17.1	14.8	18.0

Table D.3
Optimal hyperparameters as identified during hyperparameter search: SSVEP.

	Hyperparameter	SSVEP *Lee2019-SSVEP*
Data pre-proc.	Low cut-off frequency (f_0, Hz)	4.9
	High cut-off frequency (f_1, Hz)	50.0
	Trial upper limit (t_1, s)	2.6
	Spatial sampling distance (C_{step})	5 ($C = 62$)
Architecture	No. of temporal conv. kernels (K_0)	34
	Temporal conv. kernel size (F_0)	(1, 31)
	Spatial conv. depth multiplier (D_1)	3
	No. of temporal sep. conv. kernels (K_2)	180
	Temporal sep. conv. kernel size (F_2)	(1, 15)
	Temporal pooling size (P_2)	(1, 1)
	Dropout (p_{drop})	0.20
Training	Learning rate (γ_{max})	0.005
	Mini-batch size (N_{bs})	32
	No. of epochs (N_{ep})	932
	No. of averaged models ($N_{ep,avg}$)	12
Data augm.	Max no. of CutCat segments ($N_{cutcat,max}$)	3
	Max amplitude perturbation (δA_{max})	0.0043
	Max time shift (δt_{max}, s)	0.01
	White Gaussian noise low-end SNR (SNR_{low}, dB)	10.0
	White Gaussian noise high-end SNR (SNR_{high}, dB)	15.0

Appendix E. Scalability of the proposed decoding protocol using other neural networks

Here we report the results obtained by applying our proposed approach to ShallowConvNet and EEGConformer on the same motor imagery dataset on which these networks were validated (BNCI2014-001 dataset (Tangermann et al., 2012)). The data pre-processing, network training and data augmentation hyperparameters were searched in the same search space indicated in Table 1 of the main text. As concerning the architectural hyperparameters specific of these two other networks, in Table E.1 we report the search space, and the probability distributions used for architectural hyperparameters of both ShallowConvNet and EEGConformer.

Fig. E.1 reports the results obtained with the proposed decoding protocol and with the baseline state-of-the-art protocol (defined as in presented in Appendix C) when using ShallowConvNet (left panel) and EEGConformer (right panel). Our decoding protocol significantly outperformed the baseline ones ($p < 0.05$, Wilcoxon signed-rank tests), with performance improvements of +9.9% and +7.3% (on average across participants and cross-validation folds), respectively for ShallowConvNet and EEGConformer.

Appendix F. Benchmark results

In Table F.1 we report a benchmark conducted with our protocol with EEGNet, ShallowConvNet, and EEGConformer on the datasets considered in this study (9 in total). The protocol configuration used here was the best one, i.e., 2-step hyperparameter search using sequential model-based search, with performance of the final decoder averaged across 10 random seeds (10-seed random initialization). Please note that, for the last two networks the results are provided only for motor imagery as these networks were mainly validated for motor imagery decoding in the literature (Schirrmeister et al., 2017; Song et al., 2023) (indeed, ShallowConvNet (Schirrmeister et al., 2017) design is also specific for sensorimotor rhythm decoding), while EEGNet was applied to general EEG decoding on a variety of BCI paradigms (An et al., 2023; Borra, Bossi, et al., 2023; Borra et al., 2020b, 2020c, 2021; Borra & Magosso, 2021; Borra et al., 2022; Borra, Mondini, et al., 2023; Deng et al., 2021; Huang et al., 2020; Li et al., 2022; Riyad et al., 2021; Simões et al., 2020; Vahid et al., 2020; Waytowich et al., 2018; Yao et al., 2022). On motor imagery applications, the EEGNet-based

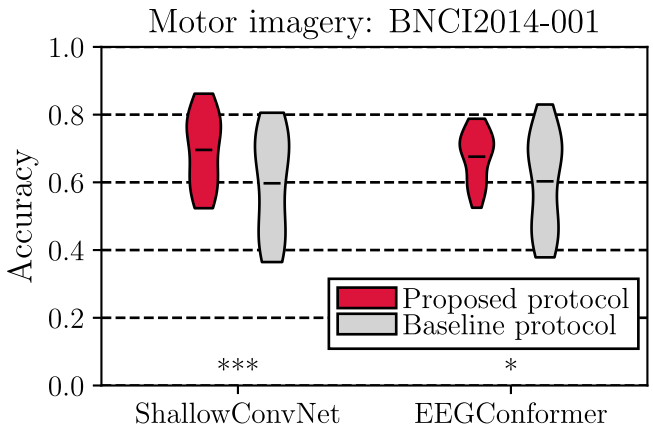


Fig. E.1. Extended comparison with the state-of-the-art. Here, the decoding performance obtained with the proposed decoding protocol is compared with the one of state-of-the-art baseline protocol when classifying motor imagery with BNCI2014-001 (Tangermann et al., 2012). Protocols were based on ShallowConvNet (Schirrmeister et al., 2017) or on EEGConformer (Song et al., 2023) for providing a wider validation also on CNNs different from EEGNet (reported in Fig. 8). The best setup of our proposed decoding protocol consisted of 2-step sequential model-based hyperparameter search (TPE-based) performed on all participants, with 10 random seeds during final training and evaluation. Performance distributions are displayed as violin plots (horizontal black line: mean value). Results from the statistical analyses are reported too (* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$), see Section 3.5.

decoding protocol was the most accurate one, consistently across most of the datasets (for 4 out of 5 motor imagery datasets).

