Neural Networks 151 (2022) 276–294

Contents lists available at ScienceDirect

Neural Networks

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

Decoding sensorimotor information from superior parietal lobule of macaque via Convolutional Neural Networks

Matteo Filippini [a,*], Davide Borra [b], Mauro Ursino [b,c], Elisa Magosso [b,c], Patrizia Fattori [a,c,*]

[a] University of Bologna, Department of Biomedical and Neuromotor Sciences, Bologna, Italy
[b] University of Bologna, Department of Electrical, Electronic and Information Engineering "Guglielmo Marconi", Cesena Campus, Cesena, Italy
[c] Alma Mater Research Institute for Human-Centered Artificial Intelligence, Bologna, Italy

ARTICLE INFO

Article history:
Received 5 August 2021
Received in revised form 17 January 2022
Accepted 29 March 2022
Available online 5 April 2022

Keywords:
Neural decoding
Posterior parietal cortex
Convolutional neural network
Sensorymotor transformation
Brain–computer interfaces
Monkey

ABSTRACT

Despite the well-recognized role of the posterior parietal cortex (PPC) in processing sensory information to guide action, the differential encoding properties of this dynamic processing, as operated by different PPC brain areas, are scarcely known. Within the monkey's PPC, the superior parietal lobule hosts areas V6A, PEc, and PE included in the dorso-medial visual stream that is specialized in planning and guiding reaching movements. Here, a Convolutional Neural Network (CNN) approach is used to investigate how the information is processed in these areas. We trained two macaque monkeys to perform a delayed reaching task towards 9 positions (distributed on 3 different depth and direction levels) in the 3D peripersonal space. The activity of single cells was recorded from V6A, PEc, PE and fed to convolutional neural networks that were designed and trained to exploit the temporal structure of neuronal activation patterns, to decode the target positions reached by the monkey. Bayesian Optimization was used to define the main CNN hyper-parameters. In addition to discrete positions in space, we used the same network architecture to decode plausible reaching trajectories. We found that data from the most caudal V6A and PEc areas outperformed PE area in the spatial position decoding. In all areas, decoding accuracies started to increase at the time the target to reach was instructed to the monkey, and reached a plateau at movement onset. The results support a dynamic encoding of the different phases and properties of the reaching movement differentially distributed over a network of interconnected areas. This study highlights the usefulness of neurons' firing rate decoding via CNNs to improve our understanding of how sensorimotor information is encoded in PPC to perform reaching movements. The obtained results may have implications in the perspective of novel neuroprosthetic devices based on the decoding of these rich signals for faithfully carrying out patient's intentions.

1. Introduction

The posterior parietal cortex (PPC) houses several areas implicated in the integration of sensory stimuli (e.g., visual, somatosensory) to guide interaction with the surrounding environment (Andersen & Cui, 2009a; Medendorp & Heed, 2019). Sensory information flows through different parietal areas, and different steps of integration support the movement control required for the interaction. A first functional subdivision was proposed by Ungerleider and Mishkin (Goodale & Milner, 1992; Ungerleider & Mishkin, 1982) with the division into ventral stream, from occipital to temporal lobe, the *what way*, and dorsal visual stream, to the parietal lobe, the *where way*. Subsequent division of the dorsal pathway into medial and lateral streams attributed the superior parietal lobule (SPL) to the branch of the medial dorsal stream, which host areas implicated in the spatial control of the action, against the areas of the inferior parietal lobule, more involved in the control of the properties of grasping (Galletti, Kutz, Gamberini, Breveglieri, & Fattori, 2003; Rizzolatti & Matelli, 2003). In humans, lesions localized in the SPL lead to severe deficits in estimation and awareness of the spatial position of objects to be reached, such as those reported in optic ataxia (Karnath & Perenin, 2005; Pisella et al., 2010).

Within PPC, V6A, PEc and PE are three contiguous areas located in the medial part of the SPL. V6A, the most caudal area of the SPL, hosts neurons modulated by different parameters linked to visuomotor coordination, including gaze signals (Galletti, Battaglini, & Fattori, 1995; Hadjidimitrakis, Breveglieri, Bosco, & Fattori, 2012; Hadjidimitrakis et al., 2011), direction and amplitude of reaching (Fattori, Kutz, Breveglieri, Marzocchi, & Galletti, 2005; Hadjidimitrakis et al., 2014) and spatial attention (Galletti et al., 2010). Rostrally to V6A, PEc maintains visual

* Correspondence to: Department of Biomedical and Neuromotor Sciences, Piazza di Porta San Donato 2, 40126 Bologna, Italy.
E-mail addresses: matteo.filippini7@unibo.it (M. Filippini), patrizia.fattori@unibo.it (P. Fattori).

https://doi.org/10.1016/j.neunet.2022.03.044

and reach related modulations but corroborated by increasing somatosensory inputs (Gamberini et al., 2018; Hadjidimitrakis, Dal Bo', Breveglieri, Galletti, & Fattori, 2015). Finally, PE strongly respond to proprioceptive stimulation with a limited presence of visual information (De Vitis et al., 2019; Seelke et al., 2012). The segregated functional properties of these areas support the idea that caudal SPL (V6A) encodes for spatial position of targets, while rostral areas (PEc-PE) actively support the movement control and execution relying on prominent sensory feedbacks (Breveglieri, Galletti, Monaco, & Fattori, 2008; Cui & Andersen, 2011; Gamberini et al., 2018; Hadjidimitrakis et al., 2015). Given the latency in the feedback inputs, the system must implement an internal model of surrounding environment (and the consequences of actions performed in the environment) and expected feedbacks. Indeed, comparing expected with real feedback enables a much more powerful dynamical interaction with the environment. This model has been proposed to run in the PPC (Land, 2014; Medendorp & Heed, 2019; Mulliken, Musallam, & Andersen, 2008a).

Despite the presence of numerous studies, a clear understanding of how these PPC's areas differentially encode visuomotor information to dynamically guide action is still lacking. In particular, these areas have been mostly characterized at single cell level (De Vitis et al., 2019; Hadjidimitrakis et al., 2014, 2015) while, at the best of our knowledge, no work has comprehensively considered and directly compared the dynamics of information encoded, at the population level, in these three areas. Neural decoding, i.e., the use of activity recorded from multiple brain sources to predict variables in the external word, represents a useful tool to characterize how much information a given area contains about an external variable and how this information differs across different areas (Glaser et al., 2020). The attainment of different decoding performances when building separate decoders, e.g., each using neural activity from a different PPC area, may be indicative of a different amount of information encoded in each population.

Machine learning (ML) algorithms are widely used to design neural decoders (Glaser et al., 2020). Deep learning – a recently proposed branch of ML recently proposed in the computer vision field (LeCun, Bengio, & Hinton, 2015) – exploits models designed by stacking layers of artificial neurons, i.e. deep neural networks (DNNs). Remarkably, DNNs are capable to handle raw/lightly pre-processed neural time series as input, automatically learning during a training process the more relevant features to decode the brain states of interest while exploiting all the information contained in the input signals. Therefore, DNNs represent an important advantage over more traditional ML approaches; the latter first extract and select features from input neural time series, and then finalize the decoding task, resulting in a workflow more driven by a priori knowledge about the expected underlying neural correlates. Furthermore, conversely to ML algorithms, DNNs provide a direct processing of the information from the neural signals to the desired output (in an end-to-end fashion), describing in general a complex non-linear function mapping input signals to desired outputs. Nevertheless, DNNs have some limitations, such as the introduction of many trainable parameters and the introduction of a second set of parameters, named hyper-parameters, that define the functional form of the decoder and must be set before the DNN training (e.g., the number of layers, the number of neurons per layer, etc.). Among DNNs, Recurrent Neural Networks (RNNs) were used to decode arm kinematics from spiking activity (Sussillo et al., 2012; Tseng, Urpi, Lebedev, & Nicolelis, 2019) also from PPC signals (Sanchez, Erdogmus, Nicolelis, Wessberg, & Principe, 2005; Shah et al., 2019), generally using complex and heavy architectures (in terms of architecture structure and number of trainable parameters, respectively), with a fixed configuration selected via empirical evaluations (i.e., manual selection) or selected without any explicit criterion, and hardly interpretable. Therefore, these DNNs do not represent a parsimonious use of trainable parameters and, due to the high number of trainable parameters, DNNs could be more prone to overfit small datasets (commonly recorded in practice). Furthermore, the adoption of manual selection to define the DNN structure and assign hyper-parameters could lead to sub-optimal decoding results, limiting the potentialities of DNNs, and require manual effort to be performed. Convolutional Neural Networks (CNNs) could also be used without hampering the decoding performance, reducing the number of trainable parameters with respect to RNNs and being easier to be interpreted in the learned features (Tjoa & Guan, 2020). These algorithms are inspired by the hierarchical structure of the ventral stream of the visual system and thus, automatically learn hierarchical representations of the input signal with multiple levels of abstraction (Yamins et al., 2014). Despite being scarcely applied for the decoding of brain states directly from the neuron activity, there is a growing interest in the design and application of CNNs over other DNNs (such as RNNs), as reported in related fields of electroencephalogram decoding (Craik, He, & Contreras-Vidal, 2019; Simões et al., 2020). In addition, techniques aimed to automatically search the optimal hyper-parameter configuration of the decoder, such as Bayesian Optimization (BO), could be used to automatically design decoder functional forms without relying on mere empirical evaluations.

The aim of this study is to develop and use CNNs to accurately decode external variables (reaching goal and trajectory) from PPC neural activity. CNNs were used to catch temporal dynamics and model non-linearity distinctive of high-order areas. To this end, we recorded the activity of single neurons from macaque V6A, PEc, and PE areas while monkeys were performing a delayed reaching 3D task to 9 reaching targets. We approached two different decoding problems for a wider validation of CNNs. At first, we performed the classification of the 9 discrete spatial positions; this problem was addressed by predicting the output class as a function of time within the reaching trials, providing a dynamic decoding of the end-point during the reaching phases. To test whether the non-linear input/output mapping as performed by DNN methods was superior then simpler linear mapping, CNN classification performances were compared against a linear classifier. Then, a regression problem consisting in predicting the 3D hand trajectory of reaching was tackled; this problem was also useful to explore internal PPC model. Remarkably, to overcome the current limitation in designing and using DNNs for neural decoding, the CNN structure and hyper-parameters were tuned using an automatic hyper-parameter search algorithm based on BO. With this approach, we mainly aspire:

i. To improve the state-of-the-art of decoding techniques. Indeed, at best of our knowledge, this is the first time that CNNs are validated and used to decode neuron spiking rate, and that Bayesian optimization is used in this context. This may represent a significant step forward, as CNNs may result highly lighter and faster than RNNs on one hand, and more accurate than simpler linear decoders on the other. The possibility to obtain decoders less handcrafted, more efficient and accurate than other solutions may not only boost a better comprehension of the characteristics of information contained inside the decoded neural populations, but also have implications in neural engineering, such as helping advancement in brain–computer communication tools.

ii. To investigate how the reaching target position and the hand kinematics (3D position) are differently encoded in the three examined PPC areas at the population level, by analyzing the performance of the tested decoders across the three areas. Via this analysis, we wish to evidence how neural decoders, via data-driven input–output mapping, can have significant potential

to inform about the nature of information contained in neural populations. We also expect that, when compared to the linear classifier, CNNs, taking into account nonlinearity and the temporal dynamics (via temporal convolutions) can better catch the characteristics of each area.

2. Methods

The experimental part of this study was performed in accordance with the guidelines of the EU Directives (86/609/EEC; 2010/63/EU) and the Italian national law (D.L. 116-92, D.L. 26-2014) on the use of animals in scientific research. Protocols were approved by the Animal-Welfare Body of the University of Bologna. During training and recording sessions, particular attention was paid to any behavioral and clinical sign of pain or distress. More details on the experimental procedures can be found in Breveglieri et al. (2012), Galletti, Fattori, Kutz, and Gamberini (1999), Galletti, Gamberini, Kutz, Baldinotti, and Fattori (2005), Hadjidimitrakis et al. (2014)

2.1. Data acquisition

2.1.1. Electrophysiological recordings

Two male macaque monkeys (Macaca fascicularis) weighing 4.4 kg (Monkey 1, M1) and 3.8 kg (Monkey 2, M2) were used. Extracellular single-cell activity was recorded by means of single electrode from areas V6A, PEc and PE (Fig. 1a). V6A is localized in the anterior bank of the parieto-occipital sulcus (POs) (Galletti et al., 1999). Next and rostrally to V6A, on the exposed surface of SPL, there is PEc; recording sites were assigned to area PEc according to the cytoarchitectural criteria described by Luppino, Ben Hamed, Gamberini, Matelli, and Galletti (2005). Finally, between somatosensory cortex and PEc, we recorded from the posteromedial part of PE (De Vitis et al., 2019). We performed multiple electrode penetrations using a five-channel multielectrode recording system that permitted to record from up to five single electrodes at once (Thomas Recording GmbH, Giessen, Germany). We recorded the activity of 258 V6A neurons, 120 cells from M1 and 138 cells from M2, 214 neurons from PEc, 94 and 120 from M1 and M2 respectively, 141 from area PE, 71 and 70 from M1 and M2 respectively. Action potentials (spikes) in each channel were isolated with a waveform discriminator (Multi Spike Detector; Alpha Omega Engineering Nazareth, Israel) and were sampled at 100 kHz. The quality of the single-unit isolation was determined by the homogeneity of spike waveforms and clear refractory periods in ISI histograms during spike-sorting. Only well-isolated units not changing across tasks were considered.

2.1.2. Behavioral task

Monkeys sat in a primate chair (Crist instruments, Hagerstown, MD, USA) and were trained to perform a Fixation-to-Reach task under controlled conditions. This task was performed in darkness with the arm contralateral to the recording hemisphere. During the task, the animals maintained steady fixation on the cued (green) reaching target with their head restrained. Before starting the arm movement, the monkeys kept their hand on a button (home button [HB], 2.5 cm in diameter) located 5 cm in front of the chest on the midsagittal plane (Fig. 1b left). Reaches were performed to one of nine light-emitting diodes (LEDs, 6 mm in diameter). The LEDs were mounted on a horizontal panel located in front of the animals, at different distances and directions with respect to the eyes but always at eye level, so the movement performed by the monkeys to reach and press the LED was upward. Target LEDs were arranged in three rows: one central, along the sagittal midline, and two laterals, at version angles of $-15°$ and $+15°$, respectively (Fig. 1b right). Along each row, three LEDs were located at vergence angles of 17.1°, 11.4°, and 6.9°. The nearest targets were located at 10 cm from the eyes, whereas the LEDs placed at intermediate and far positions were at a distance of 15 and 25 cm, respectively. The range of vergence angles was selected to include most of the peripersonal space in front of the animals, from very near (10 cm) to the farthest distances reachable by monkeys (25 cm).

