



OPEN ACCESS

EDITED BY

Jianjun Meng,
Shanghai Jiao Tong University, China

REVIEWED BY

Huan Gao,
Carnegie Mellon University, United States
Wolf-Julian Neumann,
Charité – Universitätsmedizin Berlin, Germany
Timon Merk,
Charité – Universitätsmedizin Berlin, Germany,
in collaboration with reviewer W-JN

*CORRESPONDENCE

Huiling Tan
✉ huiling.tan@ndcn.ox.ac.uk

RECEIVED 29 November 2022

ACCEPTED 04 May 2023

PUBLISHED 24 May 2023

CITATION

Martineau T, He S, Vaidyanathan R and Tan H
(2023) Hyper-parameter tuning and feature
extraction for asynchronous action detection
from sub-thalamic nucleus local field
potentials.
Front. Hum. Neurosci. 17:1111590.
doi: 10.3389/fnhum.2023.1111590

COPYRIGHT

© 2023 Martineau, He, Vaidyanathan and Tan.
This is an open-access article distributed under
the terms of the [Creative Commons Attribution
License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or
reproduction in other forums is permitted,
provided the original author(s) and the
copyright owner(s) are credited and that the
original publication in this journal is cited, in
accordance with accepted academic practice.
No use, distribution or reproduction is
permitted which does not comply with
these terms.

Hyper-parameter tuning and feature extraction for asynchronous action detection from sub-thalamic nucleus local field potentials

Thomas Martineau¹, Shenghong He^{2,3}, Ravi Vaidyanathan^{1,4} and Huiling Tan^{2,3*}

¹Biomechatronics Group, Department of Mechanical Engineering, Imperial College London, London, United Kingdom, ²Medical Research Council Brain Network Dynamics Unit, University of Oxford, Oxford, United Kingdom, ³Nuffield Department of Clinical Neurosciences, University of Oxford, Oxford, United Kingdom, ⁴UK Dementia Research Institute–Care Research and Technology, Imperial College London, London, United Kingdom

Introduction: Decoding brain states from subcortical local field potentials (LFPs) indicative of activities such as voluntary movement, tremor, or sleep stages, holds significant potential in treating neurodegenerative disorders and offers new paradigms in brain-computer interface (BCI). Identified states can serve as control signals in coupled human-machine systems, e.g., to regulate deep brain stimulation (DBS) therapy or control prosthetic limbs. However, the behavior, performance, and efficiency of LFP decoders depend on an array of design and calibration settings encapsulated into a single set of hyper-parameters. Although methods exist to tune hyper-parameters automatically, decoders are typically found through exhaustive trial-and-error, manual search, and intuitive experience.

Methods: This study introduces a Bayesian optimization (BO) approach to hyper-parameter tuning, applicable through feature extraction, channel selection, classification, and stage transition stages of the entire decoding pipeline. The optimization method is compared with five real-time feature extraction methods paired with four classifiers to decode voluntary movement asynchronously based on LFPs recorded with DBS electrodes implanted in the subthalamic nucleus of Parkinson's disease patients.

Results: Detection performance, measured as the geometric mean between classifier specificity and sensitivity, is automatically optimized. BO demonstrates improved decoding performance from initial parameter setting across all methods. The best decoders achieve a maximum performance of 0.74 ± 0.06 (mean \pm SD across all participants) sensitivity-specificity geometric mean. In addition, parameter relevance is determined using the BO surrogate models.

Discussion: Hyper-parameters tend to be sub-optimally fixed across different users rather than individually adjusted or even specifically set for a decoding task. The relevance of each parameter to the optimization problem and comparisons between algorithms can also be difficult to track with the evolution of the decoding problem. We believe that the proposed decoding pipeline and

BO approach is a promising solution to such challenges surrounding hyper-parameter tuning and that the study's findings can inform future design iterations of neural decoders for adaptive DBS and BCI.

KEYWORDS

Bayesian optimization (BO), deep brain stimulation (DBS), time-frequency analysis, local field potentials (LFPs), brain-computer interface (BCI)

1. Introduction

The successful adoption of deep brain stimulation (DBS) in the treatment of neurological disorders has prompted interest in subcortical brain regions as alternative invasive recording sites for brain-computer interface (BCI). Pathways through the basal ganglia and thalamus are heavily involved in the preparation and regularization of movement. These motor-control mechanisms are, in part, described by an asymmetry between anti-kinetic beta (12–32 Hz) and pro-kinetic gamma (32–90 Hz) waves measured in local field potentials (LFPs) (Cassidy et al., 2002; Brown, 2003; Anzak et al., 2012; Tan et al., 2013, 2016). These waves are useful biomarkers for neural decoding. Stimulation triggered during hyper-synchronous beta rhythms of LFPs in the subthalamic nucleus (STN) can help alleviate Parkinsonian rigidity and bradykinesia symptoms (Little et al., 2013). Since, other pathological and physiological states have been identified from subcortical LFPs, including tremor (Bakstein et al., 2012; Camara et al., 2015; Hirschmann et al., 2017; Tan et al., 2019; Yao et al., 2020; Zhu et al., 2020; He et al., 2021), dopamine medication effects (Sand et al., 2021), pain (Zhang et al., 2013; Luo et al., 2018), sleep stages (Chen et al., 2019), finger pressing (Mamun et al., 2015; Golshan et al., 2018, 2020; Niketeghad et al., 2018), reaching actions (Golshan et al., 2018, 2020), left- or right-hand selection (Mamun et al., 2015), speech (Golshan et al., 2018, 2020), behavioral tasks (Zaker et al., 2016; Golshan et al., 2018, 2020), isometric force (Shah et al., 2016, 2018; Tan et al., 2016), body orientation (Mace et al., 2014), and gait stages (Tan et al., 2018). Deep brain LFPs decoding supports adaptive DBS treatment (He et al., 2021) and could foster the development of new BCI applications, especially aimed at DBS patients. These can, for instance, aim at neurofeedback therapy through gamification (He et al., 2019) or drive wearable devices to assist patients with impaired mobility (Benabid et al., 2019).

This article investigates a hyper-parameter automated tuning approach for STN-LFPs decoders. A hyper-parameter is a parameter space that cannot be directly found through a fitting algorithm (the most common being regularization constants). For simplicity, these will be referred to simply as a parameter from thenceforward. For decoding based real-time BCI or adaptive DBS, the pipeline tends to be separated into several stages,

such as extracting relevant features, selecting channels, classifying, and stage transition. The impact of individual parameters in different stages of the processing pipeline has typically been gauged independently from one another (Shah et al., 2018; Yao et al., 2020). As algorithms evolve toward practical applications, increased complexity almost always results in larger parameter spaces (Mace et al., 2014). Manual tuning can become tedious when assuming no prior knowledge of the relevance of each parameter. Therefore, parameters are often sub-optimally selected based on preliminary assessments and maintained constant across all testing participants. When assuming limited computational resources, grid searching becomes impractical in large search hyper-volumes (Bergstra and Bengio, 2012). Gradient-based optimization constrains the problem to the differentiable functions and is not well suited for small-size datasets (Martineau et al., 2020). Other methods, such as genetic algorithms, have also been proposed. However, this more complex optimization approach necessitates expert knowledge in selecting the necessary genetic operators and correctly interpreting the optimization process (Mace et al., 2014), which may discourage its adoption.