It is fair to note that a previous study on motor imagery decoding (Song et al., 2023) reported that EEGConformer achieves superior performance than EEGNet and ShallowConvNet, contrasting our results. However, the experimental protocol applied by the authors considered only one fixed training-test split of the EEG dataset to estimate decoding performance, did not involve hyperparameter search (i.e., they used one fixed hyperparameter configuration, manually set), and performed network trainings and evaluations with only one random seed. On the other hand, in our robust experimental protocol we employed leave-one-session-out cross-validation (thus, we averaged performance across folds), we searched for the optimal hyperparameters characterizing the entire decoding pipeline, and we trained and evaluated networks 10 times with different random seeds to address the fluctuations of the performance estimates (thus, we averaged performance across the 10 different trained networks). For example, as regarding the fluctuations in performance estimates, by exploiting only 1 random seed the performance variability was approximately 2% on average (with a maximum variability of 4.97%). When performing multi-seed training and evaluation, and considering the average accuracy across different random seeds used during initialization, the performance variability reduces due to the averaging procedure, e.g., by averaging the performance across 10 runs, the variability was consistently below 0.5% (see Fig. 7). Therefore, the eventual differences that can be found with prior literature are primarily due to the different definition of the experimental protocol, here defined with the aim of providing a more trustworthy, transparent and replicable EEG decoding.

Data availability

Public datasets were used for ensuring reproducibility of our results. Links to datasets are provided in the reference publications associated to each dataset.

Table E.1
Hyperparameters, search space and probability distributions used for sampling architectural hyperparameter values for ShallowConvNet and EEGConformer. The same convention used for Table 1 was adopted here.

	Hyperparameter	Search space	Probability distribution
Arch. (ShallowConvNet)	No. of temporal conv. kernels (K_0)	[4, 64]	uniform, int
	Temporal conv. kernel size (F_0)	(1, [5, 62])	uniform, int
	Temporal pooling size (P_1)	(1, [4, 40])	uniform, int
	Temporal pooling stride (S_1)	(1, [4, 40])	uniform, int
	Dropout probability (p_{drop})	[0, 0.5]	uniform
Arch. (EEGConformer)	No. of temporal conv. kernels (K_0)	[4, 16]	uniform, int
	Temporal conv. kernel size (F_0)	(1, [5,62])	uniform, int
	Temporal pooling size (P_1)	(1, [4, 40])	uniform, int
	Temporal pooling stride (S_1)	(1, [4, 40])	uniform, int
	Self-attention depth (N_{depth})	[1, 4]	uniform, int
	Self-attention heads (N_{heads})	[1, 4]	uniform, int
	Dropout probability (p_{drop})	[0, 0.5]	uniform

Table F.1
Benchmark results obtained with the proposed decoding protocol (mean value ± standard deviation across the 10 random seeds). For motor imagery, we provide results obtained with three different CNN architectures. Here, bold values represent the most accurate ones, across architectures. The same distributions summarized here were the ones exploited for designing Fig. 8 and Fig. E.1.

	Dataset	Architecture		
		EEGNet	*ShallowConvNet*	*EEGConformer*
Motor imagery	BNCI2014–001	**0.732 ± 0.004**	0.696 ± 0.004	0.676 ± 0.007
	BNCI2014–004	**0.812 ± 0.002**	0.785 ± 0.003	0.799 ± 0.002
	BNCI2015–001	0.811 ± 0.002	**0.829 ± 0.004**	0.752 ± 0.0096
	Lee2019-MI	**0.694 ± 0.003**	0.658 ± 0.004	0.651 ± 0.008
	Zhou2016	**0.844 ± 0.006**	0.827 ± 0.005	0.840 ± 0.006
P300	BNCI2014–009	0.755 ± 0.002	–	–
	EPFLP300	0.635 ± 0.004	–	–
	BI2015a	0.724 ± 0.002	–	–
SSVEP	Lee2019-SSVEP	0.916 ± 0.002	–	–

References

Abiri, R., Borhani, S., Sellers, E. W., Jiang, Y., & Zhao, X. (2019). A comprehensive review of EEG-based brain–computer interface paradigms. *Journal of Neural Engineering*, *16*(1), Article 011001. http://dx.doi.org/10.1088/1741-2552/aaf12e.

Adrian, E. D., & Matthews, B. H. C. (1934a). The berger rhythm: potential changes from the occipital lobes in man. *Brain*, *57*(4), 355–385. http://dx.doi.org/10.1093/brain/57.4.355.

Adrian, E. D., & Matthews, B. H. C. (1934b). The interpretation of potential waves in the cortex. *The Journal of Physiology*, *81*(4), 440–471. http://dx.doi.org/10.1113/jphysiol.1934.sp003147.