The trial began when the animals pressed the button near their chest, outside the field of view (HB press). After 1s, one of the nine LEDs was switched on to green (Green-on). The monkeys had to fixate the LED within 500 ms, while keeping the HB button pressed. Then, the monkeys had to wait 1.7–2.5 s for a change in the color of the LED (from green to red) without performing any eye or arm movement. The latter color change was the go signal (Go) for the animals to release the home button and to start an arm movement towards the foveated target. Then, the monkeys reached the target and held their hand on the target for 0.8–1.2 s. When the target cue was switched off, the monkeys had to release this cue and return to the HB, which ended the trial and allowed the monkeys to receive a reward.

The task was performed in blocks of 90 randomized trials, including 10 repetitions for each target position (out of the 9 possible target positions). According to Fig. 1c we divided each recording trial in 5 functional epochs. Epoch 0 represented the interval in which the monkey was not engaged in the task waiting for the LED turning on; Epoch 1 and Epoch 2 labeled the delay interval, specifically since the delay interval has random duration, it was separated in its first second (epoch 1) and last second (epoch 2); Epoch 3 represented the reaction interval; Epoch 4 was the movement interval (reaching movement towards the target point); Epoch 5 represented the holding interval (hold on the target point).

The luminance of LEDs was regulated to compensate for difference in retinal size between LEDs located at different distances. To prevent dark adaptation, the background light was switched on between blocks. The presentation of stimuli and the animals' performance were automatically controlled and monitored by LabVIEW-based software (National Instruments) as described previously (Kutz, Marzocchi, Fattori, Cavalcanti, & Galletti, 2005), enabling the interruption of the trial if the monkey broke fixation, made an incorrect arm movement, or did not respect the temporal constraints of the task described above. The correct performance of movements was monitored by pulses from microswitches (monopolar microswitches, RS Components, UK) mounted under the home button and each LED. At the beginning of each recording session, the monkeys were required to perform a calibration task to calibrate an eye tracker (ISCAN, see below). Calibration data was used to correct eye signals as they are dependent on the position of the cameras which can potentially change from day to day. For the calibration, animals sequentially fixated 5 LEDs mounted on a vertically arranged panel placed at a distance of 15 cm from the eyes. For each eye, we extracted signals for calibration during fixation of five LEDs, arranged in the shape of a cross. One LED was centrally aligned with the eye's straight-ahead position and four LEDs were peripherally placed at an angle of $\pm 15°$ (distance: 4 cm) both in the horizontal and vertical axes. From the two individually-calibrated eye position signals, we derived the mean of the two eyes (version signal) and the difference between the two eyes (vergence signal) using the following equations: $version = (R + L)/2$, $vergence = R - L$, where R and L are the gaze direction of the right and left eye respectively, expressed in degrees of visual angle from the straight-ahead direction. Eye signals were monitored to be sure that the animal was staring at the target while performing the task, reducing possible modulation of neurons' firing rates due to saccade execution (Kutz, Fattori, Gamberini, Breveglieri, & Galletti, 2003).

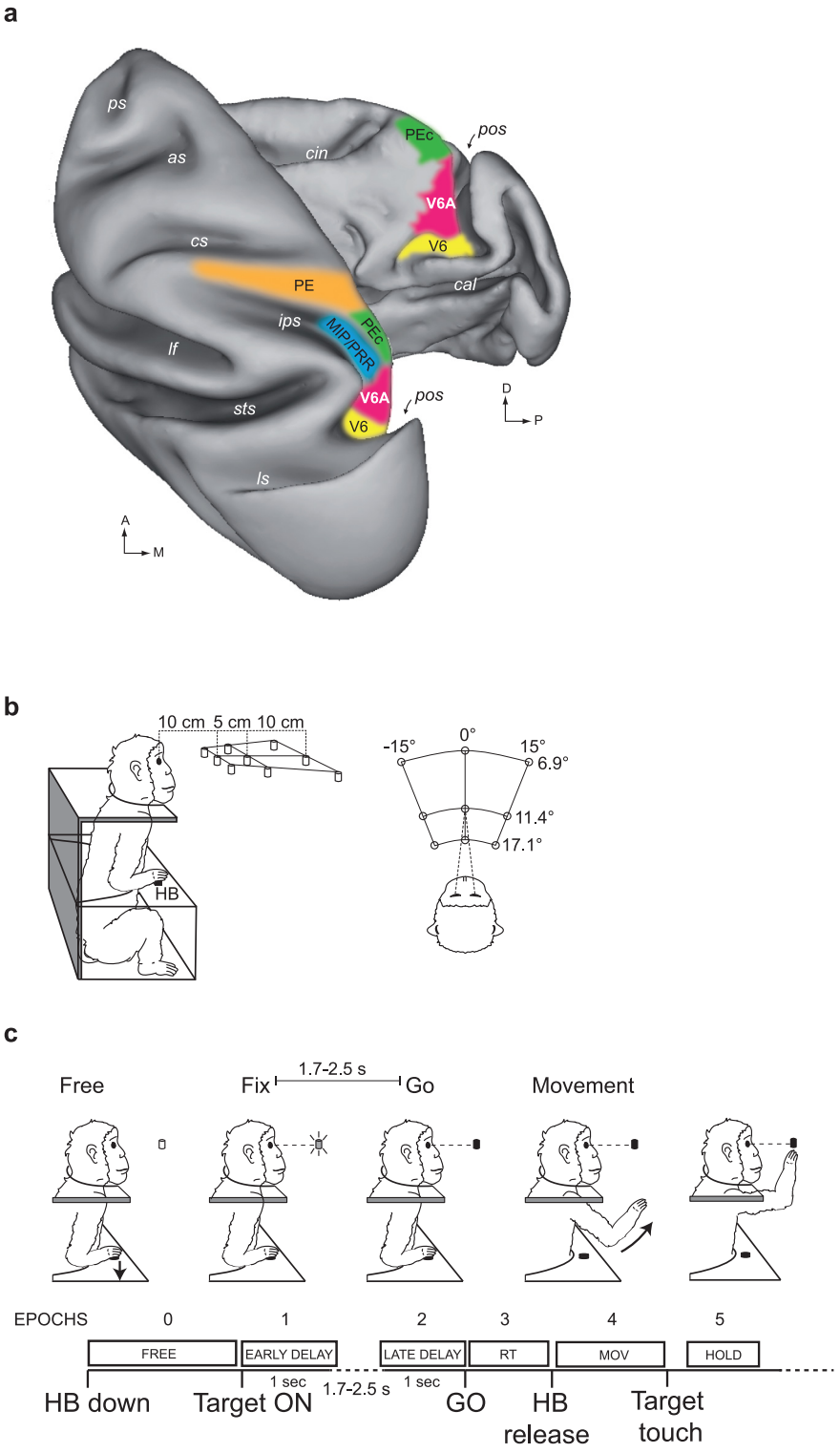

Fig. 1. Recording areas and Behavioral Task. (a) Dorsal view of left hemisphere (left) and medial view of right hemisphere (right) reconstructed in 3D. Caret software (http://brainvis.wustl.edu/wiki/index.php/Caret:Download) was used, showing the location and extent of V6A (magenta), PEc (green), PE (orange), MIP/PRR, medial intraparietal area/parietal reach region (Snyder, Batista, & Andersen, 1997) (blue), and V6 (Galletti et al., 1999) (yellow); cal, calcarine sulcus; cin, cingulate sulcus; cs, central sulcus; ips, intraparietal sulcus; lf, lateral fissure; ls, lunate sulcus; pos, parieto-occipital sulcus; ps, principal sulcus; sts, superior temporal sulcus; D, dorsal; P, posterior. (b) Schematic representation of the experimental setup for the reaching task. Exact distances are indicated in the lateral (left) and top (right) views. Nine LEDs are used as targets, embedded in a panel located at eye level. HB = home button. (c) Time courses and behavioral epochs in reaching tasks. The task starts with the animal holding down the HB (FREE 1 s, epoch 0), then a fixation LED lights up on one of the 9 positions, the animal starts to fixate and waits (first 1 s of DELAY, EARLY DELAY, epoch 1) for the change of color of the LED that occurs at the end of the delay (LATE DELAY, last 1 s of DELAY, epoch 2) with the GO event. The reaction time (epoch 3) is then from the GO signal to the actual release of the HB. The animal releases the HB to perform the movement (MOV, epoch 4) then keeps the LED target pressed for the HOLD interval (epoch 5) to then return to the HB and begin the next trial. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Number of recorded neurons for each animal and each recording area. In addition, the total number of training, validation and test examples in supervised problems 1 and 2 as resulting from the trial chunking procedure (see Section 2.2) are reported. Note that the reported values for supervised problem 2 refer to the default assignment of the desired output (i.e., with no offset in the assignment).

Monkey	Area	N	Supervised problem 1 (epochs 0–5)			Supervised problem 2 (epochs 2–5)		
			Training	Validation	Test	Training	Validation	Test
M1	V6A	138	3312	414	2070	1872	234	1170
	PEc	120	3384	423	2079	1944	243	1179
	PE	70	3312	414	2061	1872	234	1161
M2	V6A	120	3312	414	2052	1872	234	1152
	PEc	94	3312	414	2061	1872	234	1161
	PE	71	3312	414	2043	1872	234	1143

2.2. Data pre-processing and preliminary analysis

For each neuron and each individual recording trial, the activity was initially binned at 20 ms. Since the trials and epochs have a different duration, the use of a constant temporal window produces a different number of bins across trials, not allowing to construct a uniform dataset. Therefore, the average number of bins (across different neurons and trials) of each epoch was computed; then, the activity of each neuron and trial was re-binned by using that number of average bins per epoch. This procedure led to an activity binned slightly more or less with respect to the original 20 ms binning (20.1 $\pm$ 1.9, mean $\pm$ standard deviation across monkeys and areas). Thus, to address this, we computed firing rates by dividing the number of spikes occurring within the bin by the temporal length of the bin. In the following, the neuron activity is described by means of its firing rate and to study the temporal dynamics of neural coding we constructed some short signals (named "chunks" in the following) composed of overlapped windows of B bins, extracted with stride of S bins to be used as inputs to the CNN.

As preliminary analysis, we looked for the neurons modulated by the reaching task. Only in this case, the neuron activity was divided into non-overlapped chunks of $B = 5$ bins (i.e., extracted with a stride of $S = 5$) bins and was analyzed using a sliding ANOVA to assess the variance in neuronal activity between the different conditions tested. One neuron was considered significantly modulated with $p < 0.01$. Results were plotted as number of modulated neurons over time.

2.3. CNN-based population decoding

2.3.1. Problem definition

Firing rates from every single neuron of the investigated population obtained from a specific monkey – identified with $m = \{$"$M1$", "$M2$"$\}$ – and a specific recording area – identified with $a = \{$"$V6A$", "PEc", "PE"$\}$ – were processed as follows to perform decoding. At first, overlapped chunks of shape (N, B) were extracted using a stride parameter S, where N denotes the number of neurons recorded for a given animal and area (variable across animals and areas, representing the spatial dimension, see Table 1 for N values across animals and areas), and B denotes the number of bins in each extracted chunk (representing the time dimension). By denoting with X_t the firing rates of each entire trial, and with $X_t[i]$ the ith bin, chunks of neural activity were extracted as follows:

$$X_{t,i} = X_t\left[:, iS : iS + B - 1\right], 0 \leq i \leq M - 1, \tag{1}$$

indicating with $X_{t,i} \in \mathbb{R}^{N \times B}$ the ith extracted chunk of firing rates for the t-th trial (see Fig. 2a for a schematization of this chunking procedure), and with M the total number of chunks.

$X_{t,i}$ represented the CNN input, while $y_{t,i}$ denotes the corresponding desired output. Thus, each so labeled dataset can be denoted by $D^{(m,a)}$:

$$D^{(m,a)} = \{\left(X_{t,0}, y_{t,0}\right), \ldots, \left(X_{t,i}, y_{t,i}\right), \ldots, \left(X_{t,M-1}, y_{t,M-1}\right)\}^{(m,a)} \tag{2}$$

In this study, two different supervised problems were addressed using the same dataset. At first, we deepened the classification of targets to reach. In this case (supervised problem 1), $y_{t,i} \in L = \{l_0, \ldots, l_8\}$ was the desired class label assuming one among 9 possible values, corresponded to each target point that was constant in time during the trial (i.e., the end-point to reach did not change within each trial). The supervision during classification was provided using one-hot encoded labels, i.e., forming an array corresponding to the true probability distribution with zeros for all classes except for the desired one which is 1. In addition, we investigated a regression problem consisting in the prediction of the 3D hand position while reaching targets (the same targets classified in the first supervised problem). In this case (supervised problem 2), $y_{t,i} \in \mathbb{R}^3 = [p_x, p_y, p_z]$ and corresponded to the 3D hand position coordinates at a specific time point to be predicted in the regression problem related to semi-synthetic trajectories (see Section 2.3.4 for further details). That is:

$$\begin{cases} y_{t,i} \in L = \{l_0, \ldots, l_8\}, \textit{supervised problem } 1 \\ y_{t,i} \in \mathbb{R}^3 = [p_x, p_y, p_z], \textit{supervised problem } 2. \end{cases} \tag{3}$$

Concerning the regression problem, as the hand position changed during each reaching trial, the desired output $y_{t,i}$ was assigned the 3D position coordinates in correspondence of the last bin of $X_{t,i}$ (referred as "default assignment", see Section 2.3.4 for further details), i.e., the prior neural activity of $B \cdot 20$ ms (e.g., 300 ms in case of $B = 15$ bins) was assigned to the current observable parameter, see Fig. 2a for a schematization of this output association procedure. This corresponds to define $y_{t,i}$ as:

$$y_{t,i} = \left[p_x, p_y, p_z\right]\left[(iS + B - 1) + o\right], \tag{4}$$

where $iS + B - 1$ denotes the time sample corresponding to the last bin of the chunk $X_{t,i}$, and o denotes an additional offset factor. The latter is set to 0 in the default assignment, while assumes positive or negative values in the latency analyses performed in Section 2.3.4.