In this study, Bayesian optimization (BO) was initially chosen for its ease of implementation. It employs predictive Bayesian modeling to iteratively sample the parameter space of a “black-box” objective function (Shahriari et al., 2016). Unlike random search (RD), BO uses past samples to make the best-informed sampling decision and guarantees better outcomes within a limited sampling budget (Shahriari et al., 2016). In BCI research, BO has been principally dedicated to the tuning parameters related to the machine learning stages (Ahmadi et al., 2019; Nandy et al., 2019). However, Bashashati et al. (2016) have shown that BO can be used to fine-tune feature extraction and spatial filtering parameters. The foreseen advantages of BO are threefold. First, it could completely automate and unify the tuning process. Secondly, it could improve decoding performance over what is achieved with a set of known parameters (drawn from what is typically used in other past studies) fixed for different users. Thirdly, as a model-driven approach, BO intrinsically builds a representation of the tuning process mapping parameters to the optimization objective. Parameter importance can thus be determined, and the BO model can also be iteratively updated for sequential tuning (Grado et al., 2018). As pointed out by Grado et al. (2018), a BO algorithm would not be deployed alongside the decoder algorithm on a device firmware (e.g., a DBS micro-controller unit). Instead, it would operate remotely and intermittently suggest parameter changes to said device based on transmitted LFP data, thus providing a secondary outer and long-time-scale adjustment loop tuning.

Moreover, asynchronous decoding detects a state change based on temporal events present within a small analysis window extracted from a continuous data stream (Nicolas-Alonso and Gomez-Gil, 2012). Although it is required for many practical

Abbreviations: PD, Parkinson's disease; LFPs, local field potentials; DBS, deep brain stimulation; BCI, brain-computer interface; STN, subthalamic nucleus; BO, Bayesian optimization; RD, random search; IIR, infinite impulse response; STFT, short-time Fourier transform; CWT, continuous wavelet transform; RWT, recursive wavelet transform; DWT, discrete wavelet transform; ARMA, auto-regressive moving average; HJ, Hjorth parameters; MI, mutual information; LDA, linear-discriminant analysis; NB, Naïve Bayesian; LR, logistic regression; SVM, support vector machine; PCA, principal component analysis; ROC, receiver operating characteristic; TPR, true-positive rate; FPR, false-negative rate; ANOVA, analysis of variance.

applications, such as adaptive DBS, asynchronous decoding from STN-LFPs remains challenging, especially for the detection of sustained actions in time (Mace et al., 2014; Zaker et al., 2016; Niketeghad et al., 2018; Martineau et al., 2020). Reasons include poor signal-to-noise ratio and non-stationary behaviors of the feature signals (Niketeghad et al., 2018; Yao et al., 2020), high focality of the signal source in space across the electrode contacts (Anzak et al., 2012; Tan et al., 2013; Zhang et al., 2018), and significant inter-subject and inter-trial variances (Tan et al., 2013; Niketeghad et al., 2018). Therefore, several parameters must be set for asynchronous decoding, including feature extraction window length, selected recording channels, or the number of past feature samples considered for classification. An action [isometrical power gripping actions, as defined in Martineau et al. (2020)] detection task was chosen in this study to assess the feasibility of STN-LFP asynchronous decoding further.

Finally, although the offline correlation of movement with LFPs can be significant, in most previous studies, those correlations are based on average across multiple trials. As emphasized in Mace et al. (2014), Niketeghad et al. (2018), and Shah et al. (2018), the restrictions imposed by a real-time system (those being limited computational resources and memory allocation, causal operation on the data) are intrinsically linked to those of asynchronous decoding and often omitted, notably in the feature extraction stage. Estimated performances made offline for investigative purposes or parameter tuning need to represent online performance. The last requirement was thus to ensure that the decoder could be deployed in real-time, i.e., be refreshed online using streaming data. Many time-frequency decomposition techniques have been proposed to extract information about neural oscillations (Nicolas-Alonso and Gomez-Gil, 2012). Each technique relies on different interpretations of the signal (e.g., signal partitioning in both the time and frequency domain) and hence on different parameterizations. The influence of these parameters could be assessed here using BO. In total, five different real-time extraction methods were tested and compared. Causality in the extraction stage and throughout the pipeline was maintained, i.e., every operation used the current input state or past states stored in memory.

2. Materials and methods

The overall decoder architecture is shown in Figure 1. The STN-LFPs signals were first recorded and pre-processed. Feature extraction was performed, followed by channel selection, feature preparation, and state estimation. Figure 2 presents the BO-based tuning process, and it repeatedly fitted and evaluated the decoder pipeline to select optimal parameters. The decoders and the BO procedure were implemented in python using the SciPy (Virtanen et al., 2019), NumPy (Harris et al., 2020), Scikit-learn (Pedregosa et al., 2011), Imbalanced-learn (Lemaître et al., 2017), and Scikit-optimization (Head et al., 2021) libraries.

2.1. DBS recording dataset

Recordings collected from six test participants, previously reported in Tan et al. (2016) and Shah et al. (2018), were used

in this study. Three participants were tested using their left and right hands, adding up to nine individuals' datasets (Table 1). For practicality, recording across different hemispheres will be assumed to be independent, granted that factors, such as electrode type, physiology or medication may cause some dependencies. The participants were patients with Parkinson's disease undergoing DBS surgery targeting the STN. Each gave informed consent to participate in the study, which was approved by the local ethics committee board. The participants were asked to apply a given amount of grip force for a given period following the presentation of a visual or auditory queue. Precise details on the data recording procedure can be found in Tan et al. (2013, 2016). Table 1 presents the number of trials and channels recorded. On average, a recording was 402 ± 105 s long and contained 40 ± 9 trials. Each grip was held for 4.1 ± 1.4 s. The ratio of grip against rest was $31 \pm 7\%$, and the average holding force was $47 \pm 27\%$ of the maximum voluntary contraction. In each dataset, multiple channels of LFPs were recorded from lateral and contra-lateral electrodes (Table 1). The LFP signals were pre-processed following the same method outlined in Martineau et al. (2020). In summary, the prep-processing steps include detrending the recordings using a low-pass filter, removing main artifacts using notch filters, scaling the signals based on their standard deviation and resampling at the desired frequency. The dataset obtained was the following: an LFPs array x with time and electrode channel dimensions (for both hemispheres) such that $x \in \mathbb{R}^{N_k \times N_{ch}}$, where N_k is the total number of samples and N_{ch} , the total number of channels. x was re-sampled at a rate of 512 Hz, which was defined as the decoder input sampling frequency f_i . The decoder output \tilde{y} matched the detection target y at a rate of 16 Hz, defined as the decoder output sampling frequency f_o . Decoders typically operate in the range of 10–20 Hz (Shah et al., 2018; He et al., 2021). A base 2 number was chosen to facilitate downsampling, especially for implementing the discrete wavelet transform-based feature extractor. As shown in Figure 3, a sampling step-down $R = f_i/f_o$ was performed such that $\tilde{y}[k], y[k] \in \{0, 1\} \forall k \in \{0, 1, \dots, \frac{N_k}{R} - 1\}$ with 0 marking a rest state and 1 an action state. A target lead of 125 ms was also introduced (Benabid et al., 2019). It is common practice to predict the target a few decoder update steps (in this case, two update steps) in advance to compensate for potential latencies introduced by processing, communication or even stimulator ramp-up (Tarvainen et al., 2004; Gruenwald et al., 2017; Benabid et al., 2019). Finally, 10% of the length of each recording was set aside for testing after tuning.