Al-Saegh, A., Dawwd, S. A., & Abdul-Jabbar, J. M. (2021). CutCat: An augmentation method for EEG classification. *Neural Networks*, *141*, 433–443. http://dx.doi.org/10.1016/j.neunet.2021.05.032.

Alonso-Valerdi, L. M., Salido-Ruiz, R. A., & Ramirez-Mendoza, R. A. (2015). Motor imagery based brain–computer interfaces: An emerging technology to rehabilitate motor deficits. *Neuropsychologia*, *79*, 354–363. http://dx.doi.org/10.1016/j.neuropsychologia.2015.09.012.

Amaral, C., Mouga, S., Simões, M., Pereira, H. C., Bernardino, I., Quental, H., et al. (2018). A feasibility clinical trial to improve social attention in autistic spectrum disorder (ASD) using a brain computer interface. *Frontiers in Neuroscience*, *12*, http://dx.doi.org/10.3389/fnins.2018.00477.

An, J., Chen, X., & Wu, D. (2023). Algorithm contest of motor imagery BCI in the world robot contest 2022: A survey. *Brain Science Advances*, *9*(3), 166–181. http://dx.doi.org/10.26599/bsa.2023.9050011.

Angrick, M., Herff, C., Mugler, E., Tate, M. C., Slutzky, M. W., Krusienski, D. J., et al. (2019). Speech synthesis from ECoG using densely connected 3D convolutional neural networks. *Journal of Neural Engineering*, *16*(3), Article 036019. http://dx.doi.org/10.1088/1741-2552/ab0c59.

Aricò, P., Aloise, F., Schettini, F., Salinari, S., Mattia, D., & Cincotti, F. (2014). Influence of P300 latency jitter on event related potential-based brain–computer interface performance. *Journal of Neural Engineering*, *11*(3), Article 035008. http://dx.doi.org/10.1088/1741-2560/11/3/035008.

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *57*(1), 289–300. http://dx.doi.org/10.1111/j.2517-6161.1995.tb02031.x.

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, & K. Weinberger (Eds.), *Advances in neural information processing systems*: *vol. 24*, Curran Associates, Inc., URL https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*, 281–305.

Borra, D., Bossi, F., Rivolta, D., & Magosso, E. (2023). Deep learning applied to EEG source-data reveals both ventral and dorsal visual stream involvement in holistic processing of social stimuli. *Scientific Reports*, *13*(1), http://dx.doi.org/10.1038/s41598-023-34487-z.

Borra, D., Fantozzi, S., & Magosso, E. (2020a). Convolutional neural network for a P300 brain-computer interface to improve social attention in autistic spectrum disorder. In *XV mediterranean conference on medical and biological engineering and computing – MEDICON 2019* (pp. 1837–1843). Springer International Publishing, http://dx.doi.org/10.1007/978-3-030-31635-8_223.

Borra, D., Fantozzi, S., & Magosso, E. (2020b). EEG motor execution decoding via interpretable sinc-convolutional neural networks. In *XV mediterranean conference on medical and biological engineering and computing – MEDICON 2019* (pp. 1113–1122). Springer International Publishing, http://dx.doi.org/10.1007/978-3-030-31635-8_135.

Borra, D., Fantozzi, S., & Magosso, E. (2020c). Interpretable and lightweight convolutional neural network for EEG decoding: Application to movement execution and imagination. *Neural Networks*, *129*, 55–74. http://dx.doi.org/10.1016/j.neunet.2020.05.032.

Borra, D., Fantozzi, S., & Magosso, E. (2021). A lightweight multi-scale convolutional neural network for P300 decoding: Analysis of training strategies and uncovering of network decision. *Frontiers in Human Neuroscience*, *15*, http://dx.doi.org/10.3389/fnhum.2021.655840.

Borra, D., Filippini, M., Ursino, M., Fattori, P., & Magosso, E. (2023). Motor decoding from the posterior parietal cortex using deep neural networks. *Journal of Neural Engineering*, *20*(3), Article 036016. http://dx.doi.org/10.1088/1741-2552/acd1b6.

Borra, D., Filippini, M., Ursino, M., Fattori, P., & Magosso, E. (2024). Convolutional neural networks reveal properties of reach-to-grasp encoding in posterior parietal cortex. *Computers in Biology and Medicine*, *172*, Article 108188. http://dx.doi.org/10.1016/j.compbiomed.2024.108188.

Borra, D., & Magosso, E. (2021). Deep learning-based EEG analysis: investigating P3 ERP components. *Journal of Integrative Neuroscience*, *20*(4), 791–811. http://dx.doi.org/10.31083/j.jin2004083.

Borra, D., Magosso, E., Castelo-Branco, M., & Simões, M. (2022). A Bayesian-optimized design for an interpretable convolutional neural network to decode and analyze the P300 response in autism. *Journal of Neural Engineering*, *19*(4), Article 046010. http://dx.doi.org/10.1088/1741-2552/ac7908.