For both these supervised problems, the objective is to optimize – using a training set of examples (training stage) – a CNN described by a parametric model $f\left(X_{t,i}; \theta\right)$ parametrized in its trainable parameters θ, implementing a classifier in the first supervised problem, i.e. $f\left(X_{t,i}; \theta\right) : \mathbb{R}^{N \times B} \rightarrow L$, or a regressor in the second supervised problem, i.e. $f\left(X_{t,i}; \theta\right) : \mathbb{R}^{N \times B} \rightarrow \mathbb{R}^3$. As mentioned above, the CNN accepts as input the ith chunk $X_{t,i}$ and provides the predicted output $yp_{t,i}$. This optimization corresponds to find the optimal $\theta^* = arg\ \min_\theta j(\theta)$, where $j(\theta)$ denotes the loss function which is computed based on the prediction error

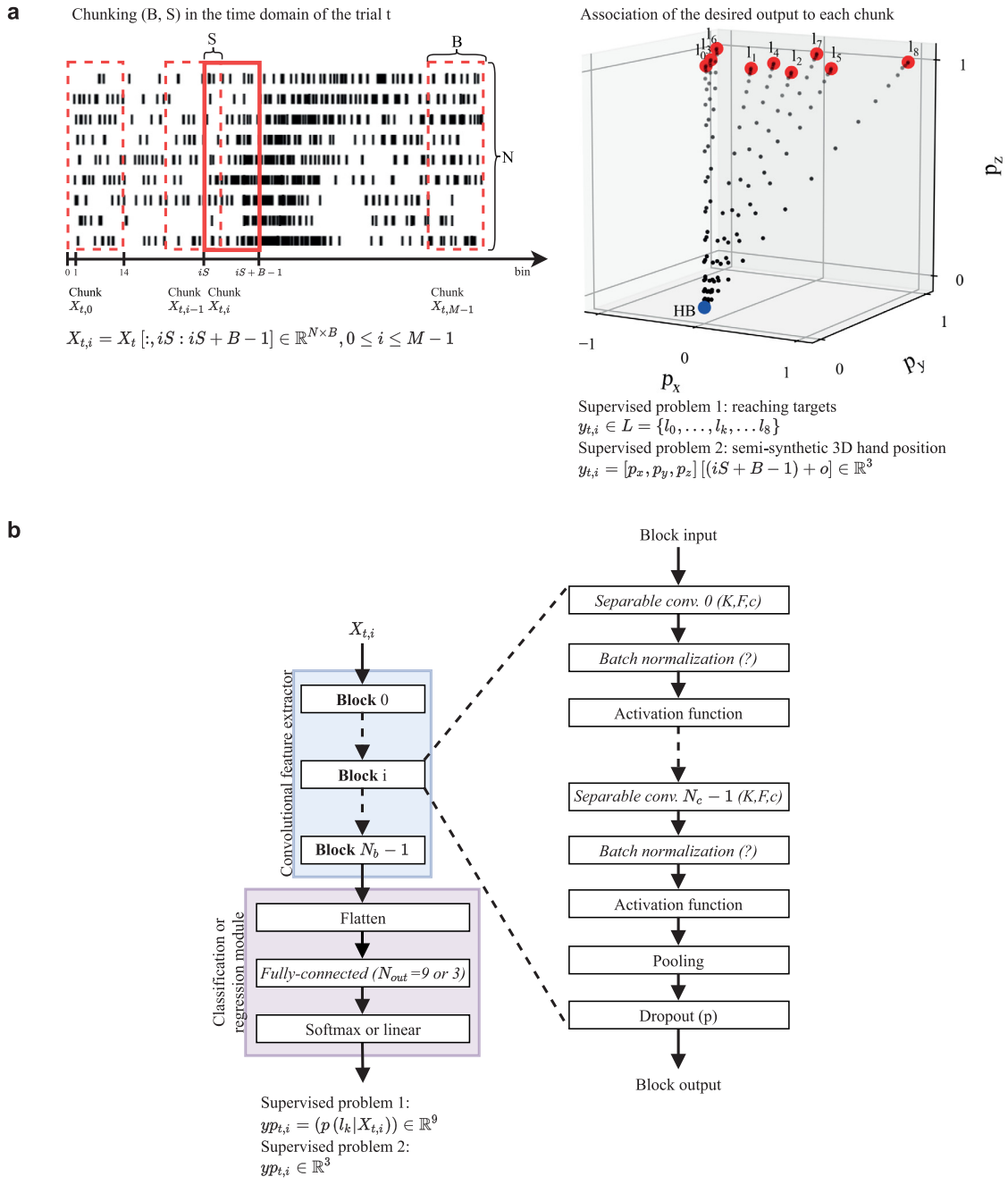

Fig. 2. Schematic representation of the dataset construction (a) and of the CNN design (b). (a) Neuron activities (left) and the desired output (right, corresponding to the reaching targets or hand trajectories) were processed to obtain ($X_{t,i}$, $y_{t,i}$), see Section 2.2. (b) $X_{t,i}$ is fed to the CNN, which provides $yp_{t,i}$ as output (predicted output). The CNN is composed by a convolutional feature extractor (blue box), followed by a classification or regressor module (purple box), see Section 2.3.2 for the meaning of the symbols. Layers are represented by white boxes; layers introducing trainable parameters are denoted by the italic font, with the main associated hyper-parameters reported within brackets (see Section 2.3.2 for the meaning of the symbols).

between the desired output $y_{t,i}$ and the predicted output $yp_{t,i}$. Afterwards, once the model is trained, it is tested on a separate test set (inference stage). Besides the trainable parameters contained in θ, the parameters defining the specific functional form of the decoder (e.g., number of convolutional layers, number of kernels to learn, etc.), called hyper-parameters, need to be set before training starts. However, the optimal hyper-parameter configuration is not a priori known and can be chosen either via an extensive empirical evaluation or via automatic hyper-parameter search. For both these solutions, to study the influence of different hyper-parameter configurations, the decoder needs to be validated on a separate set (different from the training and test set). Therefore, the dataset $D^{(m,a)}$ needs to be partitioned into separate training and test sets, respectively to optimize the parameters contained in θ and to evaluate the performance of the learning system on unseen data. In addition, to select the optimal hyper-parameter configuration (see Section 2.3.5) a separate validation set needs to be designed. To perform such data partitioning, $D^{(m,a)}$ was divided using a 10-fold cross validation scheme. Starting from 10 trials recorded for each spatial position

(a total of 90 trials were available), each cross-validation fold included 72, 9, 9 trials in the training, test and validation sets, respectively, balancing across reaching targets (i.e., sets contained the same proportion among the 9 reaching targets). During the training stage, the parameters used to design $D^{(m,a)}$ were $B = 15$ and $S = 5$, while during the inference stage, these were $B = 15$ and $S = 1$. This choice allowed to train decoders with overlapped chunks of data i.e., augmenting the overall training set, but without increasing excessively the computational time ($S = 5$ during training), and to test the decoders on all possible chunks ($S = 1$ during testing). See Table 1 for the total number of training, validation and test examples, for each animal and recording area.

2.3.2. CNN architecture

The general structure of the CNN architecture is reported in Fig. 2b and it is described in the following.

The input layer of the CNN was represented by a 2D input feature map replicating in each neuron the corresponding value of the input example $X_{t,i} \in \mathbb{R}^{N \times B}$, i.e., the input layer was a 2D matrix with N rows and B columns representing the firing rates, of N neurons in B time bins. Afterwards, the input example was processed through a *convolutional feature extractor* to learn and extract relevant feature maps from the input example, followed by a *classification or regression module* that finalized the decoding depending on the addressed supervised problem addressed and based on the feature maps provided by the first module. Regarding the convolutional feature extractor, this was composed by stacking N_b convolutional blocks. Each convolutional block is composed by N_c repetitions of 1D temporal separable convolutional layers (Chollet, 2017), each one learning K temporal kernels with a size of F, followed by batch normalization (Ioffe & Szegedy, 2015) (optional, depending on the hyper-parameter search) and a non-linear activation function. Then, after these N_c repetitions, each convolutional block included also a pooling layer — aimed to reduce the temporal dimension and thus, to reduce the overall model size by applying a pooling function (which is a hyper-parameter too, e.g., max or average pooling) – and a dropout layer (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014), with dropout probability p. All convolutional layers were constrained (optional, depending on the hyper-parameter search) in their norm, keeping the norm of their parameters upper bounded at a constant c. Overall, batch normalization, dropout and kernel max norm constraints were introduced in the convolutional feature extractor to reduce overfit (i.e., regularization mechanisms). In addition, this module was designed using convolutional layers devoted to keep limited the number of trainable parameters, i.e., separable convolutions, preventing overfitting small datasets as the ones used in this study. The main hyper-parameters of the convolutional feature extractor were searched using an automatic hyper-parameter search algorithm (see Section 2.3.5 for further details).

The classification or regression module reshaped (flatten layer) at first the feature maps provided by the first module and included a fully-connected layer with N_{out} output artificial neurons to output the desired variables (target positions, $N_{out} = 9$ or hand position coordinates, $N_{out} = 3$). Depending on the addressed decoding problem, the activation function of the fully-connected layer changed. In case of classification, $N_{out} = 9$ neurons were activated using a softmax function, to provide as output the array $yp_{t,i} = p\left(l_k|X_{t,i}\right) \in \mathbb{R}^9, 0 \leq k \leq 8$ of the predicted conditional probabilities associated to each target. Then, the most probable class was computed, i.e., $arg\max_{l_k} p\left(l_k|X_{t,i}\right)$, and the decoded class obtained ($\in L = \{l_0, \ldots, l_8\}$). In case of regression, $N_{out} = 3$ neurons were activated using a linear function, to directly provide as output the hand position coordinates $yp_{t,i} \in \mathbb{R}^3$ while reaching targets.

2.3.3. Supervised problem 1: Target decoding

In the case of the classification problem, signals of the training, test and validation sets were standardized using the mean and standard deviation computed on the training set. The network used as input the signals pre-processed as described in Section 2.2 from epochs 0–5 (see Section 2.1.2) and provided as output the conditional probabilities for each target position. During the training stage, the loss function $j(\theta)$ was defined as the cross-entropy between the predicted distribution (provided by the CNN) and the empirical distribution (provided in the labeled dataset). During the inference stage on the test set, the CNN provided as output the probabilities that the input chunk belongs to each class; the predicted class was computed as the one with the highest probability among the 9 possible classes (see the possible reaching targets n Fig. 2a right). Then, the decoding accuracy was computed based on the predicted and true classes. Accuracies on the test set were computed for each monkey and each recording area as a function of time, i.e., computing accuracies chunk by chunk. To provide a comparison with a state-of-the-art linear algorithm, a Naïve Bayes (NB) classifier as the one adopted in Filippini, Breveglieri, Hadjidimitrakis, Bosco, and Fattori (2018) was trained and evaluated with the same procedure adopted for the proposed CNN.

As the number of neurons recorded from the three areas differed (e.g., 70 neurons for PE vs. 100+ neurons in the other areas, see Table 1), we assessed whether differences in performance among different areas may be the consequence of a different number of neurons rather than intrinsically depend on differences in the information provided by neuron activities. To this aim, a dropping analysis was performed. During the dth step of the dropping analysis, in each monkey and in each area, N_d neurons were randomly selected from $N_d = 2$ to $N_d = N$ with a step of 5 cells. That is, a subset containing N_d cells was randomly sampled from the original distribution and used to train, validate and test CNNs (using the same cross-validation scheme as described in Section 2.3.1). In this way, the decoding performance was evaluated using the same number of N_d neurons in each area. In addition, this analysis simulates conditions of a reduced set of cells to decode, e.g., due to fibrosis around implanted electrodes, at different levels. The random sampling was performed 20 times for each dth step of the dropping analysis and was performed for each monkey and each recording area. Due to the high computational cost of such simulation (involving >50 K CNN optimizations), we applied the dropping analysis only for the supervised problem 1.

2.3.4. Supervised problem 2: Hand trajectory decoding

In the case of the regression problem, signals of the training, test and validation sets were standardized using the statistics computed on the training set, and the target coordinates were normalized between [−1,1] (centered on the mid sagittal axis of the animal), [0,1] (with 0 the position of the home button and 1 the maximum distance from the body corresponding to the farthest target), and [0,1] (with 0 the elevation of the home button and 1 the height of the panel at eye level), respectively for the x-, y-, z-axis . To decode the hand trajectory, the network used as input the signals pre-processed as described in Section 2.2 from the epochs 2–5 (see Section 2.1.2) and provided as output the x, y, z hand position coordinates during the reaching of each target position.

Kinematic data of hand trajectories was not available for recorded neurons, therefore the reference trajectory of the hand

during the experiment was reconstructed semi-synthetically, modeling the movement as a straight ballistic motion from the button near the chest to one of 9 positions on the panel (see the trajectories in Fig. 2a right) and imposing a classic bell-shaped profile for the acceleration and deceleration (Roy, Paulignan, Farnè, Jouffrais, & Boussaoud, 2000). The bell-shaped profile (a Gaussian bell) was fitted to the average speed profile collected from real kinematic data, calculated over 144 trials on a separate monkey executing the same 3D reaching task, using as reference the x, y, z positions of index finger. A motion capture system (VICON 460, 100 Hz sampling rate) recorded the 3D position of a reflective marker placed on the monkey's index finger. Data were run through a fifth-order Butterworth low-pass filter, finally trajectory were downscaled to 0%–100% of movement to make possible aligning the different trials collected. Mean and standard deviation of the Gaussian bell determined the peak of maximum velocity (43% of movement time) and acceleration/deceleration ramp ($\sigma = 18\%$ of movement time) and were used to reconstruct the reference hand trajectory for all cells.

As here we were interested in epochs including the last part of the waiting period until the target was maintained, the CNN had to learn to hold the initial position during the waiting time (0,0,0), gradually move towards the target during the movement interval, and hold the position during the last interval.

During the training stage, the loss function $j(\theta)$ was defined as the mean squared error between the predicted trajectory value (provided by the CNN) and the empirical trajectory value (provided in the labeled dataset). During the inference stage on the test set, the network output was directly the predicted trajectory value for the corresponding input firing rate chunk. The predicted trajectory was then obtained by rearranging all values in the time-domain, and was compared with the semi-synthetically reconstructed trajectory, used as a ground truth. R-squared values were computed for each monkey and each recording area (reporting mean and standard deviation across folds).