2.2. Feature extraction

Five different frequency-domain and real-time extraction methods, including a bank of infinite impulse response (IIR) band-pass filters, a short-time Fourier transform (STFT), a recursive wavelet transform (RWT), a parameterized discrete wavelet transform (DWT), an adaptive auto-regressive moving average (ARMA) filter, based method, were implemented.

Filter banks have been proposed as a computationally efficient solution to extract standard electrophysiological sub-bands (Zhu et al., 2020). The method proposed by Gruenwald et al. (2017) was followed to construct a bank of fourth-order IIR Butterworth filters.

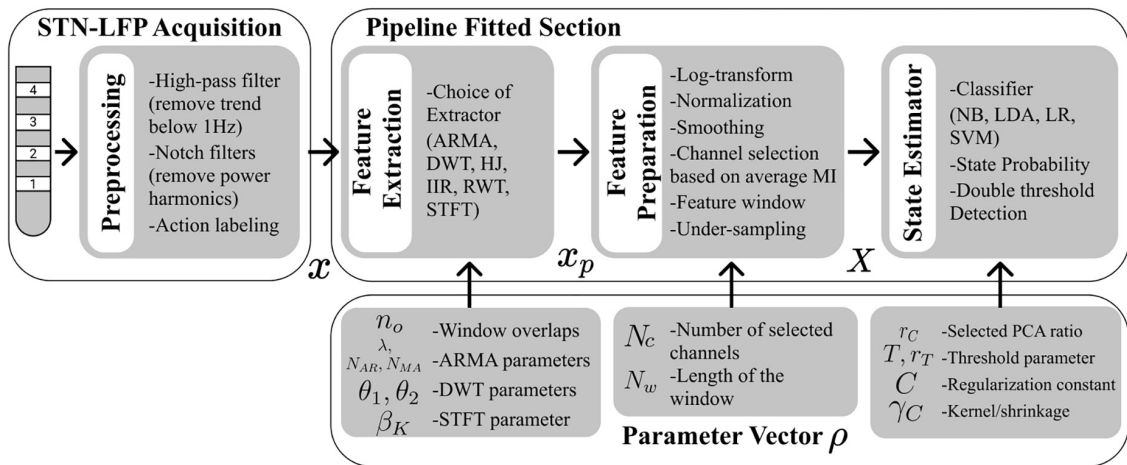


FIGURE 1 Flow diagram summarizing the sequence of processes for decoding and the different parameters assigned to them. Pre-processing was done prior to decoder pipeline fitting and tuning. A decoder pipeline had three main stages: a feature extraction (using either the ARMA, DWT, HJ, IIR, RWT, STFT method), a feature preparation stage including channel selection based on mutual information (MI) ranking, and a classifier (using either the NB, LR, LDA, SVM-PCA) stage including thresholding. Parameters in each of these three stages were tuned using BO.

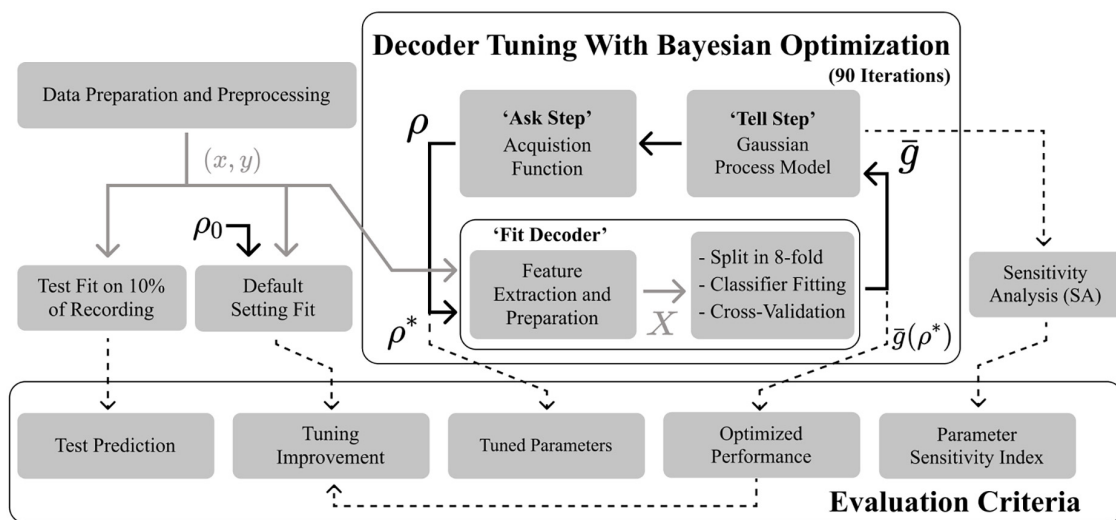


FIGURE 2 Flow diagram summarizing the Bayesian optimization and decoder fitting procedures, with associated evaluation criteria. Data is prepared in a pair (x, y) of neural signal and detection target and passed on to the optimizer. Ten percent of the recording is set aside for testing. The optimizer iteratively fits the pipeline presented in **Figure 1** to yield a score \bar{g} . The first parameter vector ρ evaluated is the default vector ρ_0 . After what the optimizer fits a surrogate model mapping ρ to \bar{g} . An acquisition function uses the surrogate model to predict the next ρ . After a fixed number of iterations, the tuned parameters, optimal score, and improvement on default performance are recorded. Parameter importance is obtained through a sensitivity analysis of the surrogate model.

The frequency band were defined as θ (1–8 Hz), α (8–12 Hz), β (12–32 Hz), γ_1 (32–50 Hz), γ_2 (50–100 Hz), γ_3 (100–256 Hz), leading to a total of N_b frequency bands. The power of each sub-band $x_p \in \mathbb{R}^{\frac{N_k}{R} \times N_{ch} \times N_b}$ was estimated from the signal by measuring the variance of the sub-signals on a window length L and strides R . L was defined as a multiple of R such that:

$$L = R(n_o + 1) \tag{1}$$

where n_o is the window overlap number. A maximum L of 250 ms was allowed as in **Flint et al. (2012)**, such that $n_o \in \llbracket 0; 3 \rrbracket$.

Secondly, the STFT is a reference time-frequency estimation method used in previous STN-LFP decoding studies (**Bakstein et al., 2012; Hirschmann et al., 2017; Shah et al., 2018**). The same epoching parameterization was used and the Kaiser window was chosen to parameterize the window shape through a single parameter $\beta_K \in [0; 15]$. The STFT yielded a complete time-frequency transformation $\chi \in \mathbb{C}^{\frac{N_k}{R} \times N_{ch} \times (\frac{L}{R} + 1)}$ of x . The same N_b sub-bands were extracted by averaging the correct frequency bins following the power operator $\chi[k, l, n] \chi^*[k, l, n]$ to obtain x_p . A drawback of the STFT is that the resolution of the transform is directly dependent on L . With a small number of or no

TABLE 1 Participant details, including DBS electrode channels available for recording and laterality.