Borra, D., Mondini, V., Magosso, E., & Müller-Putz, G. R. (2023). Decoding movement kinematics from EEG using an interpretable convolutional neural network. *Computers in Biology and Medicine*, *165*, Article 107323. http://dx.doi.org/10.1016/j.compbiomed.2023.107323.

Borra, D., Paissan, F., & Ravanelli, M. (2024). SpeechBrain-MOABB: An open-source python library for benchmarking deep neural networks applied to EEG signals. *Computers in Biology and Medicine*, *182*, Article 109097. http://dx.doi.org/10.1016/j.compbiomed.2024.109097.

Bouthillier, X., Delaunay, P., Bronzi, M., Trofimov, A., Nichyporuk, B., Szeto, J., et al. (2021). Accounting for variance in machine learning benchmarks. In A. Smola, A. Dimakis, I. Stoica (Eds.), *Proceedings of machine learning and systems*: *vol. 3*, (pp. 747–769). URL https://proceedings.mlsys.org/paper_files/paper/2021/file/0184b0cd3cfb185989f858a1d9f5c1eb-Paper.pdf.

Bouthillier, X., Tsirigotis, C., Corneau-Tremblay, F., Schweizer, T., Dong, L., Delaunay, P., et al. (2023). *Epistimio/orion: Asynchronous distributed hyperparameter optimization*. Zenodo, http://dx.doi.org/10.5281/zenodo.3478592.

Chen, X., Teng, X., Chen, H., Pan, Y., & Geyer, P. (2024). Toward reliable signals decoding for electroencephalogram: A benchmark study to EEGNeX. *Biomedical Signal Processing and Control*, *87*, Article 105475. http://dx.doi.org/10.1016/j.bspc.2023.105475.

Chowdhury, R. R., Muhammad, Y., & Adeel, U. (2023). Enhancing cross-subject motor imagery classification in EEG-based brain–computer interfaces by using multi-branch CNN. *Sensors*, *23*(18), 7908. http://dx.doi.org/10.3390/s23187908.

Cipresso, P., Carelli, L., Solca, F., Meazzi, D., Meriggi, P., Poletti, B., et al. (2012). The use of P300-based BCIs in amyotrophic lateral sclerosis: from augmentative and alternative communication to cognitive assessment. *Brain and Behavior*, *2*(4), 479–498. http://dx.doi.org/10.1002/brb3.57.

de Oliveira, I. H., & Rodrigues, A. C. (2023). Empirical comparison of deep learning methods for EEG decoding. *Frontiers in Neuroscience*, *16*, http://dx.doi.org/10.3389/fnins.2022.1003984.

Deng, X., Zhang, B., Yu, N., Liu, K., & Sun, K. (2021). Advanced TSGL-EEGNet for motor imagery EEG-based brain-computer interfaces. *IEEE Access*, *9*, 25118–25130. http://dx.doi.org/10.1109/access.2021.3056088.

Ding, Y., Li, Y., Sun, H., Liu, R., Tong, C., & Guan, C. (2024). EEG-deformer: A dense convolutional transformer for brain-computer interfaces. http://dx.doi.org/10.48550/ARXIV.2405.00719, URL https://arxiv.org/abs/2405.00719.

Faller, J., Vidaurre, C., Solis-Escalante, T., Neuper, C., & Scherer, R. (2012). Autocalibration and recurrent adaptation: Towards a plug and play online ERD-BCI. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *20*(3), 313–319. http://dx.doi.org/10.1109/tnsre.2012.2189584.

Farahat, A., Reichert, C., Sweeney-Reed, C. M., & Hinrichs, H. (2019). Convolutional neural networks for decoding of covert attention focus and saliency maps for EEG feature visualization. *Journal of Neural Engineering*, *16*(6), Article 066010. http://dx.doi.org/10.1088/1741-2552/ab3bb4.

Farwell, L., & Donchin, E. (1988). Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology*, *70*(6), 510–523. http://dx.doi.org/10.1016/0013-4694(88)90149-6.

Fazel-Rezai, R., Allison, B., Guger, C., Sellers, E., Kleih, S., & Kübler, A. (2012). P300 brain computer interface: current challenges and emerging trends. *Frontiers in Neuroengineering*, *5*, http://dx.doi.org/10.3389/fneng.2012.00014, URL https://www.frontiersin.org/articles/10.3389/fneng.2012.00014.

Filippini, M., Borra, D., Ursino, M., Magosso, E., & Fattori, P. (2022). Decoding sensorimotor information from superior parietal lobule of macaque via convolutional neural networks. *Neural Networks*, *151*, 276–294. http://dx.doi.org/10.1016/j.neunet.2022.03.044.

Gao, S., Wang, Y., Gao, X., & Hong, B. (2014). Visual and auditory brain–computer interfaces. *IEEE Transactions on Biomedical Engineering*, *61*(5), 1436–1447. http://dx.doi.org/10.1109/tbme.2014.2300164.