In addition, only for the supervised problem 2, we conducted an analysis to study possible latencies, i.e., −120 ms, −40 ms, 40 ms, 120 ms, between neuron activity (firing rates contained in each chunk $X_{t,i}$) and detected behavior (instantaneous trajectory value in its x, y, z position coordinates). To this aim, we introduced a time shift o between the desired output (x, y, z position coordinates) assignment and the neuron activity when designing the datasets $D^{(m,a)}$ and we trained, validated and tested CNNs for each offset condition. By default (see Section 2.3.1), a zero offset ($o = 0$) was used, indicating that the assigned 3D trajectory value was sampled in correspondence of the last bin of the input chunk (i.e., to the 15th bin, see Section 2.3.1). In addition to the default assignment, we deepened other conditions, by using offsets $o = \{-6, -2, 2, 6\}$. Positive (or negative) offset values denote conditions where CNNs were forced to learn features from past (or future) neuron activity ($X_{t,i}$). This suggests that neurons are coding for future trajectories (feedforward anticipation) or past trajectories (sensory feedback). As an example, when $o = +2$, position coordinates sampled at +40 ms ($= o \cdot 20$ ms, see Section 2.2) in the future respect to input neuron activity were decoded.

2.3.5. Hyper-parameter search via Bayesian Optimization (BO)

Deep learning-based algorithms are defined by many hyper-parameters that are not a priori known. Therefore, in this study, to identify the optimal configuration of the convolutional feature extractor, automatic hyper-parameter search via BO (Snoek, Larochelle, & Adams, 2012) was adopted. This algorithm was applied to the identification of the optimal hyper-parameter set in the supervised problem 1. Due to the performed data split (10-fold cross-validation scheme) and to the nature of the dataset (2 monkeys and 3 recording areas), BO led to an optimal configuration specific for each fold, monkey, and area (60 configurations in total). Once BO was performed, the most frequent value (across folds, monkeys and areas) of each hyper-parameter was computed and used to design the CNNs to be trained from scratch in both supervised problems 1 and 2. Therefore, this BO-based procedure was used to identify a single configuration of hyper-parameters occurring more frequently across folds, monkeys and areas, i.e. a single functional form of the decoder f that was then exploited to train CNNs in both supervised problems. CNNs were trained — within each BO iteration and while training the most frequent CNN configuration in the supervised problems 1 and 2 – using Adam optimizer (Kingma & Ba, 2014) with a batch size of 64 for a maximum number of 1000 epochs and applying early stopping on the validation loss.

In the following, an overview of automatic hyper-parameter search and of BO is reported. Hyper-parameter optimization is devoted to find the hyper-parameter configuration of a learning system (e.g., a CNN) associated with the best performance measured on a separate validation set. Let us denote with h the array containing the hyper-parameters of interest, with $h \in H$ where H is the hyper-parameter search space. In this study, we investigated the main hyper-parameters defining the convolutional feature extractor N_b (number of blocks), N_c (number of separable convolutional layers per block), K (number of filters per layer), F (filter size), c (max norm constraint), the use of batch normalization, the activation function for the convolutional layers, the pooling function, the dropout probability p and the learning rate. Formally, hyper-parameter optimization consists in finding $h^* = arg\ \min_{h \in H} k(h)$, where $k(h) : H \rightarrow \mathbb{R}$ represents the objective function to be minimized on the validation set (the loss function in this study). To evaluate $k(h)$ for each configuration h the learning system needs to be trained on the training set and then evaluated on the validation set.

Depending on the model complexity (typically high for deep learning-based decoders) and on the number of hyper-parameters to optimize, the evaluation of a trained model on the validation set can be expensive. Common hyper-parameter search algorithms (e.g., grid search or random search), perform many evaluations on the validation set, each one using a trained model with a hyper-parameter configuration based on a pre-defined rule (e.g., by sampling all possible hyper-parameter configurations or by randomly sampling a fixed number of configurations) ignoring the results of past evaluations. This often leads to wasting time in evaluating 'bad' hyperparameters. Bayesian optimization methods overcome this limitation, as they suggest in an informed way the next hyper-parameter configuration to be evaluated, thus, investigating hyper-parameters that seem promising based on past evaluations. Specifically, these methods build a Bayesian statistical model $p(k|h)$ of the objective function, called surrogate probability model, which maps hyper-parameter values to the probability of getting a certain value of the objective function. The surrogate model is formed by keeping track of the past evaluation results and is easier to optimize than the actual objective function $k(h)$; thus, the next set of hyper-parameters to be evaluated on the actual objective function is chosen by selecting the hyper-parameters that perform best on the current surrogate model. Once the surrogate $p(k|h)$ has been initialized, the procedure involves several optimization iterations (100 iterations were performed in this study), run sequentially one after another, with each iteration consisting of the following steps:

i. Optimize the surrogate finding the hyper-parameters that perform best on the surrogate. The criterion used to optimize the surrogate is called "selection function".

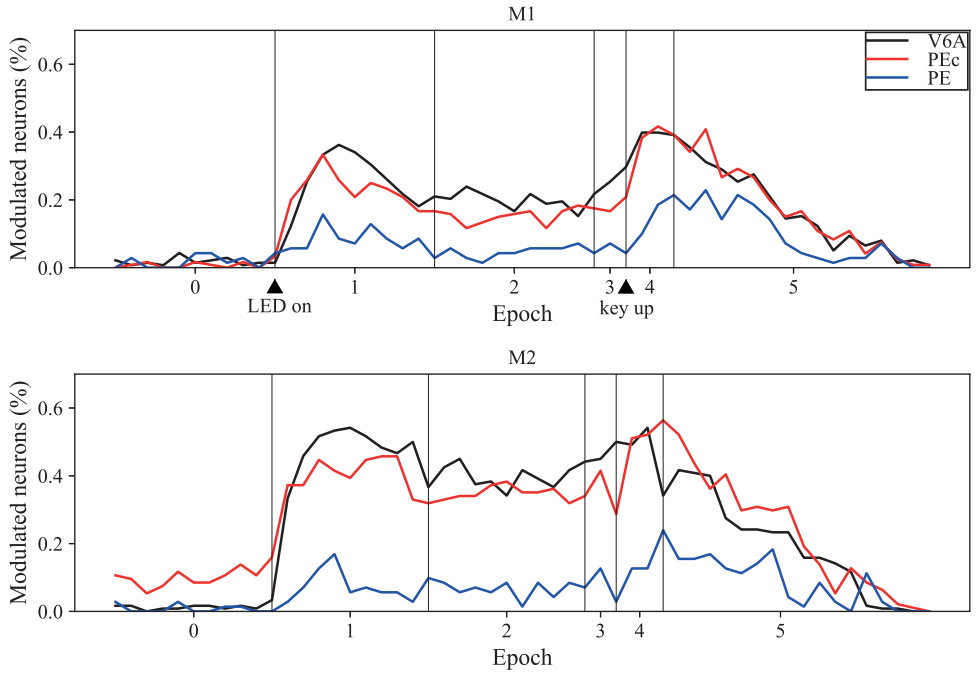

Fig. 3. Percentage of modulated neurons as obtained with a sliding ANOVA. The modulation is reported over time for each monkey (M1 on top and M2 on bottom) and each recording area (V6A black, Pec red, PE blue). Vertical bars denote the separation between epochs 0–5. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

ii. Design the learning system using the hyper-parameters selected at point i. Train the learning system and evaluate the objective function k.
iii. Update the surrogate probability model depending on the history of past evaluations, including the last evaluation result (at point ii).

Different choices exist for the surrogate probability model and criterion function used to optimize it. In this study, as commonly adopted (Bergstra, Bardenet, Bengio, & Kégl, 2011), Tree Parzen Estimator (TPE) and Expected Improvement (EI) were used as surrogate model and selection function, respectively. By applying the Bayes rule, the surrogate probability model can be expressed as $p(k|h) = p(h|k)p(k)/p(h)$, and by using TPE, $p(h|k)$ is modeled as:

$$p(h|k) = \begin{cases} l(h), k < k^* \\ g(h), k \geq k^*, \end{cases} \quad (5)$$

$l(h)$ and $g(h)$ are the distributions of the hyper-parameters, one modeled by using the previously evaluated hyper-parameters that resulted in objective function below the threshold k^*, and the other by using the previously evaluated hyper-parameters that resulted in objective function above the threshold k^*. These distributions are modeled with Gaussian mixture models in TPE. To initialize the algorithm (i.e., initialize the values needed to model the distributions) 20 iterations were performed by randomly sampling the hyper-parameters (i.e., performing random search). Furthermore, the TPE algorithm depends on the threshold k^*, the latter is chosen larger than the lowest observed k so that some points can be used to model $l(h)$. The algorithm selects k^* so that $\gamma = p(k < k^*)$, but no specific modeling for $p(k)$ is needed (Bergstra et al., 2011). In this study, hyper-parameter values were all sampled from uniform distributions (defining $p(h)$) over the values reported in Table 1. Then, the expected improvement (EI, expectation that the surrogate model, by using h, assumes values below the threshold k^*) can be computed as:

$$EI_{k^*}(h) = \int_{-\infty}^{\infty} \max(k^* - k, 0) p(k|h)\, dk = \int_{-\infty}^{k^*} (k^* - k)\, p(k|h)\, dk. \quad (6)$$

In this scenario, the optimization problem (point i.) is reduced to a maximization of the EI. As reported in Bergstra et al. (2011), by expressing $p(k|h)$ using the TPE modeling (Eq. (5)):

$$EI_{k^*}(h) \propto (\gamma + (g(h)/l(h))(1 - \gamma))^{-1} \propto l(h)/g(h). \quad (7)$$

Therefore, maximizing $EI_{k^*}(h)$ corresponds to maximizing the ratio $l(h)/g(h)$, i.e., find the optimal h with high probability under $l(h)$ and low probability under $g(h)$. Then, the true objective function $k(h)$ is evaluated with this optimal h (point ii.) and, subsequently, the two distributions $l(h)$ and $g(h)$ defining $p(h|k)$ are updated depending on the history of the past evaluations by taking into account the result of this last iterations (point iii.).

3. Results

We recorded the activity of single neurons from 3 contiguous areas, V6A, PEc, and PE, in the superior parietal lobule of 2 macaques. In this study we were interested in testing if these three areas encode spatial information about the reaching goal and reaching trajectories with the same strength, and if they encode the temporal dynamics of this encoding. CNNs were used as decoders for the addressed supervised problems, and we were also interested in searching an optimal CNN design via automatic hyper-parameter using BO. In the following, the main results are reported.

3.1. Preliminary data analysis

A simple sliding ANOVA (Fig. 3) was enough to show that the percentage of neurons modulated by spatial position of reaching targets was different depending on the considered area. PE showed half of neurons modulated (with peaks of approx. 20%) compared to V6A and PEc (with peaks between 50%–60%) which were very similar. It is also interesting to note that the percentage

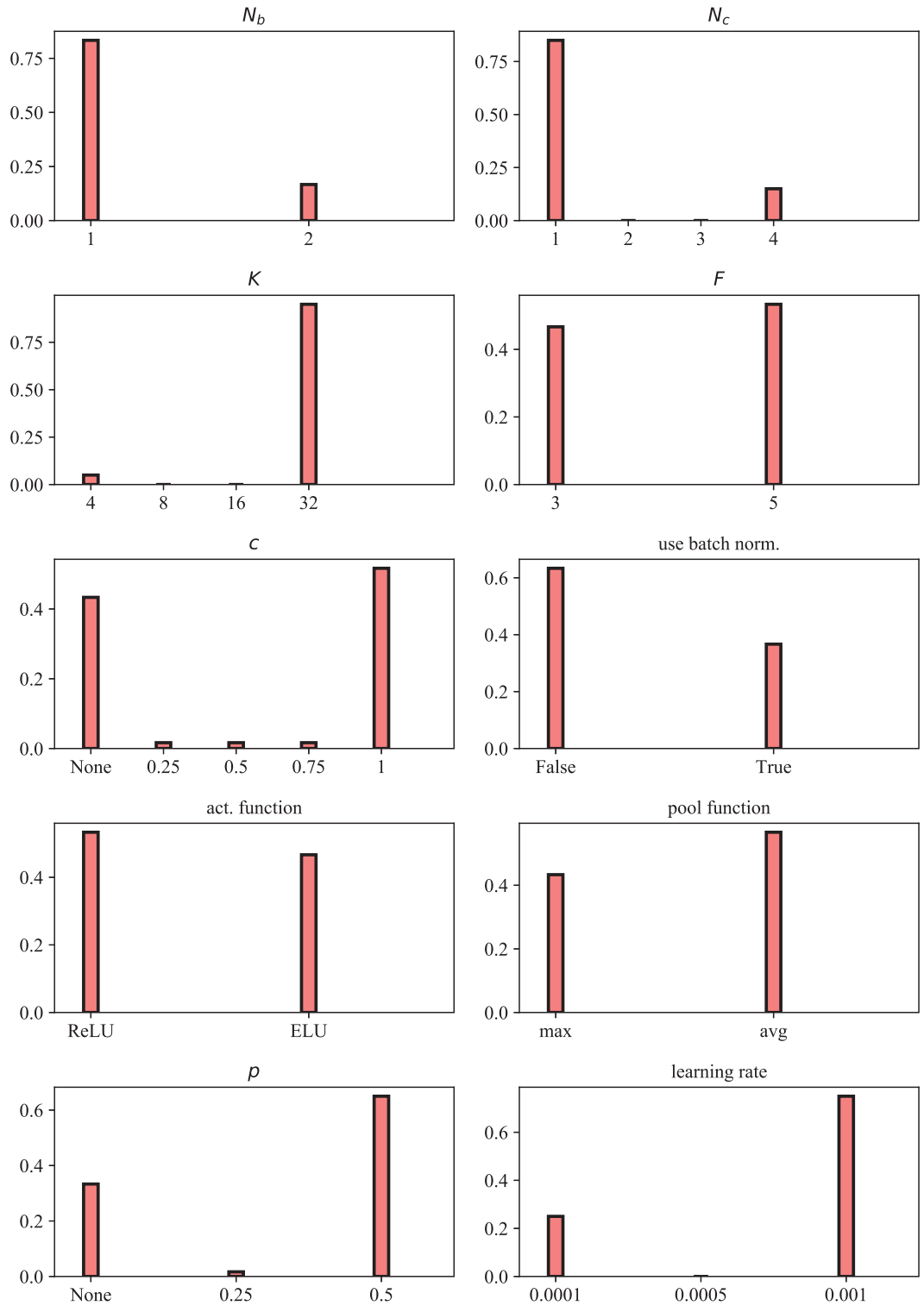

Fig. 4. Hyper-parameter probability distributions of the convolutional feature extractor resulting from BO. On the x-axis the hyper-parameter values are reported, while on the y-axis the probability that BO selected as optimal each hyper-parameter value is reported. N_b and N_c denote the number of convolutional blocks and temporal separable convolutions per block, respectively. K and F denote the number of temporal kernels and the kernel size, respectively. Lastly, c and p indicate the maximum norm to use in max-norm constraint and the dropout probability, respectively.

of modulated cells was not stable over time but roughly was characterized by 2 prominent peaks in all areas: a first increase of modulated cells in the first phase of target presentation (Epoch 1), a second peak during the execution of the reaching movement (Epoch 4). These differences prompted us to decode the overall dynamics, which also reflect non-linear interactions and temporal aspects. Such dynamics can be shown with deep learning-based decoders such as CNNs and are reported in the following sections.