Participant ID	PD01R	PD02L	PD02R	PD03L	PD04L	PD05L	PD05R	PD06L	PD06R	Mean \pm SD
Contralateral channels	4	4	3	4	4	4	4	4	2	4
Ipsilateral channels	4	3	4	0	4	4	4	2	4	3
Number of trials	26	22	24	21	20	34	34	49	40	30 \pm 9
Recording duration (s)	383	337	335	240	257	512	453	553	530	402 \pm 105
Grip duration (mean \pm SD, s)	2.95 \pm 0.64	5.37 \pm 0.19	5.28 \pm 0.67	5.95 \pm 1.54	3.25 \pm 0.16	5.10 \pm 1.33	3.61 \pm 0.91	3.30 \pm 1.72	3.79 \pm 1.34	4.1 \pm 1.4
Ratio of grip to rest (%)	20	35	33	49	25	34	27	29	30	31 \pm 7
Grip level ratio (mean \pm SD, %)	33 \pm 19	0.82 \pm 12	68 \pm 15	69 \pm 17	55 \pm 28	37 \pm 26	29 \pm 21	39 \pm 24	44 \pm 28	47 \pm 27

PD denominates Parkinson's disease, L indicates that participant used their left hand to execute an isometric grip action. R indicates their right hand. A trial consists of a single isometric grip action including onset, holding, and relaxation stages. Grip duration is the time from the beginning of the action to its end averaged across trials. Grip ratio indicates the class balance of action vs. rest time points for every dataset. Grip level ratio is the trial peak force relative to the maximum force exerted in the recording averaged across trials.

overlaps, narrower lower bands are thus not properly represented (in particular α) and not included as features.

This problem is typically addressed by opting for the continuous wavelet transform (CWT) instead. In addition to better capturing non-stationary signal behaviors, the CWT allows for multi-resolution signal analysis. The Morlet wavelet in particular is predominantly used in the field and thus is also regularly employed in decoder architectures (Bashashati et al., 2016; Tan et al., 2019; Golshan et al., 2020). To reduce the computational cost and latency of the CWT, Ren and Kezunovic (2010) proposed a complex sinusoid wavelet function, ψ , shaped with an asymmetric fifth-order polynomial envelope:

$$\psi(t) = \left(-\frac{8}{9\sqrt{3}}(\pi t)^3 - \frac{8}{27}(\pi t)^4 - \frac{32}{135\sqrt{3}}(\pi t)^5 \right) e^{2\pi\left(\frac{1}{\sqrt{3}}+j\right)t} \quad (2)$$

Ren and Kezunovic (2010) demonstrated that the CWT with ψ can be recursively expressed. This RWT approach was implemented as a complex IIR filter bank. Each filter projected the signal on a wavelet basis. Four wavelets were used per octave between 1 and 256 Hz, or a total of 32 filters, yielding a time-frequency transformation $\chi \in \mathbb{C}^{N_k \times N_{ch} \times 32}$. The power array $x_p \in \mathbb{R}^{\frac{N_k}{R} \times N_{ch} \times N_b}$ was obtained after a power operation and application of epoch averages on χ .

Moreover, the more common alternative to the CWT is the DWT (Bakstein et al., 2012; Mamun et al., 2015; Luo et al., 2018). The DWT is implemented using small and efficient filter bank which samples discrete points in the CWT. The main design difficulty associated with the DWT is the choice of a mother wavelet ψ (Mamun et al., 2015; Luo et al., 2018). Farina et al. (2007) proposed using the parameterized lattice realization of the quadrature mirror filter to shape the mother wavelet to neural signals which in turn helped increase decoding performance. Each node in the DWT decomposition tree was constructed using a three-tap filter. The quadrature mirror filter split a signal x into low x_g and high x_h components through two rotations $\theta_1, \theta_2 \in \left] \frac{\pi}{2}; -\frac{\pi}{2} \right[$ (Vaidyanathan, 1993; Farina et al., 2007). To match f_o , $N_l = \log_2(R)$ levels were required. The decomposition tree recursively filtered x_g at each level l , resulting in a set of wavelet coefficients $S_c = \left\{ x_h^1, x_h^2, \dots, x_h^l, \dots, x_h^{N_l}, x_g^{N_l} \right\}$. The power array $x_p \in \mathbb{R}^{\frac{N_k}{R} \times N_{ch} \times (N_l+1)}$ was assembled by epoching each coefficient of S_c with a matching stride $R_l = \frac{R}{2^l}$, length L_l , and computing the sum of squares.

Last, a parametric modeling and spectral estimation methods have been regularly used in STN-LFP (Foffani et al., 2005; Mamun et al., 2015; Shah et al., 2018) and other neural decoders (Chisci et al., 2010). The recursive-least-square ARMA scheme proposed by Tarvainen et al. (2004) was here investigated. The algorithm extracted the ARMA coefficients $x_c \in \mathbb{R}^{N_k \times N_{ch} \times (N_{AR}+N_{MA})}$ from x . $N_{AR} \in \llbracket 1, 12 \rrbracket$, $N_{MA} \in \llbracket 0, 6 \rrbracket$ are the number of auto-regressive and moving average coefficients, respectively (Tarvainen et al., 2004). x_c was updated using a forgetting factor $\lambda \in]0, 1[$. Epoching served to average and downsample x_c before estimating of the

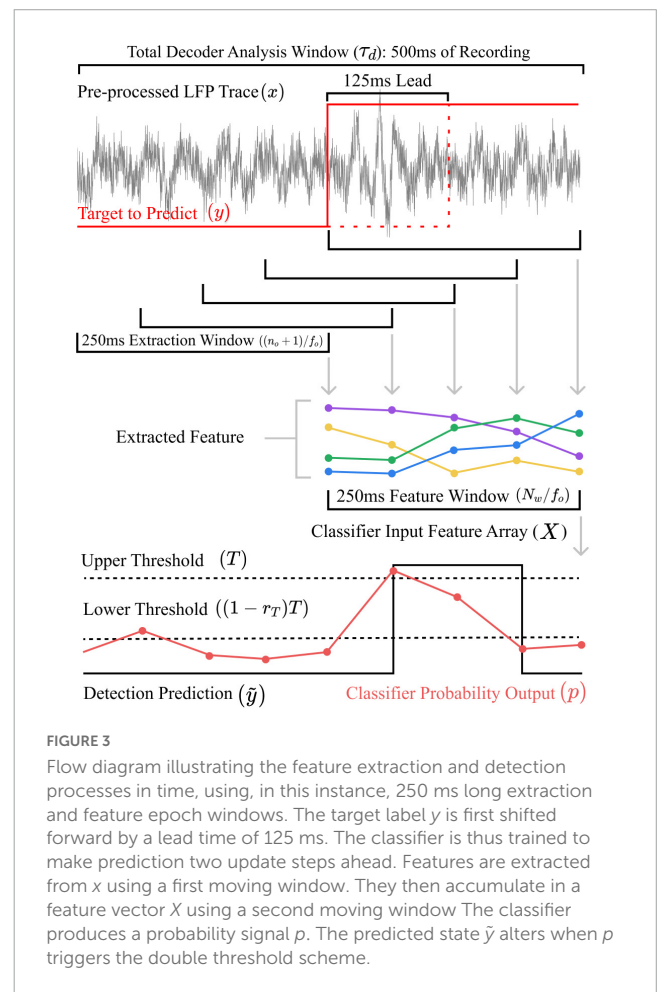


FIGURE 3

Flow diagram illustrating the feature extraction and detection processes in time, using, in this instance, 250 ms long extraction and feature epoch windows. The target label y is first shifted forward by a lead time of 125 ms. The classifier is thus trained to make prediction two update steps ahead. Features are extracted from x using a first moving window. They then accumulate in a feature vector X using a second moving window. The classifier produces a probability signal p . The predicted state \hat{y} alters when p triggers the double threshold scheme.

power spectrum (Tarvainen et al., 2004). The spectrum was partitioned into N_b bands to obtain x_p .