George, O., Smith, R., Madiraju, P., Yahyasoltani, N., & Ahamed, S. I. (2022). Data augmentation strategies for EEG-based motor imagery decoding. *Heliyon*, *8*(8), Article e10240. http://dx.doi.org/10.1016/j.heliyon.2022.e10240.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In Y. W. Teh, & M. Titterington (Eds.), *Proceedings of machine learning research*: *vol. 9*, *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249–256). Chia Laguna Resort, Sardinia, Italy: PMLR, URL https://proceedings.mlr.press/v9/glorot10a.html.

Golovin, D., Solnik, B., Moitra, S., Kochanski, G., Karro, J., & Sculley, D. (2017). Google vizier: A service for black-box optimization. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1487–1495). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3097983.3098043.

Hoffmann, U., Vesin, J.-M., Ebrahimi, T., & Diserens, K. (2008). An efficient P300-based brain–computer interface for disabled subjects. *Journal of Neuroscience Methods*, *167*(1), 115–125. http://dx.doi.org/10.1016/j.jneumeth.2007.03.005.

Hossain, K. M., Islam, M. A., Hossain, S., Nijholt, A., & Ahad, M. A. R. (2023). Status of deep learning for EEG-based brain–computer interface applications. *Frontiers in Computational Neuroscience*, *16*, http://dx.doi.org/10.3389/fncom.2022.1006763.

Huang, W., Xue, Y., Hu, L., & Liuli, H. (2020). S-EEGNet: Electroencephalogram signal classification based on a separable convolution neural network with bilinear interpolation. *IEEE Access*, *8*, 131636–131646. http://dx.doi.org/10.1109/access.2020.3009665.

Jayaram, V., & Barachant, A. (2018). MOABB: trustworthy algorithm benchmarking for BCIs. *Journal of Neural Engineering*, *15*(6), Article 066011. http://dx.doi.org/10.1088/1741-2552/aadea0.

Kim, S.-P. (2018). Preprocessing of EEG. In *Computational EEG analysis* (pp. 15–33). Springer Singapore, http://dx.doi.org/10.1007/978-981-13-0908-3_2.

Kim, J.-H., Biessmann, F., & Lee, S.-W. (2015). Decoding three-dimensional trajectory of executed and imagined arm movements from electroencephalogram signals. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *23*(5), 867–876. http://dx.doi.org/10.1109/tnsre.2014.2375879.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. http://dx.doi.org/10.48550/ARXIV.1412.6980, URL https://arxiv.org/abs/1412.6980.

Klepl, D., Wu, M., & He, F. (2024). Graph neural network-based EEG classification: A survey. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *32*, 493–503. http://dx.doi.org/10.1109/tnsre.2024.3355750.

Korczowski, L., Cederhout, M., Andreev, A., Cattan, G., Coelho Rodrigues, P. L., Gautheret, V., et al. (2019). *Brain invaders calibration-less P300-based BCI with modulation of flash duration dataset (bi2015a)*. Zenodo, http://dx.doi.org/10.5281/ZENODO.3266929, URL https://zenodo.org/record/3266929.

Korik, A., Sosnik, R., Siddique, N., & Coyle, D. (2018). Decoding imagined 3D hand movement trajectories from EEG: Evidence to support the use of mu, beta, and low Gamma oscillations. *Frontiers in Neuroscience*, *12*, http://dx.doi.org/10.3389/fnins.2018.00130.

Kwak, N.-S., Müller, K.-R., & Lee, S.-W. (2017). A convolutional neural network for steady state visual evoked potential classification under ambulatory environment. In F. Schwenker (Ed.), *PLoS One*, *12*(2), Article e0172578. http://dx.doi.org/10.1371/journal.pone.0172578.

Lashgari, E., Liang, D., & Maoz, U. (2020). Data augmentation for deep-learning-based electroencephalography. *Journal of Neuroscience Methods*, *346*, Article 108885. http://dx.doi.org/10.1016/j.jneumeth.2020.108885.

Lawhern, V. J., Solon, A. J., Waytowich, N. R., Gordon, S. M., Hung, C. P., & Lance, B. J. (2018). EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces. *Journal of Neural Engineering*, *15*(5), Article 056013. http://dx.doi.org/10.1088/1741-2552/aace8c.

Lee, M.-H., Kwon, O.-Y., Kim, Y.-J., Kim, H.-K., Lee, Y.-E., Williamson, J., et al. (2019). EEG dataset and OpenBMI toolbox for three BCI paradigms: an investigation into BCI illiteracy. *GigaScience*, *8*(5), http://dx.doi.org/10.1093/gigascience/giz002.

Leeb, R., Lee, F., Keinrath, C., Scherer, R., Bischof, H., & Pfurtscheller, G. (2007). Brain–computer communication: Motivation, aim, and impact of exploring a virtual apartment. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *15*(4), 473–482. http://dx.doi.org/10.1109/tnsre.2007.906956.

Li, P., Su, J., Belkacem, A. N., Cheng, L., & Chen, C. (2022). Corrigendum: Multi-person feature fusion transfer learning-based convolutional neural network for SSVEP-based collaborative BCI. *Frontiers in Neuroscience*, *16*, http://dx.doi.org/10.3389/fnins.2022.1024150.