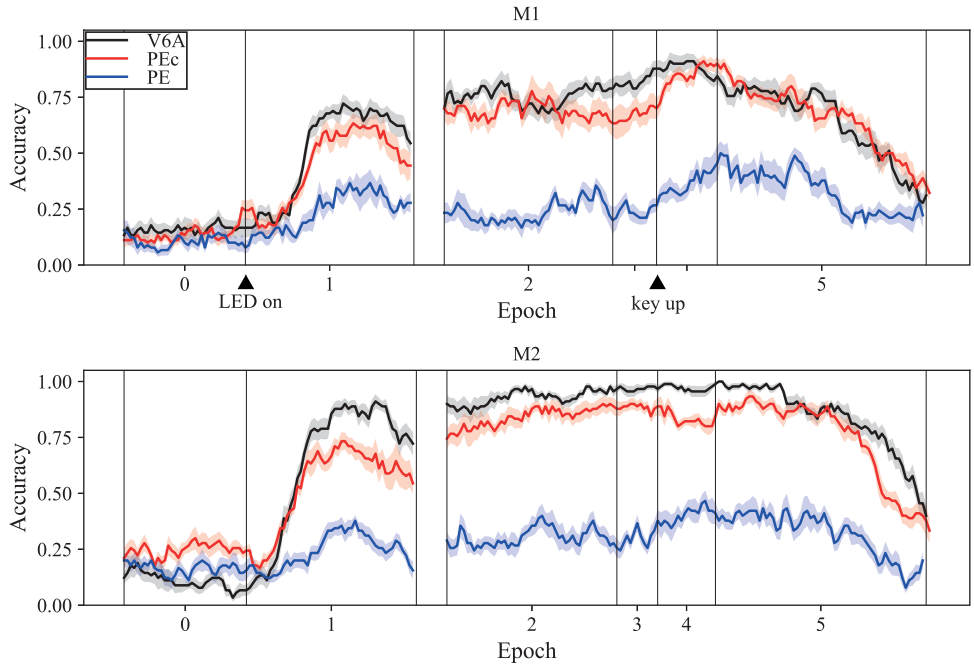

Fig. 5. Decoding accuracy over time. The performance metric obtained in the supervised problem 1 for each monkey (M1 on top and M2 on bottom) and each recording area (V6A, PEc, PE) used in the decoding is reported as a function of time. Mean values are reported (thick lines) ± standard error of the mean (overlayed shaded areas) across folds. Vertical bars denote the separation between epochs 0–5.

3.2. Optimal convolutional feature extractor

The convolutional feature extractor of the adopted CNN underwent automatic hyper-parameter search via BO in the supervised problem 1. Convolutional filters slide over the temporal dimension catching temporal patterns in firing rates; each training sample included 15 temporally consecutive bins, each bin represented the firing rate of a time interval of 20 ms (see Section 2.3.1). In Fig. 4 the distributions of the searched hyper-parameters are reported in cumulative histograms, considering all the optimal configurations across folds, monkeys, and areas (60 configurations in total). From these results the most frequent configuration (higher bar on each plot corresponding to a given hyper-parameter in Fig. 4) was a simple shallow CNN, characterized by a number of blocks $N_b = 1$, number of convolutional layers per block $N_c = 1$, number of convolutional filters $K = 32$, filter size $F = 5$, and max norm $c = 1$. Furthermore, options such as no batch normalization, ReLU activation functions for hidden units, and average pooling were more frequently adopted. Lastly, a dropout probability $p = 0.5$ and a learning rate of 0.001 were optimal. The CNN defined by this specific hyper-parameter configuration was used to solve the decoding problems 1 and 2, and the subsequent reported results are related to this specific functional form of the decoder.

3.3. Supervised problem 1: Target decoding

In Fig. 5 decoding accuracies are reported as a function of time, for both monkeys and all areas. CNNs learned to accurately map the activity of the collected neurons to the spatial location of the targets, as demonstrated by the average accuracy well above the chance level (11%) in both monkeys and in each area. Accuracies began to increase with target detection (epoch 1), remained sustained with a ramping trend during movement preparation (epoch 2), peaked during movement execution (epoch 4), and then began to decline as touch on the target was maintained (epoch 5). In the case of V6A and PEc a maximum accuracy in decoding the correct target position above 80% was reached in epoch 4. Although the trend was similar for the 3 areas, PE had significantly lower accuracies than V6A and PEc, varying between 20%–40%. This lower accuracy of decoding in area PE could be due to the lower percentage of modulated cells in PE within each epoch (see Fig. 3) and/or to the smaller population available (roughly 1/3 of neurons were available for PE with respect of V6A and PEc, see Section 2.1.1). To better explore this last point, a dropping analysis (see Section 2.3.3) was applied, and its results are reported in Fig. 6. Here, accuracies as a function of the number of cells (N_d) used to decode the reaching targets are reported for each monkey and each recording area. For V6A and PEc, few neurons were enough to obtain accuracies well above the chance level, e.g., from 7 sampled neurons accuracies >30% were achieved both in monkey 1 and 2 (M1 and M2). Furthermore, the initial slope of the curve 'accuracy vs. number of cells' followed V6A > PEc > PE. Lastly, in V6A and PEc the trend kept improving more than PE as the number of neurons available increased. The dropping analysis therefore confirmed a lower ability to decode spatial information from area PE neurons compared to the better performance on V6A and PEc. These last two areas appear very similar in their ability to encode the information of the target position with the same strength.

Finally, Fig. 7 shows the comparison between the decoding accuracy as a function of time obtained with the CNN-based decoder and the NB-based decoder. Overall, our proposed decoder exhibited higher accuracy scores. Moreover and interestingly, the time pattern of accuracy during the reaching movement differed across the two classifiers. The NB classifier after the initial increase in epoch 1, tended to exhibit an about constant (or slightly decreasing) accuracy across the other epochs, declining in epoch 5, and did not show the increasing trend peaking in the movement and hold phases as the CNN classifier. Statistical

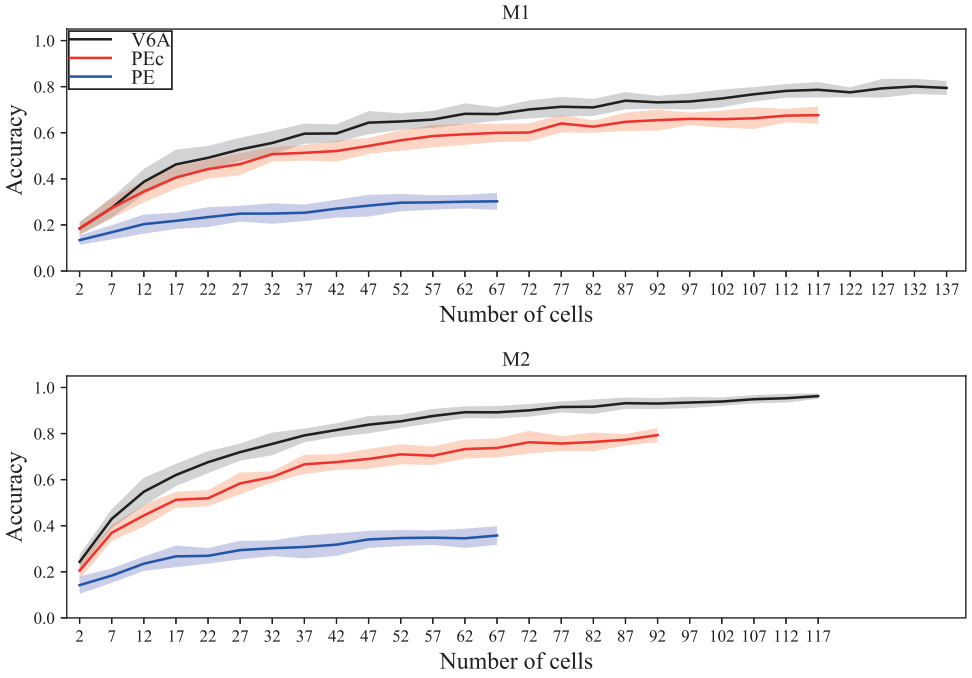

Fig. 6. Neuron dropping analysis. The decoding accuracy in the supervised problem 1 for each monkey (M1 on top and M2 on bottom) and each recording area (V6A, PEc, PE) used in the decoding is reported as a function of the number of cells used to classify reaching targets. For each step of the dropping analysis, the performance was averaged across folds. The figure reports the mean values (thick lines) ± standard error of the mean (overlayed shaded areas) across the 20 random samplings.

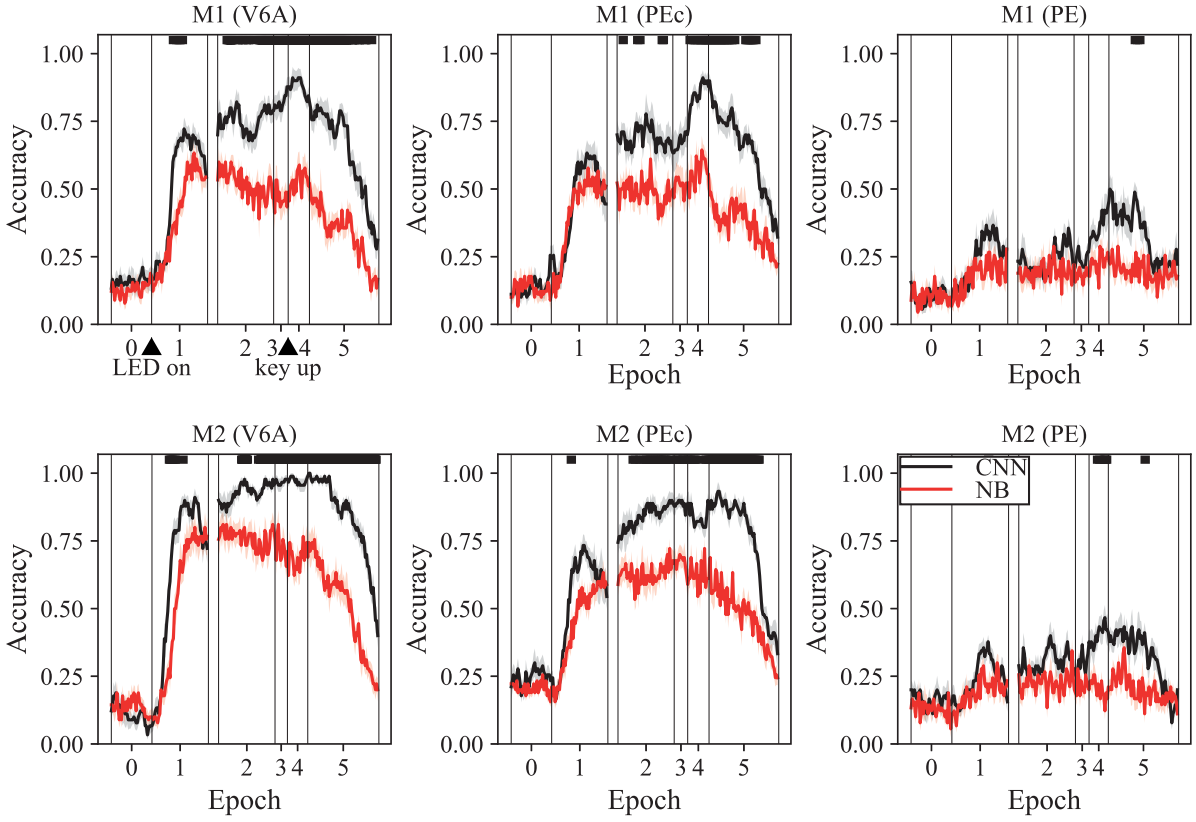

Fig. 7. Decoding accuracy over time obtained with the proposed CNN and with a NB linear classifier. The accuracy scored with both the CNN (black) and NB (red) in the supervised problem 1, for each monkey (M1 and M2) and each recording area (V6A, PEc, PE), is reported as a function of time. The figure reports the mean values (thick lines) ± standard error of the mean (overlayed shaded areas) across folds. Vertical bars denote the separation between epochs 0–5. Permutation cluster t tests were performed for each monkey and each recording area to analyze differences between the two algorithms; temporal intervals with significant performance differences ($p < 0.05$) between the two algorithms are reported on top of each panel with thick horizontal bars.