In addition, the usage of Hjorth (HJ) parameters (activity, mobility, and complexity) (Shah et al., 2018; Zhu et al., 2020; He et al., 2021) was investigated. The signal first, x_d , and second derivatives were estimated using discrete differences. The parameters, h_1 , h_2 , and h_3 , were then computed by taking the following the signal variance on epochs of length L and stride R :

$$h_1(x) = \text{Var}_{\text{LR}}(x) \quad (3)$$

$$h_2(x) = \sqrt{\frac{\text{Var}_{\text{LR}}(x)}{\text{Var}_{\text{LR}}(x_d)}} \quad (4)$$

$$h_3(x) = \frac{h_2(x_d)}{h_2(x)} \quad (5)$$

where Var_{LR} is the epoch-variance operator. Hjorth parameters indirectly capture some characteristics of the signal spectrum through the time domain (Hjorth, 1970). Compared to most other time-frequency methods, HJ parameters are inexpensive to compute, making them desirable for real-time applications (Zhu et al., 2020; He et al., 2021).

A computational benchmark was conducted for every feature extraction method by measuring the time taken to extract features from a fix length (100 s) random signal and estimating the time taken to process a signal block of size R . Under real-time constraints, such a block is needed to update the decoder at a rate f_o . The computational time, thus, needs to be inferior to $1/f_o$ (or 62.5 ms). The SciPy implementation (Virtanen et al., 2019) of the STFT algorithm was used partly to serve as a reference, partly because it is well-optimized and uses the fast-Fourier transform algorithm. Given the relatively large size of the extraction windows, a direct filter implementation would have been inefficient. Aside from the STFT, all extractors were implemented using the NumPy library (Harris et al., 2020). In every extractor, every filter was recursively updated at each time step to imitate real-time conditions. When implementing the step update, the well-optimized multi-dimensional array operation capabilities of NumPy (Harris et al., 2020) were carefully considered.

2.3. Channel selection and feature window

From a real-time deployment perspective, it could be costly to carry signal acquisition on all channels, extract all features and only select a sub-set for decoding. Instead, a channel selection approach was chosen whereby channels can be de-activated if non-selected. Channels were selected based on the cumulated mutual information (MI) of their extracted features in relation to the decoded target. Given x_p and y , the average information content \overline{MI} of the l^{th} channel was defined as:

$$\overline{MI}[l] = \frac{1}{N_i} \sum_{i=0}^{N_i-1} MI(x_p[k, l, i], y) \quad (6)$$

where MI is the mutual information operator (Pedregosa et al., 2011; Ross, 2014), N_i is the total number of features for each channel, so either N_b , $N_b + 1$ (if DWT is used) or 3 (if HJ is used),

TABLE 2 Summary of search space S and default vector ρ_0 .

Pipeline	n_o	N_c	N_w	T	r_t
S	[[0; 3]]	[[1, N_{ch}]]	[[0; 15]]	[0, 1]	[0, 1]
ρ_0	3	2	0	0.5	0
Extractor	STFT	DWT	ARMA		
	β_k	θ_1, θ_2	λ	N_{AR}	N_{MA}
S	[[0; 15]]	$\frac{\pi}{2}; -\frac{\pi}{2}$ [[[0, 1[[[1, 12]]	[[0, 6]]
ρ_0	5	(0.50, -0.105)	0.99	6	2
Classifier	LR	LDA	SVM		
	C	γ_c	r_c	C	γ_c
S	[[0, 5]]	[[0, 1]]	[[0, 1].	[[0, 5]]	[[0, 1]]
ρ_0	1	0	0.1	1	0.1

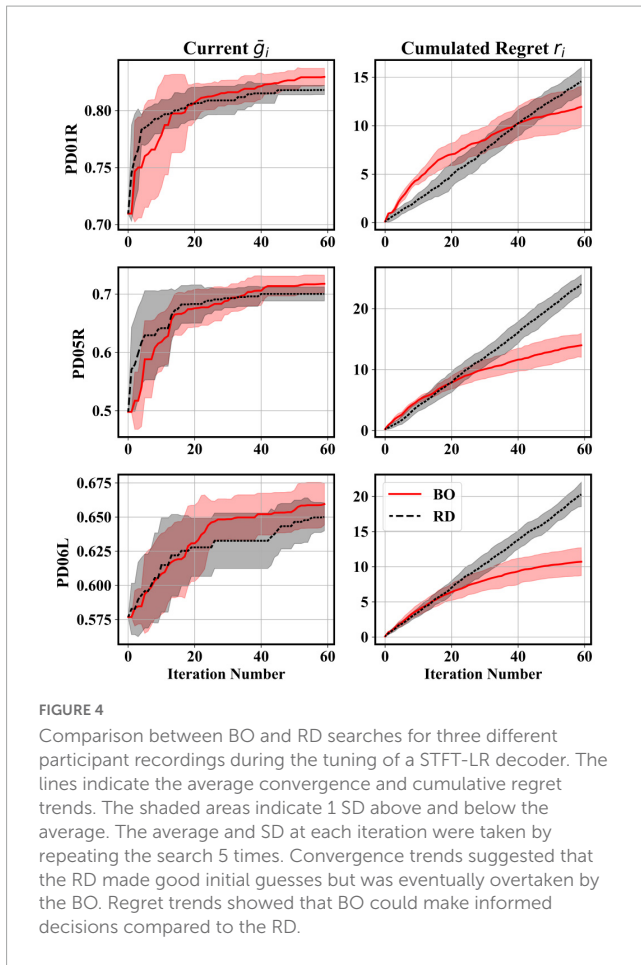
n_o is the number of overlaps between feature extraction windows (with a maximum window of 250 ms). N_c is the number of channels selected amongst N_{ch} after ranking based on MI. N_w controls the length of the feature window (for a maximum of 1 s). T is the upper threshold placed on the output of the classifier. r_t controls the placement of the lower-threshold relative to T . β_k is the shape factor of the STFT window function. θ_1, θ_2 are the lattice parameters for the DWT. λ is the decay rate for the ARMA filter. N_{AR}, N_{MA} control the coefficient settings for the ARMA filter. C are regularization constants for the LR and SVM. γ_c is respectively the shrinkage coefficient for the LDA and a kernel parameter for the SVM. r_c is the PCA ratio for the SVM.