Liu, R., Chao, Y., Ma, X., Sha, X., Sun, L., Li, S., et al. (2024). ERTNet: an interpretable transformer-based framework for EEG emotion recognition. *Frontiers in Neuroscience*, *18*, http://dx.doi.org/10.3389/fnins.2024.1320645.

Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., et al. (2018). A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update. *Journal of Neural Engineering*, *15*(3), Article 031005. http://dx.doi.org/10.1088/1741-2552/aab2f2.

Ma, W., Gong, Y., Zhou, G., Liu, Y., Zhang, L., & He, B. (2021). A channel-mixing convolutional neural network for motor imagery EEG decoding and feature visualization. *Biomedical Signal Processing and Control*, *70*, Article 103021. http://dx.doi.org/10.1016/j.bspc.2021.103021.

Malu, M., Dasarathy, G., & Spanias, A. (2021). Bayesian optimization in high-dimensional spaces: A brief survey. In *2021 12th international conference on information, intelligence, systems and applications*. IEEE, http://dx.doi.org/10.1109/iisa52424.2021.9555522.

Maris, E., & Oostenveld, R. (2007). Nonparametric statistical testing of EEG- and MEG-data. *Journal of Neuroscience Methods*, *164*(1), 177–190. http://dx.doi.org/10.1016/j.jneumeth.2007.03.024.

Mayor-Torres, J. M., Ravanelli, M., Medina-DeVilliers, S. E., Lerner, M. D., & Riccardi, G. (2021). Interpretable SincNet-based deep learning for emotion recognition from EEG brain activity. In *2021 43rd annual international conference of the IEEE engineering in medicine and biology society*. IEEE, http://dx.doi.org/10.1109/embc46164.2021.9630427.

McFarland, D., Anderson, C., Muller, K.-R., Schlogl, A., & Krusienski, D. (2006). BCI meeting 2005-workshop on BCI signal processing: feature extraction and translation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *14*(2), 135–138. http://dx.doi.org/10.1109/tnsre.2006.875637.

Millán, J. D. R. (2010). Combining brain-computer interfaces and assistive technologies: state-of-the-art and challenges. *Frontiers in Neuroscience*, *1*, http://dx.doi.org/10.3389/fnins.2010.00161.

Mohsenvand, M. N., Izadi, M. R., & Maes, P. (2020). Contrastive representation learning for electroencephalogram classification. In E. Alsentzer, M. B. A. McDermott, F. Falck, S. K. Sarkar, S. Roy, & S. L. Hyland (Eds.), *Proceedings of machine learning research*: *vol. 136*, *Proceedings of the machine learning for health neurIPS workshop* (pp. 238–253). PMLR, URL https://proceedings.mlr.press/v136/mohsenvand20a.html.

Moriconi, R., Deisenroth, M. P., & Sesh Kumar, K. S. (2020). High-dimensional Bayesian optimization using low-dimensional feature spaces. *Machine Learning*, *109*(9–10), 1925–1943. http://dx.doi.org/10.1007/s10994-020-05899-z.

Mulder, T. (2007). Motor imagery and action observation: cognitive tools for rehabilitation. *Journal of Neural Transmission*, *114*(10), 1265–1278. http://dx.doi.org/10.1007/s00702-007-0763-z.

Muller-Putz, G., & Pfurtscheller, G. (2008). Control of an electrical prosthesis with an SSVEP-based BCI. *IEEE Transactions on Biomedical Engineering*, *55*(1), 361–364. http://dx.doi.org/10.1109/tbme.2007.897815.

Nguyen, T.-H., & Chung, W.-Y. (2019). A single-channel SSVEP-based BCI speller using deep learning. *IEEE Access*, *7*, 1752–1763. http://dx.doi.org/10.1109/access.2018.2886759.

Nijboer, F., Sellers, E., Mellinger, J., Jordan, M., Matuz, T., Furdea, A., et al. (2008). A P300-based brain–computer interface for people with amyotrophic lateral sclerosis. *Clinical Neurophysiology*, *119*(8), 1909–1916. http://dx.doi.org/10.1016/j.clinph.2008.03.034.

Norcia, A. M., Appelbaum, L. G., Ales, J. M., Cottereau, B. R., & Rossion, B. (2015). The steady-state visual evoked potential in vision research: A review. *Journal of Vision*, *15*(6), 4. http://dx.doi.org/10.1167/15.6.4.

Ofner, P., & Muller-Putz, G. R. (2015). Using a noninvasive decoding method to classify rhythmic movement imaginations of the arm in two planes. *IEEE Transactions on Biomedical Engineering*, *62*(3), 972–981. http://dx.doi.org/10.1109/tbme.2014.2377023.

Olivas-Padilla, B. E., & Chacon-Murguia, M. I. (2019). Classification of multiple motor imagery using deep convolutional neural networks and spatial filters. *Applied Soft Computing*, *75*, 461–472. http://dx.doi.org/10.1016/j.asoc.2018.11.031.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. http://dx.doi.org/10.48550/ARXIV.1912.01703, URL https://arxiv.org/abs/1912.01703.