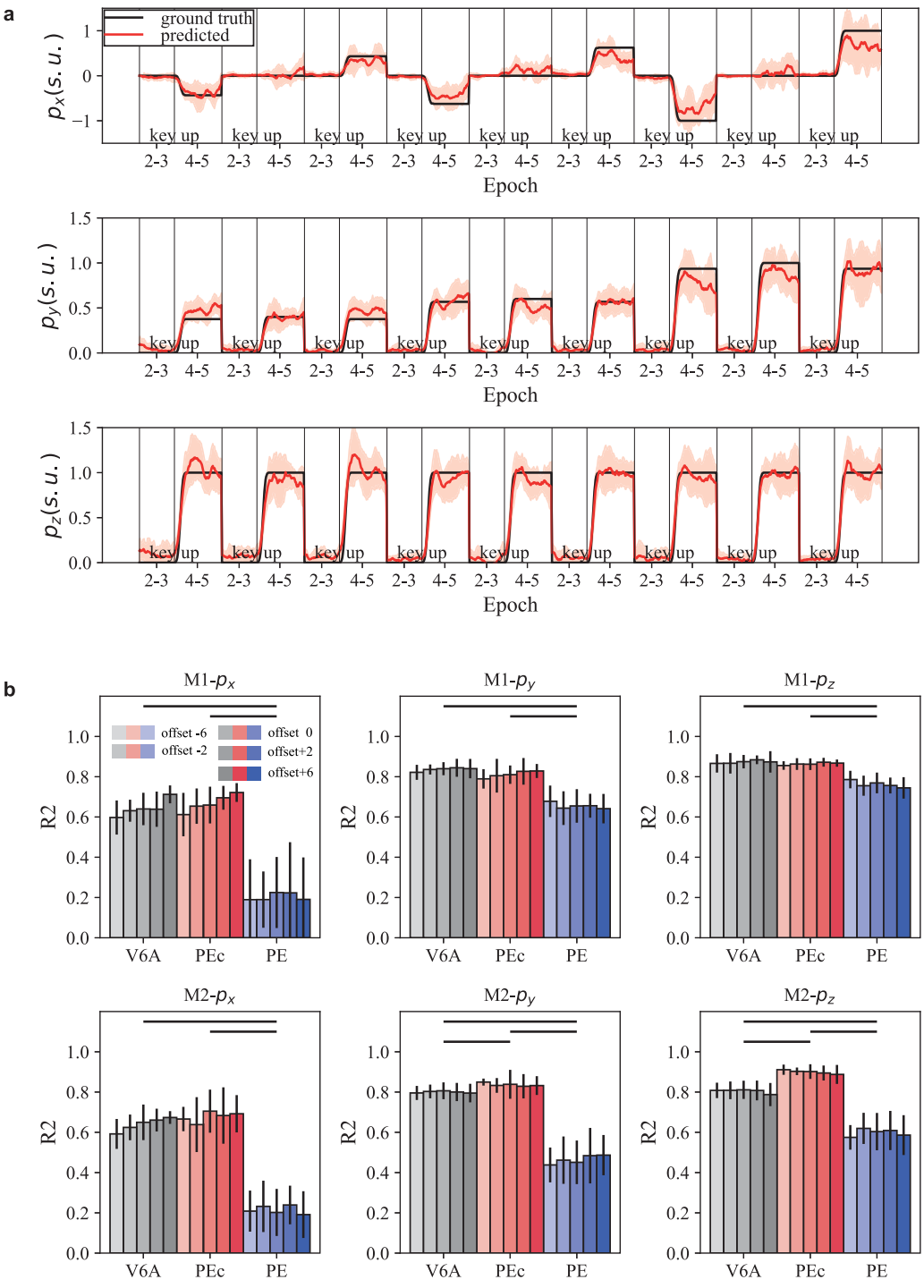

Fig. 8. Representative example of 3-D decoded trajectory (a) and R2 regression scores (b). (a) A representative example of decoded trajectory (red) alongside with the ground truth (black) over time (epochs 2–5) using V6A neural activity from M2 is reported. Coordinates are reported in their scaled units (s.u., see Section 2.3.4). Mean and standard deviations (shaded area) are calculated over cross validations. (b) The performance metric obtained in the supervised problem 2 while decoding position coordinates is reported for each monkey (M1 on top and M2 on bottom) and recording areas (V6A, PEc, PE) used in the decoding. Each plot reports R2 also as a function of the offset chosen while associating the target label (see Section 2.3.4), where the default association is denoted by offset = 0. R2 scores are reported in their mean values (bar height) ± standard deviation (vertical line) across folds. Horizontal black bars connect the areas found with significantly different R2.

analysis shows that CNN outperformed ($p < 0.05$) the NB-based decoder for all monkeys and areas, especially after the movement onset (e.g., during epoch 4 and 5), with improvements up to 46% (in M2 decoding from V6A, during epoch 5). These differences may arise from the capability of the CNN to learn non-linear dependencies exploiting complex hierarchical features in the temporal domain from the input temporal samples, while the classic linear algorithm based on NB is plausibly unable to catch these dynamics as it linearly combines the inputs and assumes conditional independences between the input temporal samples.

In this problem, we reconstructed discrete spatial positions, targets of the reaching. The 9 positions (9 classes) were recognized by CNN with high accuracy during all time intervals in which the animal was aware of the reaching position, from the

first stages of target fixation to the holding of touch on the target. Besides decoding target position, we also tested whether and to what extent other movement aspects, such as hand trajectories, could be potentially decoded from the activity of neurons in the investigated PPC areas.

3.4. Supervised problem 2: Hand trajectory decoding

A representative result while decoding position coordinates using V6A signals is reported in Fig. 8a (with an offset = 0 in the kinematic association, see Section 2.3.4). Epochs 2–5 for the 9 targets are concatenated one after the other to show the range of x, y and z along the whole movement executed by the monkey. Predictions are plotted against ground truth trajectories. In this example, it is clear how V6A strongly contains enough information to accurately decode movement 3-D trajectory. However, not all areas allowed to reconstruct the movement trajectory with the same fidelity. In agreement with previous results as to target position decoding, area PE contains less useful information for the reconstruction of the trajectory, as shown by Fig. 8b reporting the CNN performance metric for each monkey and recording area. Decoding performance using the default kinematic assignment (i.e., $o = 0$, see Section 2.3.4) of the x-coordinate in V6A was significantly lower (for both monkeys, $p < 0.01$, Wilcoxon signed-rank test corrected for multiple tests using the Benjamini–Hochberg procedure (Benjamini & Hochberg, 1995) than y- and z-coordinates, but no differences were found between y- and z-coordinates in the same area. This may be multifactorial, in part given by a methodological drawback, the greater spread of the x (between −1 and 1) with respect to y and z (both between 0 and 1), but has also physiological substantiation. Indeed, previous studies (see Section 4.2) have demonstrated a prominent encoding of the depth component in areas of the superior parietal lobule, and this could be the cause of a greater difficulty in reconstructing the x component of trajectories.

In Fig. 8b, decoding performance was reported also as a function of the offset used in the multi-lag kinematic association adopted in this study, i.e., ±6, ±2, 0 (see Section 2.3.4). This has allowed us to probe whether feedback or feedforward information is prominent in the areas: in the case of negative offsets a possible increase in decoding accuracy indicates the importance of feedback information, i.e. the current activity of the neurons explains an event that has already happened; in the case of positive offsets the increase in accuracy is related to a greater prominence of the elaboration of the movement plan that will be realized in subsequent times. While evaluating differences across offsets for each coordinate, area and monkey, significant differences ($p < 0.01$, Friedman test) across offsets were only found for V6A for the x-coordinate for both monkeys, where a positive linear trend from negative to positive offsets was observed. By focusing only on the performance obtained with zero-lag, we observed that PE scored lower R2 than other areas ($p < 0.01$, Wilcoxon signed-rank test corrected for multiple tests using the Benjamini–Hochberg procedure), for all monkeys and coordinates (see statistical analysis results displayed as solid bars in Fig. 8b).

4. Discussion

In this study we explored the possibility to predict reaching goals and movement trajectories by decoding the activity of single neurons recorded from different areas of the SPL. To decode neural signals, CNNs were used to extract temporal dynamics via temporal convolutions, and their design was defined by automatic hyper-parameter search using BO. This is particularly relevant as in the literature (Glaser et al., 2020; Sussillo et al., 2012; Tseng et al., 2019) DNN designs are commonly defined by empirical evaluation (i.e., test a limited bunch of configurations and using the best performing one) which is a time-consuming process, often leading to sub-optimal DNN configurations. CNNs were used in this study with two main purposes: (i) leveraging temporal convolution to better characterize encoding dynamics in the different areas of SPL reaching network; (ii) validating for the first time (to best knowledge of the authors) CNNs in decoding spiking activity, as CNNs among DNNs (e.g., respect to RNNs) achieve good performance and explainability, while keeping a lower number of trainable parameters. Finally, we probed CNNs superiority compared to simpler linear classifiers. Decoding results allowed to uncover the amount and characteristics of information each area contains about the external variables (endpoint position and effector trajectory) and the differences across the different areas.

Signals collected from neurons of SPL were predictive of reaching targets also before the movement onset. Decoding performances were different between areas: while the decoding accuracies for V6A and PEc were very similar (over 80% of accuracy in detecting the correct spatial position), PE diverged by obtaining lower values (lower than 40%), suggesting a different role in the circuit for the different areas. Similarly, moving from decoding reaching goals to movement trajectories, V6A and PEc maintained good R2 values, PE regained some points in the reconstruction of the depth (y) and elevation (z) components of the movement while reconstruction of the direction (x) remained more difficult. Neural networks have proven to be robust and easily adaptable to the required task, preferring simple architectures (see the more frequent optimal configuration in Section 3.2) and therefore quickly trainable and generalizable, being able to capture the dynamics of neuronal activity.

4.1. A visual to somatosensory gradient over the network is reflected in the decoding accuracy

The posterior parietal cortex sits in a crucial node for integration of sensory stimuli to guide action receiving visual input, somatosensory and proprioceptive feedback, and afferent motor copy from premotor cortex (Andersen & Cui, 2009b; Gamberini, Passarelli, Fattori, & Galletti, 2020). In agreement with this, electrophysiological studies that probed different areas of the PPC found different levels of activation for these neurons, mostly linked to visual stimuli for caudal regions close to extrastriate cortex, tactile and proprioceptive moving rostrally towards somatosensory cortex. The network we studied bridges visual and somatic domains, with area V6A bordering the extrastriate visual area V6 (Galletti et al., 1995; Matelli, Govoni, Galletti, Kutz, & Luppino, 1998), PE bordering Brodmann's area 2 and PEc in between (Breveglieri et al., 2008). In agreement with this, the number of neurons modulated by the spatial goal position (see sliding ANOVAs of Fig. 3) was already high in V6A in the first part of the trial (epoch 1), being the visual input of great relevance in V6A. The number of neurons modulated by goal position is maximal in PE in the second part of the trial, that was associated with the execution of the reaching movement (epoch 4–5). Although even in the second part of the task the percentage of modulated neuron of PE was not exceeding those of V6A and PEc, the stronger representation of the goal position in the movement epoch reflects a stronger somatomotor than visuospatial representation in PE. The neural activity from the first visualization of the target to the movement onset has been associated with the elaboration of the motor plan, the maintenance of spatial attention towards relevant targets and the integration of sensorimotor inputs (Andersen & Cui, 2009b; Galletti & Fattori, 2018).

Moving from the previous ANOVA analysis, we then performed a more sensitive decoding analysis using CNN; the latter, taking into account neural dynamics and non-linearity, can guide a more robust comparison. Given the link between preparatory activity and the spatial positions of the targets, decoders successfully extracted the correct position given the preparatory spiking activity. Our results suggest that V6A and PEc strongly encode visuospatial information already in the first part of the task, enabling the decoders to easily extract useful information about the goal spatial positions. During the planning epoch, visuospatial inputs are converted to visuomotor signals which are required to guide execution phase. During this second part of the task, signals related to goal position remained strong and easily decodable. PE in comparison, presents a less pronounced activation in relation to the task tested, nevertheless during the execution phase stronger sensorimotor signals can be decoded by CNNs. It is worth noticing that gaining accuracy in the movement phases, in all areas and especially in PE, is a peculiarity of the CNN classifier, not exhibited by the linear NB classifier (see Fig. 7); this may indicate that CNNs are more apt to catch the richness of information contained in these areas. Ultimately, PE and PEc both part of the Brodmann's area 5 (Pandya & Seltzer, 1982) appear more different than expected: the activity of PEc is much more similar to V6A, part of Brodmann's area 7, and should therefore be considered part of the latter area 7. This idea, advanced by Gamberini et al. (2020) is supported by present data.

4.2. Decoding movement goals and trajectories from PPC

Decoding of reaching goals is particularly efficient in different areas of the PPC especially from areas of the dorso medial network. The Parietal Reach Region (PRR) has been used as a source of these signals in several studies (Andersen, Hwang, & Mulliken, 2010; Hauschild, Mulliken, Fineman, Loeb, & Andersen, 2012; Mulliken, Musallam, & Andersen, 2008b; Musallam, Corneil, Greger, Scherberger, & Andersen, 2004). Interestingly, the PRR is close to areas V6A and PEc and mostly overlap with area MIP (anterior bank of the medial intraparietal sulcus) (Snyder et al., 1997). Homologue of PRR in humans, together with the more lateral anterior intraparietal area (AIP), were used to decode motor imagery in a center-out task by a tetraplegic implanted patient (Aflalo et al., 2015). Thus, the interest in decoding movement intention from PPC remains high in light of the possibility to extract several parameters related to cognitive processing rather than simpler motor kinematics (Andersen, Kellis, Klaes, & Aflalo, 2014). Most of the studies used a task with reaching movements towards a monitor placed in front of the subject, without studying the movement in depth. Conversely, in our study the movements were made on three different degrees of depth simulating more naturalistic movements. Several pieces of evidence support the diversity of networks processing direction and depth information, with different percentages of cells modulated in depth and direction for the different areas tested (Crawford, Henriques, & Medendorp, 2011; Hadjidimitrakis et al., 2014, 2015; Tramper & Gielen, 2011). Depth encoding is plausibly stronger for areas that rely more on proprioceptive (such as PE) rather than visual (De Vitis et al., 2019). We did not find different accuracies by decoding the two components separately when we decoded reaching goals (data not shown). While decoding the 9 reaching targets (supervised problem 1), not only movement attributes, planning and execution could be exploited in the learning system, but also spatial attention, sensory feedback, and movement imagery. All these types of information are known to be encoded in PPC (Section 4.1 and contribute to the generation of patterns in the discharge of neurons, patterns that can be extracted from the neural network and mapped to the classes corresponding to spatial locations. We then tried to predict the trajectories of hand position (supervised problem 2) from the population activity of neurons providing the algorithm with past neural activity up to 300 ms before the current movement (corresponding to an offset = 0, see Section 2.3.4 and Fig. 8b). Unfortunately, the real trajectories were not available, so we have reconstructed the plausible trajectories of movement semi-synthetically (see Section 2.3.4), from the pressure of the home button to the reaching movement up to the holding of goal position. Remarkably, while V6A and PEc R2s were high (over 0.6) obtaining a good trajectory reconstruction, R2s of PE were lower (see Fig. 8b), especially for the x-coordinate (corresponding to the direction of movement) supporting the view of preferential depth (y) encoding from rostral SPL. Our task was ideal for testing visuospatial transformations and probably little activates areas more devoted to somatomotor control. Nevertheless, it is plausible that the semi-synthetic trajectories we used, forcing a non-natural straight-line trajectory, could be not optimal for decoding. While reaching goal location and trajectory decoding were good for both V6A and PEc, and no particular difference emerges between the two methods, trajectory decoding from PE seems to perform slightly better than classification (especially for depth and elevation). This could be related to the specialization of the area PE in dealing with proprioceptive signals, so that information of the absolute position of the target (used for classification problem) is scarcely useful, and signals are more related to the movement of the limbs than visuospatial representation.