and l is the channel index. \overline{MI} was sorted in descending order. $N_c \in \llbracket 1, N_{ch} \rrbracket$ top ranked channels were selected. Features of the selected channels were further processed by applying a logarithm transform, normalization, and further smoothing using the feature-wise Kalman filter proposed by Gruenwald et al. (2017) (except for the ARMA extractor, where it could be adjusted through λ). The feature matrix $X \in \mathbb{R}^{\frac{N_k}{k} \times N_f}$, with $N_f = N_c N_w N_i$, was assembled by flattening all the channels and bands (or coefficients in the DWT case, or parameters in the HJ case) dimensions and concatenating $N_w \in \llbracket 0; 15 \rrbracket$ past time points to form a moving window of features (or a maximum length of 1 s) (Flint et al., 2012; Shah et al., 2018; Tan et al., 2019; He et al., 2021). The feature and label pair (X, y) were under-sampled according to repeated-edited-nearest-neighbors method to create a more balanced training set before classifier fitting (Tomek, 1976; Lemaître et al., 2017).

Furthermore, the parameters n_o, N_c and N_w controlled the total decoder window τ_d (Figure 3) and the total number of features N_f . These two metrics were used to evaluate the complexity of the resulting decoder. A decoder with a smaller τ_d established more temporally located relationships between neural features and the decoded state. Practically, decoder with longer τ_d required more memory to store past signal epochs for extraction and past feature epochs for state estimation. Likewise, N_f also directly influenced memory storage and, more importantly, the computational cost of updating the state estimator. Tuning N_c and N_w also acted as a form of feature selection, which could help reduce model complexity.

2.4. State estimator and objective function for detection

Four different classifiers retained from He et al. (2021) were tested to estimate y from X : (1) the Gaussian Naïve Bayes (NB) classifier was chosen for its lack of tuning parameters (Mamun et al., 2015). The NB served as a special case to evaluated if decoder



parameters (e.g., n_o , N_c , or N_w) other than fitting regularization constants could be leveraged to increase performance. (2) The linear-discriminant analysis (LDA) classifier has been shown to deliver good decoding performance (Flint et al., 2012; Gruenwald et al., 2019). A shrinkage parameter $\gamma_c \in [0, 1]$ was used to regularize the covariance estimation of X . (3) Logistic regression (LR) has successfully been applied to STN-LFPs decoding (Shah et al., 2016; Tan et al., 2019; He et al., 2021), particularly for sparse model fitting using L_1 -norm regularization. The regularization was set through a parameter $C \in]0, 5]$. Its setting acted in part as a form of feature selector. (4) Thanks to their versatile non-linear kernels, support vector machine (SVM) classifiers have been regularly employed for high-performance decoding (Mamun et al., 2015; Golshan et al., 2016, 2018; Nandy et al., 2019; He et al., 2021). The radial-basis variant was chosen. It required a regulation and kernel parameter C , $\gamma_c \in]0, 5]$ (Nandy et al., 2019). Given the expandable nature of the feature space (of size N_F), the computational demand of the SVM fitting algorithm and the fact that decoder fitting occurred across full-length time-series at each BO iteration, principal component analysis (PCA) dimensionality reduction had to be introduced to compress X by a factor $r_c \in]0, 1]$ (Golshan et al., 2018). The maximum number of selected components was as well capped at 64. Once deployed, the computational cost of operating the NB, LDA, and LR scales with N_f . For the SVM, it also scales with the number of support vectors used by the classifier. Other non-linear methods such as simple

decision trees and nearest-neighbor algorithm were not considered as they were found to be less performant (He et al., 2021). In all, a range of incrementally more sophisticated, and arguably more complicated to tune, classifiers was chosen.

Moreover, to denoise the output prediction of the classifier, a pair of thresholds operated as a transition trigger. Given the classifier probability $p(y = 1)[k]$, the predicted state $\tilde{y}[k]$ was given by:

$$\tilde{y}[k] = \begin{cases} 1 & \text{if } p[k] > T \quad \text{and } \tilde{y}[k-1] = 0 \\ 0 & \text{if } p[k] < (1 - r_T)T \quad \text{and } \tilde{y}[k-1] = 1 \end{cases} \quad (7)$$

where $T \in [0, 1]$ is the upper threshold, and $r_T \in [0, 1]$ is the position of the lower threshold relative to T . Thresholds are typically set at the optimal point of the receiver operating characteristic (ROC) curves, balancing the sensitivity and specificity of a classifier. Unfortunately, this curve cannot be drawn for a dual-threshold system. Instead, the geometric mean g between sensitivity and specificity was chosen to estimate an optimal trade-off (Cai et al., 2010). Given the detection true-positive rate (TPR) and false-positive rate (FPR), g was given by:

$$g = \sqrt{TPR(y, \tilde{y}) \times (1 - FPR(y, \tilde{y}))} \quad (8)$$

If $TPR = 0$, then $g = 0$. Likewise, if $FPR = 1$, then $g = 0$. Consequently, g only leans toward 1 if both objectives are balanced, thus avoiding the extreme situation where, for instance, the state-estimator only detects true positives or negatives. On a traditional ROC diagram, g would indicate a curve point sticking furthest toward $TPR = 1$ and $FPR = 0$. An eightfold cross-validation was employed to evaluate the average detection performance \bar{g} . The number of folds was chosen to be double that of the number of cores available on the processor (Intel® Core i7-7700K 4.20 GHz) to accelerate the evaluation of the folds through parallelization.

2.5. Decoder parameter tuning

Figure 1 illustrates different parameters introduced in the different processing stages and, Figure 2 the BO procedure. BO was implemented to optimize \bar{g} using the scikit-optimize library (Head et al., 2021). The parameters formed a bounded search space S from which a parameter vector $\rho \in S$ was sampled. To quantify the performance increase resulting from tuning, initial quantities, reflecting values typically found in other studies, formed a default parameter vector ρ_0 . A 3-tap Daubechies wavelet was used for the DWT by default (Rieder et al., 1998). A summary of all tuned parameter, search intervals and default values is given in Table 2. Given \bar{g} as a function of x , y , and ρ , the best parameters ρ^* were defined as:

$$\rho^* = \arg \max_{\rho \in S} \bar{g}(\rho, x, y) \quad (9)$$

The corresponding maximum cross-validation performance was noted as $\bar{g}^*(\rho^*, x, y)$. The BO scheme repeatedly fitted and evaluated the pipeline at each iteration (until 90 iterations were reached). The algorithm was seeded by first evaluating ρ_0 . In a “tell” step, the optimizer fitted a surrogate model which mapped S to \bar{g} . In an “ask” step, the optimizer, following the policy of a given acquisition function, suggested a new ρ (Shahriari et al., 2016).