Pfurtscheller, G., & Lopes da Silva, F. (1999). Event-related EEG/MEG synchronization and desynchronization: basic principles. *Clinical Neurophysiology*, *110*(11), 1842–1857. http://dx.doi.org/10.1016/s1388-2457(99)00141-8.

Pfurtscheller, G., Müller, G. R., Pfurtscheller, J., Gerner, H. J., & Rupp, R. (2003). 'Thought' – control of functional electrical stimulation to restore hand grasp in a patient with tetraplegia. *Neuroscience Letters*, *351*(1), 33–36. http://dx.doi.org/10.1016/s0304-3940(03)00947-9.

Pfurtscheller, G., & Neuper, C. (1997). Motor imagery activates primary sensorimotor area in humans. *Neuroscience Letters*, *239*(2–3), 65–68. http://dx.doi.org/10.1016/s0304-3940(97)00889-6.

Polich, J. (2007). Updating P300: An integrative theory of P3a and P3b. *Clinical Neurophysiology*, *118*(10), 2128–2148. http://dx.doi.org/10.1016/j.clinph.2007.04.019.

Polyak, B. T., & Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, *30*(4), 838–855. http://dx.doi.org/10.1137/0330046.

Ravanelli, M., Brakel, P., Omologo, M., & Bengio, Y. (2018). Light gated recurrent units for speech recognition. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *2*(2), 92–102. http://dx.doi.org/10.1109/TETCI.2017.2762739.

Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., et al. (2021). SpeechBrain: A general-purpose speech toolkit. arXiv:2106.04624.

Ravanelli, M., Zhong, J., Pascual, S., Swietojanski, P., Monteiro, J., Trmal, J., et al. (2020). Multi-task self-supervised learning for robust speech recognition. In *ICASSP 2020 - 2020 IEEE international conference on acoustics, speech and signal processing* (pp. 6989–6993). http://dx.doi.org/10.1109/ICASSP40776.2020.9053569.

Riyad, M., Khalil, M., & Adib, A. (2021). MI-EEGNET: A novel convolutional neural network for motor imagery classification. *Journal of Neuroscience Methods*, *353*, Article 109037. http://dx.doi.org/10.1016/j.jneumeth.2020.109037.

Rommel, C., Paillard, J., Moreau, T., & Gramfort, A. (2022). Data augmentation for learning predictive models on EEG: a systematic comparison. *Journal of Neural Engineering*, *19*(6), Article 066020. http://dx.doi.org/10.1088/1741-2552/aca220.

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

Roy, S., Chowdhury, A., McCreadie, K., & Prasad, G. (2020). Deep learning based inter-subject continuous decoding of motor imagery for practical brain-computer interfaces. *Frontiers in Neuroscience*, *14*, http://dx.doi.org/10.3389/fnins.2020.00918.

Sadik, E. S., Saraoglu, H. M., Canbaz Kabay, S., Tosun, M., & Akdag, G. (2021). Comparison of different data augmentation methods with an experimental EEG dataset. In *2021 13th international conference on electrical and electronics engineering*. IEEE, http://dx.doi.org/10.23919/eleco54474.2021.9677865.

Saha, S., Mamun, K. A., Ahmed, K., Mostafa, R., Naik, G. R., Darvishi, S., et al. (2021). Progress in brain computer interface: Challenges and opportunities. *Frontiers in Systems Neuroscience*, *15*, http://dx.doi.org/10.3389/fnsys.2021.578875.

Schirrmeister, R. T., Springenberg, J. T., Fiederer, L. D. J., Glasstetter, M., Eggensperger, K., Tangermann, M., et al. (2017). Deep learning with convolutional neural networks for EEG decoding and visualization. *Human Brain Mapping*, *38*(11), 5391–5420. http://dx.doi.org/10.1002/hbm.23730.

Simões, M., Borra, D., Santamaría-Vázquez, E., Bittencourt-Villalpando, M., Krzemiński, D., Miladinović, A., et al. (2020). BCIAUT-P300: A multi-session and multi-subject benchmark dataset on autism for P300-based brain-computer-interfaces. *Frontiers in Neuroscience*, *14*, http://dx.doi.org/10.3389/fnins.2020.568104.

Smith, L. N. (2015). Cyclical learning rates for training neural networks. http://dx.doi.org/10.48550/ARXIV.1506.01186, URL https://arxiv.org/abs/1506.01186.

Smith, S., & Nichols, T. (2009). Threshold-free cluster enhancement: Addressing problems of smoothing, threshold dependence and localisation in cluster inference. *NeuroImage*, *44*(1), 83–98. http://dx.doi.org/10.1016/j.neuroimage.2008.03.061.

Song, Y., Zheng, Q., Liu, B., & Gao, X. (2023). EEG conformer: Convolutional transformer for EEG decoding and visualization. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *31*, 710–719. http://dx.doi.org/10.1109/tnsre.2022.3230250.

Sutton, S., Braren, M., Zubin, J., & John, E. R. (1965). Evoked-potential correlates of stimulus uncertainty. *Science*, *150*(3700), 1187–1188. http://dx.doi.org/10.1126/science.150.3700.1187.