4.3. Feedforward model

To perform rapid, targeted movements, our brain must rely on a feedforward predictive model given the latency of incoming sensory feedback signals. One of the theories that is gaining momentum is that the brain must continuously integrate the state of the environment and the body into a feedback control loop to perform congruent movements in real time (optimal feedback control, Todorov, 2004). The SPL fits nicely into this framework with caudal regions encoding the environment in relation to the body and more rostral regions encoding the state of the body (Medendorp & Heed, 2019). Within this model it is possible to frame the generation and deployment of the trajectories of movement. We found that it is possible to predict the instantaneous position of hand in the space by providing the activity of neurons in a short prior interval, but also with increasing lags. Negative lags extract features related to sensory feedback (sensory outcomes of action), positive lags suggest the existence of a predictive model or motion planning. Mulliken and colleagues (Mulliken et al., 2008a) observed how PRR neurons encode for either the movement angle or goal angle (or both), finding single-cell preferences for encoding future states (positive lag), past states (negative lag), and many cells that represented the current state (zero lag, in particular for a task that used an obstacle on the trajectory requiring a more dynamic control), demonstrating the existence of the feedforward model. Our decoding analysis loses sensitivity to a single neuron combining all contributions at the population level but still gives us clues about the existence along the entire network of the SPL of the running feedforward model. The fact that the areas we studied simultaneously contain a representation of sensory feedback and signals related to the planning of future movements, supports the existence of an inner model that compares the expected outcome of an action with the real outcome, even if with intrinsic latency. While in most cases the decoding accuracy was not affected by the different offsets, only the x component for the V6A area shows significant dependence (Fig. 8b). We think that this reflects two properties of the network we are studying. Previous work from our lab found that direction

Table 2
Hyper-parameter space of the convolutional feature extractor. N_b and N_c denote the number of convolutional blocks and temporal separable convolutions per block, respectively. K and F denote the number of temporal kernels and the kernel size, respectively. Lastly, c and p indicate the maximum norm to use in max-norm constraint and the dropout probability, respectively. The hyper-parameter values were sampled using uniform distributions during the hyper-parameter optimization. Among the values, "None" denotes no usage of a specific technique (i.e., no use of kernel max norm constraint and no use of dropout).

Hyper-parameter	Values
N_b	[1, 2]
N_c	[1, 2, 3, 4]
K	[4, 8, 16, 32]
F	[3, 5]
c	[None, 0.25, 0.5, 0.75, 1]
Use batch norm.	[False, True]
Activation function	(ReLU, ELU Clevert, Unterthiner, & Hochreiter, 2016)
Pool function	[max, avg]
p	[None, 0.25, 0.5]
Learning rate	[0.0001, 0.0005, 0.001]

information (x-axis) is processed earlier than depth (y-axis, see Hadjidimitrakis et al., 2014) that relies more on somatosensory signals arriving later as sensory feedback from the moving arm. In addition, somatosensory afferents are greater for more rostral areas (PEc and PE, see Introduction). Accordingly, V6A decoding accuracy is more affected by offsets that rely less on feedback signals (see lower accuracy for negative offset in Fig. 8b), where a high accuracy is maintained using movement preparatory signals (positive offsets). The offset effect is less evident in PEc, with a trend that is not statistically significant and in PE it is not present (although it could be masked by the lower overall accuracy).

4.4. Convolutional neural networks for neural decoding

The revival of neural networks supported by a large variety of applications in computer vision has led to the widespread use of neural networks in various fields that can now benefit from advanced pattern recognition techniques (Richards et al., 2019). We have borrowed techniques from time series analysis and have shown how neural networks are well suited for the study of neural dynamics. Since our CNN-based algorithm is able to extract and leverage temporal features, the decoding performance significantly increases (Fig. 7) compared to a classic algorithm based on Naïve Bayesian.

Conversely to other studies (Glaser et al., 2020; Shah et al., 2019; Sussillo et al., 2012), in this study we adopted CNNs to decode neural signals while automatically searching for its best configuration using BO. Therefore, it is worth remarking that the main CNN structural hyper-parameters were automatically optimized within the search space exploiting an automatic search algorithm, rather than manually select them based on a trial-and-error procedure. From BO, a shallow CNN architecture (i.e., $N_b = 1$, $N_c = 1$) with one separable convolutional layer and one fully-connected layer resulted optimal for decoding the neural activity during reaching. The adoption of a shallow CNN has the advantage of lower training times and good generalization with limit-sized datasets as the one adopted in this study, achieving high decoding performance both in classifying the target reaching end-point, and in predicting semi-synthetic trajectories. Indeed, results suggest that with the adopted optimal shallow CNN few trials (72 in the training set, see Section 2.1.2) are enough to train the networks with the chunking procedure (augmenting the training set up to 3384 examples, see Table 1), achieving high performance. Temporal patterns of the single separable convolutional layer led to an optimal decoding when extracted within a window of 100 ms (i.e., $F = 5$ bins). Interestingly, despite being a shallow CNN, the optimal architecture learned the highest number of features among the admitted values of hyper-parameter space (i.e., $K = 32$ feature maps were learned, see Table 2). Thus, instead of selecting a deep convolutional neural network (e.g., 2 blocks with 4 convolutional layers per block, corresponding to the maximum depth in the defined search space) and a low number of filters per layer (e.g., 4 filters), which is a common design principle in computer vision applications (Simonyan & Zisserman, 2015), BO selected a simple shallow CNN learning the highest possible number of filters in the single convolutional layer included.

Therefore, by analyzing these structural hyper-parameters, results suggest that the information contained in the input neural chunks did not require extracting high-level and more abstract features (e.g., as resulting from a deep CNN in deeper layers) to perform an accurate decoding, but rather learning many low-level and less abstract features directly from the raw input chunks was more beneficial. This result may depend on the fact that some high-level features have already been extracted upstream, from the visual and somatosensory processing flows that precede the posterior parietal lobule; i.e., the inputs to our network are not directly taken from the external world, as usually done in deep neural networks, but have been significantly pre-processed by the primary brain areas. Future applications in neural decoding could benefit in designing shallow but wide CNNs rather than deep and narrow CNNs. Furthermore, regarding regularization hyper-parameters, constraints such as kernel max-norm constraint (with $c = 1$) and dropout (selecting the highest dropout probability, i.e., $p = 0.5$, see Table 2) proved their utility improving the generalization in the addressed decoding tasks. Interestingly, batch normalization did not result as useful as the previous regularization methods (BO selected less frequently this regularizer). Thus, in perspective, neural decoders could benefits in applying the specific combination between kernel max-norm constraint and dropout (with a high dropout probability, e.g., set to 0.5 as in this study) to perform regularization.

Despite the main objective of this study was to propose a CNN architecture for decoding neural activity and enabling the analysis of three different PPC areas, some methodological aspects may prospectively have significant implications for BCI. First, the proposed CNN structure resulted from an automatic algorithm (BO) and resulted optimal in terms of performance on a separate validation set, significantly outperforming a linear classifier. Furthermore, the design included separable convolution that are lighter and more efficient than standard convolutions, Chollet (2017), providing a neural network that is less prone to overfit small datasets and that produces a fast inference. That is, the proposed CNN resulted in an accurate, light, and efficient non-linear decoder of neurons' spiking rate that, in prospective, may find some applicability in BCI systems.

4.5. Future directions

Although we focused on CNNs in our analysis because we think they should be better explored as method for neural decoding thanks to their simplicity and interpretability (Tjoa & Guan, 2020), many studies used different implementations based on RNNs (as mentioned in the introduction) obtaining good results especially in the decoding of trajectories for practical applications (despite RNNs remain in many cases black boxes). RNNs are being trained to simulate the frontoparietal network of grasping (dorso-lateral pathway rather than dorso-medial of reaching), with artificial units resembling the neural activity of real neurons recorded in areas of grasping circuits. In such model, virtual lesion to the artificial network produced outputs similar to lesion/inactivation studies on monkeys (Michaels, Schaffelhofer,

Agudelo-Toro, & Scherberger, 2020). Given the interconnections between the reaching and grasping networks with neurons sensitive to grip types found in V6A (Fattori et al., 2010) and neurons sensitive to reach locations in anterior intraparietal area (Lehmann & Scherberger, 2013) it is expected that a similar RNN architecture can also be applied to the dorso-medial network. Future developments include the adoption of RNN to compare with CNNs in a benchmark (with many datasets and decoding algorithms) and evaluate which decoding approach represents the best compromise between performance, training time and model size (i.e., number of trainable parameters). Furthermore, techniques aimed to improve the interpretability of the CNN (e.g., occlusion techniques and saliency maps Simonyan, Vedaldi, & Zisserman, 2014), recently exploited to investigate neural signatures in the electroencephalogram while decoding brain states (Borra, Fantozzi, & Magosso, 2020, 2021), can be of value also to explain network's decision when decoding neurons' spiking rates. In particular, these explanation techniques could help to characterize the impacts of individual input neurons or subpopulations of neurons (e.g., at different locations) inside each specific area in the decoding process, as well as the importance of specific time bins, contributing to understand their role at the level of brain network dynamics. Working on these two points, i.e., developing algorithms that decode efficiently and accurately neural dynamics, and explaining decoding decision could bring great benefits. First, it may further increase our knowledge about the link between neural activity and behavioral outcome; second, and prospectively, it may contribute to advance BCI technologies by driving improvements aimed to maximize brain information extraction and better brain–computer communication.

Finally, it is important to stress that neural decoders as the ones proposed here are of significance to determine the amount and nature of information neural populations contain about specific external variables, but are not designed for mechanistic interpretation, i.e., for explaining the neural mechanisms underlying multisensory and sensorymotor integration in PPC for guide actions. To this aim, biologically inspired neural networks are needed, designed to functionally and structurally resemble specific parts of the brain and to implement more biological learning rules than back-propagation. Data-driven deep learning approaches and biologically constrained interpretative networks are complementary approaches that can both boost a better comprehension of how information is encoded and processed in the brain and each one can support the advancement of the other; for example, a better description of the information encoded by the different neural populations gained by CNNs decoders may guide the design of interpretative models (Kay, 2018).

5. Conclusion

We decoded the activity of neurons from three areas of the reaching network within the superior parietal lobule of macaque, V6A, PEc, and PE to reconstruct the position of the goal in space and the trajectory required to accomplish the reaching. CNNs were used as neural decoders and proved to accurately decode both the reaching target and 3D hand position. The optimal design of the CNNs, as obtained with hyper-parameter search, resulted in shallow (but wide) architectures with only one hidden separable convolutional layer. While the more caudal V6A and PEc encoded more strongly the position of the target in space (decoding accuracy was already good at the presentation of the target) the area PE, more rostral, was weaker in this representation, its accuracy reaching a peak during the execution of the movement. This supports a model of the PPC where the more caudal areas represent the body-environment relationship and the more rostral areas the effects of the action on the body. The results can be framed in the role played by PPC in the neural control of reaching movements. New generations of BCIs can gain benefits from a better combined study between system neuroscience and renewed deep learning technologies.

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.

Funding

This work was supported by MAIA project, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 951910. This article reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

References

Aflalo, T., Kellis, S., Klaes, C., Lee, B., Shi, Y., Pejsa, K., et al. (2015). Neurophysiology. Decoding motor imagery from the posterior parietal cortex of a tetraplegic human. *Science*, *348*, 906–910, Available at: http://www.ncbi.nlm.nih.gov/pubmed/25999506.

Andersen, R. A., & Cui, H. (2009a). Intention, action planning, and decision making in parietal-frontal circuits. *Neuron*, *63*, 568–583, Available at: http://linkinghub.elsevier.com/retrieve/pii/S0896627309006394.

Andersen, R. A., & Cui, H. (2009b). Intention, action planning, and decision making in parietal-frontal circuits. *Neuron*, *63*, 568–583, Available at: http://linkinghub.elsevier.com/retrieve/pii/S0896627309006394.

Andersen, R. A., Hwang, E. J., & Mulliken, G. H. (2010). Cognitive neural prosthetics. *Annual Review of Psychology*, *61*, 169–190, C1-3 Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2849803&tool=pmcentrez&rendertype=abstract.

Andersen, R. A., Kellis, S., Klaes, C., & Aflalo, T. (2014). Toward more versatile and intuitive cortical brain-machine interfaces. *Current Biology*, *24*, R885–97, Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4410026&tool=pmcentrez&rendertype=abstract.

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*, 289–300, Available at: https://onlinelibrary.wiley.com/doi/10.1111/j.2517-6161.1995.tb02031.x.

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems 24: 25th annual conference on neural information processing systems 2011, NIPS 2011*.

Borra, D., Fantozzi, S., & Magosso, E. (2020). Interpretable and lightweight convolutional neural network for EEG decoding: Application to movement execution and imagination. *Neural Networks*, *129*, 55–74, Available at: https://linkinghub.elsevier.com/retrieve/pii/S0893608020302021.

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*, 304, Available at: https://www.frontiersin.org/articles/10.3389/fnhum.2021.655840/full.

Breveglieri, R., Galletti, C., Monaco, S., & Fattori, P. (2008). Visual, somatosensory, and bimodal activities in the Macaque Parietal Area PEc. *Cerebral Cortex*, *18*, 806–816, Available at: https://academic.oup.com/cercor/article-lookup/doi/10.1093/cercor/bhm127.

Breveglieri, R., Hadjidimitrakis, K., Bosco, A., Sabatini, S. P., Galletti, C., & Fattori, P. (2012). Eye position encoding in three-dimensional space: integration of version and vergence signals in the medial posterior parietal cortex. *The Journal of Neuroscience*, *32*, 159–169, Available at: http://www.jneurosci.org/content/32/1/159.abstract?ijkey=238567af5317414928c59cb0ab18c24dac5f559d&keytype2=tf_ipsecsha.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings - 30th IEEE conference on computer vision and pattern recognition, CVPR 2017*.

Clevert, D. A., Unterthiner, T., & Hochreiter, S. (2016). Fast and accurate deep network learning by exponential linear units (ELUs). In *4th international conference on learning representations, ICLR 2016 - Conference track proceedings*.

Craik, A., He, Y., & Contreras-Vidal, J. L. (2019). Deep learning for electroencephalogram (EEG) classification tasks: a review. *Journal of Neural Engineering*, *16*, Article 031001, Available at: https://iopscience.iop.org/article/10.1088/1741-2552/ab0ab5.

Crawford, J. D., Henriques, D. Y. P., & Medendorp, W. P. (2011). Three-dimensional transformations for goal-directed action. *Annual Review of Neuroscience*, *34*, 309–331, Available at: http://www.annualreviews.org/doi/10.1146/annurev-neuro-061010-113749.