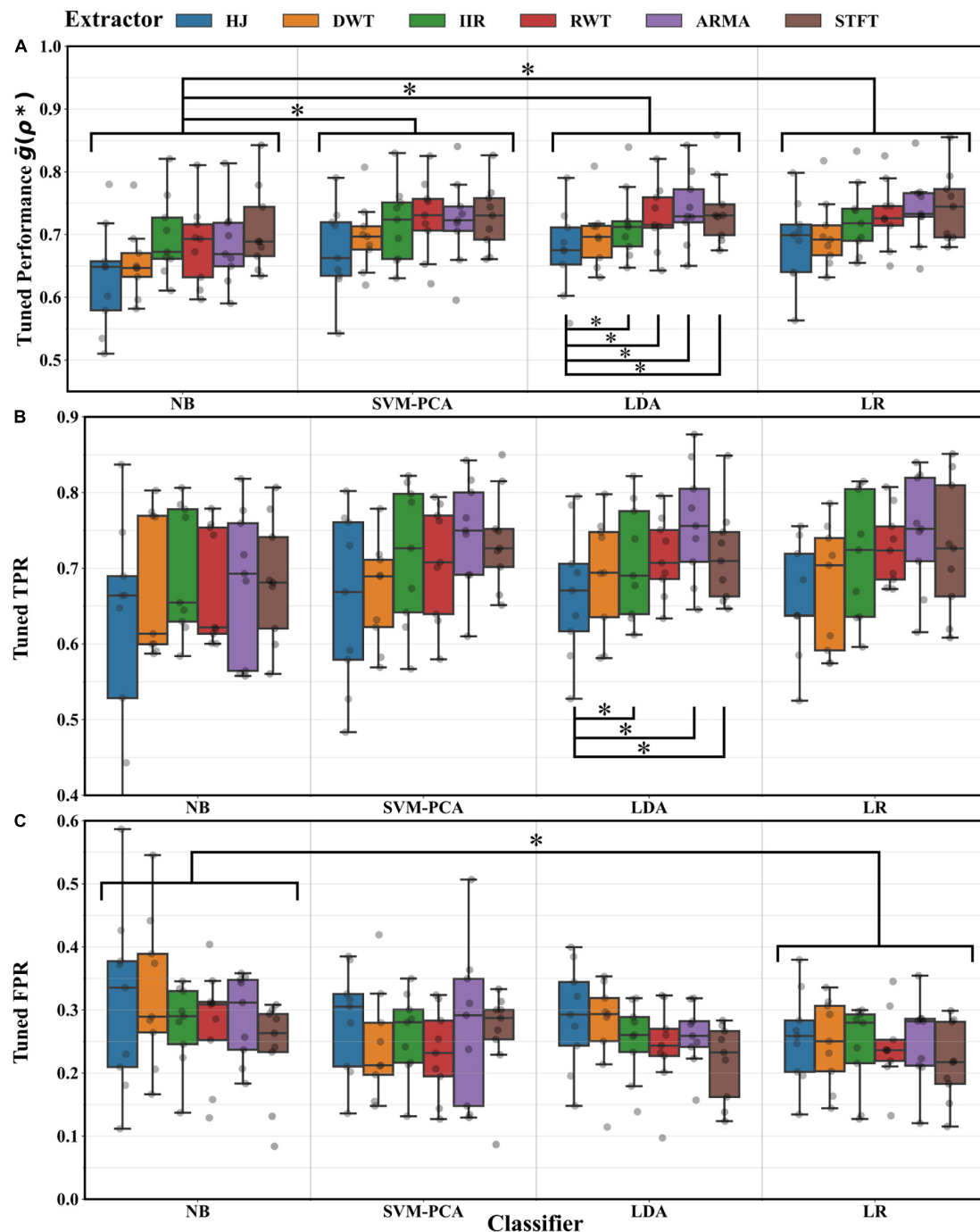


FIGURE 5

Maximal performance distribution across distributions and participants resulting from cross-validation averages after BO tuning for every extractor and classifier combination. (A) For the TPR and FPR geometric mean \bar{g} . Significant differences (*, Tukey test, $p < 0.05$) were found between decoders using the NB and those using other classifiers. Significant differences were also found between decoders using the HJ and those using other extractors. (B) For the TPR. Significant differences (*, Tukey test, $p < 0.05$) were found between decoders using the HJ and those using other extractors. (C) For the FPR. A significant difference was found between decoders using the NB and those using the LR classifier (*, Tukey test, $p < 0.05$).

A Gaussian-process with Matern Kernel (Bashashati et al., 2016) was chosen for the surrogate model alongside the Gaussian-process hedge acquisition function (Shahriari et al., 2016; Head et al., 2021).

Before optimizing every feature extractor and classifier pair, a preliminary comparison between BO and RD was conducted by optimizing the STFT-LR decoder for three participant recordings

(PD01R, PD05R, and PD06L), which represent well different optimized performance levels within the population sample. The optimization procedure was, in this case, repeated five times with different random seeds to produce average convergence trends. Individual trends were obtained by noting for every i -th iteration of the maximum performance obtained on or before said iteration

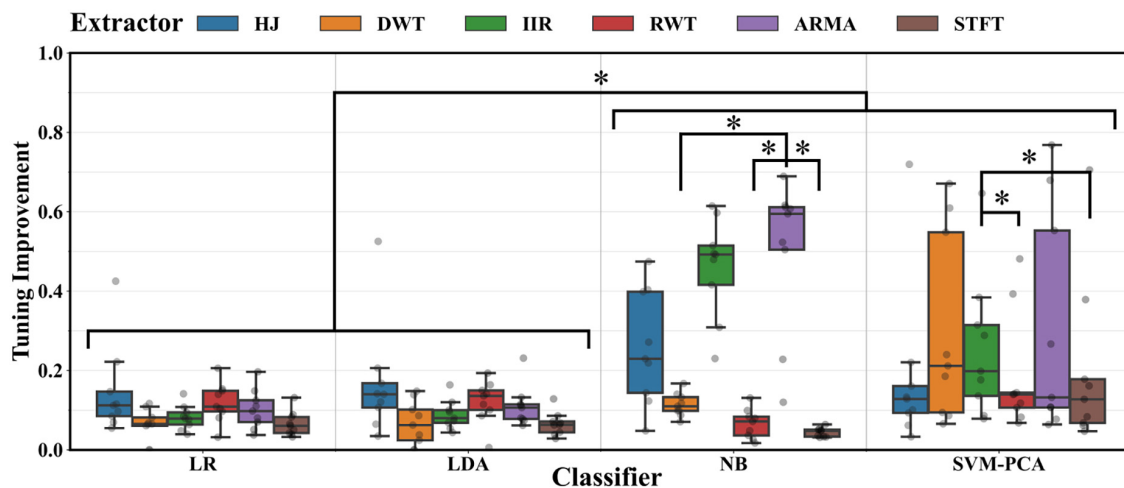


FIGURE 6

Distribution of performance improvement across participants when using BO tuned parameters over default parameters for every extractor and classifier combination. Significant differences (*, Tukey test, $p < 0.05$) were found between decoders using the NB, SVM-PCA on the one hand and those using the LDA, LR classifier on the other. Significant differences (*, Tukey test, $p < 0.05$) were found between decoders using the ARMA or IIR and those using other extractors.