Tang, X., Zhang, J., Qi, Y., Liu, K., Li, R., & Wang, H. (2024). A spatial filter temporal graph convolutional network for decoding motor imagery EEG signals. *Expert Systems with Applications*, *238*, Article 121915. http://dx.doi.org/10.1016/j.eswa.2023.121915.

Tangermann, M., Müller, K.-R., Aertsen, A., Birbaumer, N., Braun, C., Brunner, C., et al. (2012). Review of the BCI competition IV. *Frontiers in Neuroscience*, *6*, http://dx.doi.org/10.3389/fnins.2012.00055.

Turner, R., Eriksson, D., McCourt, M., Kiili, J., Laaksonen, E., Xu, Z., et al. (2021). Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In H. J. Escalante, K. Hofmann (Eds.), *Proceedings of machine learning research*: *vol. 133*, *Proceedings of the neurIPS 2020 competition and demonstration track* (pp. 3–26). PMLR, URL https://proceedings.mlr.press/v133/turner21a.html.

Vahid, A., Mückschel, M., Stober, S., Stock, A.-K., & Beste, C. (2020). Applying deep learning to single-trial EEG data provides evidence for complementary theories on action control. *Communications Biology*, *3*(1), http://dx.doi.org/10.1038/s42003-020-0846-z.

Vialatte, F.-B., Maurice, M., Dauwels, J., & Cichocki, A. (2010). Steady-state visually evoked potentials: Focus on essential paradigms and future perspectives. *Progress in Neurobiology*, *90*(4), 418–438. http://dx.doi.org/10.1016/j.pneurobio.2009.11.005.

Waytowich, N., Lawhern, V. J., Garcia, J. O., Cummings, J., Faller, J., Sajda, P., et al. (2018). Compact convolutional neural networks for classification of asynchronous steady-state visual evoked potentials. *Journal of Neural Engineering*, *15*(6), Article 066031. http://dx.doi.org/10.1088/1741-2552/aae5d8.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, *1*(6), 80. http://dx.doi.org/10.2307/3001968.

Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., & Vaughan, T. M. (2002). Brain–computer interfaces for communication and control. *Clinical Neurophysiology*, *113*(6), 767–791. http://dx.doi.org/10.1016/s1388-2457(02)00057-3.

Xie, Z., Schwartz, O., & Prasad, A. (2018). Decoding of finger trajectory from ecog using deep learning. *Journal of Neural Engineering*, *15*(3), Article 036009. http://dx.doi.org/10.1088/1741-2552/aa9dbe.

Xu, D., Tang, F., Li, Y., Zhang, Q., & Feng, X. (2023). An analysis of deep learning models in SSVEP-based BCI: A survey. *Brain Sciences*, *13*(3), 483. http://dx.doi.org/10.3390/brainsci13030483.

Xue, Q., Song, Y., Wu, H., Cheng, Y., & Pan, H. (2024). Graph neural network based on brain inspired forward-forward mechanism for motor imagery classification in brain-computer interfaces. *Frontiers in Neuroscience*, *18*, http://dx.doi.org/10.3389/fnins.2024.1309594.

Yao, H., Liu, K., Deng, X., Tang, X., & Yu, H. (2022). FB-EEGNet: A fusion neural network across multi-stimulus for SSVEP target detection. *Journal of Neuroscience Methods*, *379*, Article 109674. http://dx.doi.org/10.1016/j.jneumeth.2022.109674.

Yin, E., Zhou, Z., Jiang, J., Yu, Y., & Hu, D. (2015). A dynamically optimized SSVEP brain–computer interface (BCI) speller. *IEEE Transactions on Biomedical Engineering*, *62*(6), 1447–1456. http://dx.doi.org/10.1109/tbme.2014.2320948.

Yu, T., & Zhu, H. (2020). Hyper-parameter optimization: A review of algorithms and applications. http://dx.doi.org/10.48550/ARXIV.2003.05689, URL https://arxiv.org/abs/2003.05689.

Yuan, H., & He, B. (2014). Brain–computer interfaces using sensorimotor rhythms: Current state and future perspectives. *IEEE Transactions on Biomedical Engineering*, *61*(5), 1425–1435. http://dx.doi.org/10.1109/tbme.2014.2312397.

Zhang, J., Li, K., Yang, B., & Han, X. (2023). Local and global convolutional transformer-based motor imagery EEG classification. *Frontiers in Neuroscience*, *17*, http://dx.doi.org/10.3389/fnins.2023.1219988.

Zhao, D., Tang, F., Si, B., & Feng, X. (2019). Learning joint space–time–frequency features for EEG decoding on small labeled data. *Neural Networks*, *114*, 67–77. http://dx.doi.org/10.1016/j.neunet.2019.02.009.

Zhou, B., Wu, X., Lv, Z., Zhang, L., & Guo, X. (2016). A fully automated trial selection method for optimization of motor imagery based brain-computer interface. In B. He (Ed.), *PLoS One*, *11*(9), Article e0162657. http://dx.doi.org/10.1371/journal.pone.0162657.