Cui, H., & Andersen, R. A. (2011). Different representations of potential and selected motor plans by Distinct Parietal Areas. *The Journal of Neuroscience*, *31*, 18130–18136, Available at: https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.6247-10.2011.

De Vitis, M., Breveglieri, R., Hadjidimitrakis, K., Vanduffel, W., Galletti, C., & Fattori, P. (2019). The neglected medial part of macaque area PE: segregated processing of reach depth and direction. *Brain Structure and Function*, *224*, 2537–2557, Available at: http://link.springer.com/10.1007/s00429-019-01923-8.

Fattori, P., Kutz, D. F., Breveglieri, R., Marzocchi, N., & Galletti, C. (2005). Spatial tuning of reaching activity in the medial parieto-occipital cortex (area V6A) of macaque monkey. *European Journal of Neuroscience*, *22*, 956–972, Available at: http://www.ncbi.nlm.nih.gov/pubmed/16115219.

Fattori, P., Raos, V., Breveglieri, R., Bosco, A., Marzocchi, N., & Galletti, C. (2010). The dorsomedial pathway is not just for reaching: grasping neurons in the medial parieto-occipital cortex of the macaque monkey. *The Journal of Neuroscience*, *30*, 342–349, Available at: http://www.jneurosci.org/cgi/doi/10.1523/JNEUROSCI.3800-09.2010.

Filippini, M., Breveglieri, R., Hadjidimitrakis, K., Bosco, A., & Fattori, P. (2018). Prediction of reach goals in depth and direction from the parietal cortex. *Cell Reports*, *23*, 725–732, Available at: http://www.ncbi.nlm.nih.gov/pubmed/29669279.

Galletti, C., Battaglini, P. P., & Fattori, P. (1995). Eye position influence on the parieto-occipital area PO (V6) of the macaque monkey. *European Journal of Neuroscience*, *7*, 2486–2501, Available at: http://doi.wiley.com/10.1111/j.1460-9568.1995.tb01047.x.

Galletti, C., Breveglieri, R., Lappe, M., Bosco, A., Ciavarro, M., & Fattori, P. (2010). Covert shift of attention modulates the ongoing neural activity in a reaching area of the macaque dorsomedial visual stream. *PLoS One*, *5*, Article e15078, Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2993960&tool=pmcentrez&rendertype=abstract.

Galletti, C., & Fattori, P. (2018). The dorsal visual stream revisited: Stable circuits or dynamic pathways? *Cortex*, *98*, 203–217, Available at: http://www.ncbi.nlm.nih.gov/pubmed/28196647.

Galletti, C., Fattori, P., Kutz, D. F., & Gamberini, M. (1999). Brain location and visual topography of cortical area V6A in the macaque monkey. *European Journal of Neuroscience*, *11*, 575–582, Available at: http://www.ncbi.nlm.nih.gov/pubmed/10051757.

Galletti, C., Gamberini, M., Kutz, D. F., Baldinotti, I., & Fattori, P. (2005). The relationship between V6 and PO in macaque extrastriate cortex. *European Journal of Neuroscience*.

Galletti, C., Kutz, D. F., Gamberini, M., Breveglieri, R., & Fattori, P. (2003). Role of the medial parieto-occipital cortex in the control of reaching and grasping movements. In *Experimental brain research* (pp. 158–170). Available at: http://www.ncbi.nlm.nih.gov/pubmed/14517595.

Gamberini, M., Dal Bò, G., Breveglieri, R., Briganti, S., Passarelli, L., Fattori, P., et al. (2018). Sensory properties of the caudal aspect of the macaque's superior parietal lobule. *Brain Structure and Function*, *223*, 1863–1879, Available at: http://link.springer.com/10.1007/s00429-017-1593-x.

Gamberini, M., Passarelli, L., Fattori, P., & Galletti, C. (2020). Structural connectivity and functional properties of the macaque superior parietal lobule. *Brain Structure and Function*, *225*, 1349–1367, Available at: http://link.springer.com/10.1007/s00429-019-01976-9.

Glaser, J. I., Benjamin, A. S., Chowdhury, R. H., Perich, M. G., Miller, L. E., & Kording, K. P. (2020). Machine learning for neural decoding. *eNeuro*, *7*, ENEURO.0506-19.2020 Available at: https://www.eneuro.org/lookup/doi/10.1523/ENEURO.0506-19.2020.

Goodale, M. A., & Milner, A. D. (1992). Separate visual pathways for perception and action. *Trends in Neurosciences*, *15*, 20–25, Available at: http://www.ncbi.nlm.nih.gov/pubmed/1374953.

Hadjidimitrakis, K., Bertozzi, F., Breveglieri, R., Bosco, A., Galletti, C., & Fattori, P. (2014). Common neural substrate for processing depth and direction signals for reaching in the monkey medial posterior parietal cortex. *Cerebral Cortex*, *24*, 1645–1657.

Hadjidimitrakis, K., Breveglieri, R., Bosco, A., & Fattori, P. (2012). Three-dimensional eye position signals shape both peripersonal space and arm movement activity in the medial posterior parietal cortex. *Frontiers in Integrative Neuroscience*, *6*, 37, Available at: http://www.ncbi.nlm.nih.gov/pubmed/22754511.

Hadjidimitrakis, K., Breveglieri, R., Placenti, G., Bosco, A., Sabatini, S. P., & Fattori, P. (2011). Fix your eyes in the space you could reach: neurons in the macaque medial parietal cortex prefer gaze positions in peripersonal space. *PLoS One*, *6*, Article e23335, Gribble PL, ed. Available at: http://dx.plos.org/10.1371/journal.pone.0023335.

Hadjidimitrakis, K., Dal Bo', G., Breveglieri, R., Galletti, C., & Fattori, P. (2015). Overlapping representations for reach depth and direction in caudal superior parietal lobule of macaques. *Journal of Neurophysiology*, *114*, 2340–2352, Available at: http://www.ncbi.nlm.nih.gov/pubmed/26269557.

Hauschild, M., Mulliken, G. H., Fineman, I., Loeb, G. E., & Andersen, R. A. (2012). Cognitive signals for brain-machine interfaces in posterior parietal cortex include continuous 3D trajectory commands. *Proceedings of the National Academy of Sciences of the United States of America*, *109*, 17075–17080.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *32nd int conf mach learn ICML 2015*. Available at: http://arxiv.org/abs/1502.03167.

Karnath, H.-O., & Perenin, M.-T. (2005). Cortical control of visually guided reaching: evidence from patients with optic ataxia. *Cerebral Cortex*, *15*, 1561–1569, Available at: https://academic.oup.com/cercor/article-lookup/doi/10.1093/cercor/bhi034.

Kay, K. N. (2018). Principles for models of neural information processing. *Neuroimage*, *180*, 101–109, Available at: https://doi.org/10.1016/j.neuroimage.2017.08.016.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. In *3rd int conf learn represent ICLR 2015 - Conf track proc*. Available at: http://arxiv.org/abs/1412.6980.

Kutz, D. F., Fattori, P., Gamberini, M., Breveglieri, R., & Galletti, C. (2003). Early- and late-responding cells to saccadic eye movements in the cortical area V6A of macaque monkey. *Experimental Brain Research*, *149*, 83–95, Available at: http://www.ncbi.nlm.nih.gov/pubmed/12592506.

Kutz, D. F., Marzocchi, N., Fattori, P., Cavalcanti, S., & Galletti, C. (2005). Real-time supervisor system based on trinary logic to control experiments with behaving animals and humans. *Journal of Neurophysiology*, *93*, 3674–3686, Available at: http://www.ncbi.nlm.nih.gov/pubmed/15703220.

Land, M. F. (2014). Do we have an internal model of the outside world? *Philosophical Transactions of the Royal Society B*, *369*, Article 20130045, Available at: https://royalsocietypublishing.org/doi/10.1098/rstb.2013.0045.

LeCun, Yann, Bengio, Yoshua, & Hinton, G. (2015). Deep learning. *Nature*, Y. LeCun, Y. Bengio and G. Hinton.

Lehmann, S. J., & Scherberger, H. (2013). Reach and gaze representations in macaque parietal and premotor grasp areas. *The Journal of Neuroscience*, *33*, 7038–7049, Available at: http://www.jneurosci.org/content/33/16/7038.abstract.

Luppino, G., Ben Hamed, S., Gamberini, M., Matelli, M., & Galletti, C. (2005). Occipital (V6) and parietal (V6A) areas in the anterior wall of the parieto-occipital sulcus of the macaque: A cytoarchitectonic study. *European Journal of Neuroscience*, *21*, 3056–3076, Available at: http://www.ncbi.nlm.nih.gov/pubmed/15978016.

Matelli, M., Govoni, P., Galletti, C., Kutz, D. F., & Luppino, G. (1998). Superior area 6 afferents from the superior parietal lobule in the macaque monkey. *The Journal of Comparative Neurology*, *402*, 327–352, Available at: http://www.ncbi.nlm.nih.gov/pubmed/9853903.

Medendorp, W. P., & Heed, T. (2019). State estimation in posterior parietal cortex: Distinct poles of environmental and bodily states. *Progress in Neurobiology*, *183*, Article 101691, Available at: https://linkinghub.elsevier.com/retrieve/pii/S0301008219301145.

Michaels, J. A., Schaffelhofer, S., Agudelo-Toro, A., & Scherberger, H. (2020). A goal-driven modular neural network predicts parietofrontal neural dynamics during grasping. *Proceedings of the National Academy of Sciences*, *117*, 32124–32135, Available at: http://www.pnas.org/lookup/doi/10.1073/pnas.2005087117.

Mulliken, G. H., Musallam, S., & Andersen, R. A. (2008a). Forward estimation of movement state in posterior parietal cortex. *Proceedings of the National Academy of Sciences*, *105*, 8170–8177, Available at: http://www.ncbi.nlm.nih.gov/pubmed/18499800.

Mulliken, G. H., Musallam, S., & Andersen, R. A. (2008b). Decoding trajectories from posterior parietal cortex ensembles. *The Journal of Neuroscience*, *28*, 12913–12926, Available at: http://www.jneurosci.org/cgi/doi/10.1523/JNEUROSCI.1463-08.2008%5Cnpapers3://publication/doi/10.1523/JNEUROSCI.1463-08.2008.

Musallam, S., Corneil, B. D., Greger, B., Scherberger, H., & Andersen, R. (2004). Cognitive control signals for neural prosthetics. *Science*, *305*, 258–262, Available at: http://www.ncbi.nlm.nih.gov/pubmed/15247483.

Pandya, D. N., & Seltzer, B. (1982). Intrinsic connections and architectonics of posterior parietal cortex in the rhesus monkey. *The Journal of Comparative Neurology*, *204*, 196–210, Available at: http://www.ncbi.nlm.nih.gov/pubmed/6276450.

Pisella, L., Sergio, L., Blangero, A., Torchin, H., Vighetto, A., & Rossetti, Y. (2010). Erratum to optic ataxia and the function of the dorsal stream: Contributions to perception and action. *Neuropsychologia*; (2009). *Neuropsychologia*, *47*(14), 3033-3044.

Richards, B. A., et al. (2019). A deep learning framework for neuroscience. *Nature Neuroscience*, *22*, 1761–1770, Available at: http://www.nature.com/articles/s41593-019-0520-2.

Rizzolatti, G., & Matelli, M. (2003). Two different streams form the dorsal visual system: Anatomy and functions. In *Experimental brain research* (pp. 146–157).

Roy, A. C., Paulignan, Y., Farnè, A., Jouffrais, C., & Boussaoud, D. (2000). Hand kinematics during reaching and grasping in the macaque monkey. *Behavioural Brain Research*, *117*, 75–82, Available at: https://linkinghub.elsevier.com/retrieve/pii/S0166432800002849.

Sanchez, J. C., Erdogmus, D., Nicolelis, M. A. L., Wessberg, J., & Principe, J. C. (2005). Interpreting spatial and temporal neural activity through a recurrent neural network brain-machine interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*.

Seelke, A. M. H., Padberg, J. J., Disbrow, E., Purnell, S. M., Recanzone, G., & Krubitzer, L. (2012). Topographic maps within brodmann's area 5 of macaque monkeys. *Cerebral Cortex*.

Shah, S., Haghi, B., Kellis, S., Bashford, L., Kramer, D., Lee, B., et al. (2019). Decoding kinematics from human parietal cortex using neural networks. In *International IEEE/EMBS conference on neural engineering*. NER.

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*, Available at: https://www.frontiersin.org/article/10.3389/fnins.2020.568104/full.

Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd international conference on learning representations, ICLR 2014 - Workshop Track Proceedings*.

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *3rd international conference on learning representations, ICLR 2015 - Conference track proceedings*.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In *Adv neural inf process syst*. Available at: http://arxiv.org/abs/1206.2944.

Snyder, L. H., Batista, a. P., & Andersen, R. a. (1997). Coding of intention in the posterior parietal cortex. *Nature*, *386*, 167–170.

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

Sussillo, D., Nuyujukian, P., Fan, J. M., Kao, J. C., Stavisky, S. D., Ryu, S., et al. (2012). A recurrent neural network for closed-loop intracortical brain–machine interface decoders. *Journal of Neural Engineering*, *9*, Article 026027, Available at: https://iopscience.iop.org/article/10.1088/1741-2560/9/2/026027.

Tjoa, E., & Guan, C. (2020). A survey on explainable artificial intelligence (XAI): Toward medical XAI. *IEEE Transactions on Neural Networks and Learning Systems*, 1–21, Available at: https://ieeexplore.ieee.org/document/9233366/.

Todorov, E. (2004). Optimality principles in sensorimotor control. *Nature Neuroscience*, *7*, 907–915, Available at: http://www.nature.com/articles/nn1309.

Tramper, J. J., & Gielen, C. C. A. M. (2011). Visuomotor coordination is different for different directions in three-dimensional space. *The Journal of Neuroscience*, *31*, 7857–7866, Available at: https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.0486-11.2011.

Tseng, P.-H., Urpi, N. A., Lebedev, M., & Nicolelis, M. (2019). Decoding movements from cortical ensemble activity using a long short-term memory recurrent network. *Neural Computation*, *31*, 1085–1113, Available at: https://direct.mit.edu/neco/article/31/6/1085-1113/8478.

Ungerleider, L. G., & Mishkin, M. (1982). Two cortical visual systems. *Analysis of Visual Behavior*, *54*, 9–586.

Yamins, D. L. K., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., & DiCarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, *111*, 8619–8624, Available at: http://www.pnas.org/cgi/doi/10.1073/pnas.1403112111.