\bar{g}_i^* . Average cumulated regret trends were also produced for both algorithms (Shahriari et al., 2016). A regret trend r_i was defined as:

$$\begin{cases} r_i = r_{i-1} + (\bar{g}_i^* - \bar{g}_i) \forall i = 1, \dots, N_i \\ r_0 = 0 \end{cases} \quad (10)$$

where \bar{g}_i is the score obtained on the i -th iteration and N_i , the total number of iterations. An initial 60 iterations were set. $\bar{g}_i^* - \bar{g}_i$ measures the quality of every decision the optimizer takes. Perfect decision-making would find a new optimum point at every step such that $\bar{g}_i^* - \bar{g}_i = 0$. Both convergence and regret trends gave insight into the progression of the optimization sequence.

2.6. Decoder evaluation and statistical analysis

Following the preliminary comparison, the gathered evaluation criteria summarized in Figure 2 were collected for every feature-extractor-classifier pair. The maximum cross-validation performance \bar{g}^* , TPR and FPR and the optimal parameter vector ρ^* were recorded. The significance of the absolute improvement $\bar{g}^* - \bar{g}(\rho_0, x, y)$ was evaluated using a Wilcoxon paired test. A Mood test was also performed between \bar{g}^* and $\bar{g}(\rho_0, x, y)$ to measure any reduction in inter-subject variance. Performance evaluation also included plotting the decoder output prediction on a test segment. Sensitivity analysis of the BO models was used to relate the contribution of each parameter to \bar{g}^* using the SALib library (Herman and Usher, 2017). S was sampled using the Saltelli method (Saltelli, 2002), and the BO surrogate model was analyzed using the Sobol method (Sobol, 2001). The Sobol indexes measured the ratio of individual parameter input variances to surrogate model output variance (Sobol, 2001). Total order indexes for every parameter were recorded to capture both first-order and higher-order interactions (Herman and Usher, 2017). Analysis of variance (ANOVA) and *post-hoc* Tukey pairwise tests

were applied to measure the dependence of $\bar{g}(\rho^*, x, y)$, TPR, FPR, improvement, ρ^* , τ_d , N_f and parameter sensitivity indexes on the choice of extractor and classifier. Statistical testing was conducted using the SciPy (Virtanen et al., 2019) and Statsmodel (Seabold and Perktold, 2010) toolboxes. Figures were produced using the Seaborn (Waskom, 2021) and Matplotlib (Hunter, 2007) toolboxes.

3. Results

3.1. Preliminary comparison between random search and Bayesian optimization

Figure 4 presents averaged convergence and cumulated regret trends computed for three representative participant recordings. The trends indicated that the RD scheme was very effective at uncovering good initial parameter vectors, whilst BO had a rather slower convergence rate starting from ρ_0 . Nonetheless, after a few iterations, the convergence rate and plateau of the BO algorithm overtook that of the RD scheme. The cumulated regret trends confirmed that BO mapped sufficiently well the parameter space with each iteration, thus decreasing the $\bar{g}_i^* - \bar{g}_i$ through informed decision-making. As expected, since RD is memoryless, that gap remained constant, leading to a linear cumulation of decision errors.

Thus, for the presented decoder tuning problem, BO was better at ordering good solutions (i.e., \bar{g}_i is more likely to be superior to \bar{g}_{i-1} , which is important for sequential learning) and at fine-tuning in later iterations. Consequently, for the remainder of the tuning runs, to bootstrap the BO search starting at ρ_0 , five additional random samples were drawn to seed the initial fit of the BO surrogate model. Thirty more iterations (adding to a total of $N_i = 90$ iterations) were added to allow for further fine-tuning.

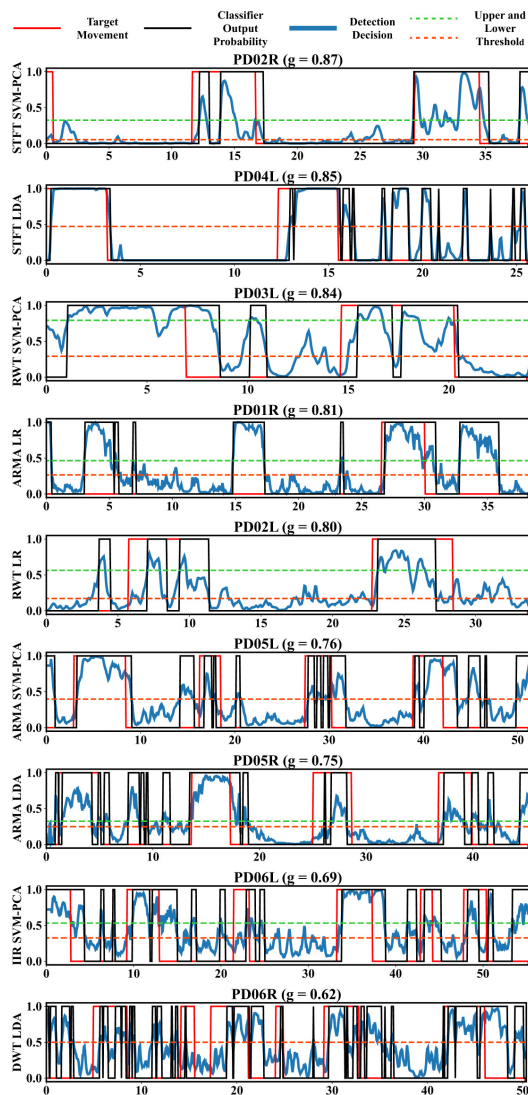


FIGURE 7

Best decoded test segment (10% of each recording length) for every participant with associated test score and extractor, classifier pair. Thresholds were set in an attempt to optimize detection rate based on the dynamics (range, latency, noisiness, or sudden changes) of the decoder output probability, where possible larger deadband helped avoid spurious transitions. The decoder had the option to remove the deadband by setting r_T to 0.

3.2. Decoding performance

Figure 5 reports the optimal cross-fold average $\bar{g}(\rho^*, x, y)$, TPR and FPR achieved after BO tuning for every extractor and classifier pair. Medians laid within 0.6–0.75 for the \bar{g} and TPR, and within 0.20–0.35 for the FPR. Combinations of ARMA STFT extractors with LDA and LR classifiers achieved the highest average \bar{g} performance of 0.74 ± 0.06 (mean \pm SD). ANOVA testing on \bar{g} suggested that the choice of classifier and extractor impacted performance ($p < 0.01$). *Post-hoc* testing showed that all classifiers and extractors performed comparatively well except for the NB ($p < 0.05$) and the HJ ($p < 0.05$). The TPR depended only on the selected extractor (ANOVA, $p < 0.01$), with the HJ underperforming relative to the ARMA, IIR, and STFT (*post-hoc*

tests, $p < 0.05$). Conversely, the FPR depended on the choice of classifier (ANOVA, $p < 0.05$), with the NB underperforming relative to the LR (*post-hoc* tests, $p < 0.05$).

Figure 6 reports the improvement in the overall decoding performance with the BO-based parameter optimization compared to that with default often used parameters fixed across all participants $[\bar{g}(\rho^*, x, y) - \bar{g}(\rho_0, x, y)]$. Paired testing showed that this gap was significant in every case (Wilcoxon, $p < 0.05$). ANOVA testing showed that the improvement gap depended again on the classifier and extractor pair ($p < 0.01$). *Post-hoc* testing indicated that improvement margins were tighter for the LR and LDA classifiers ($p < 0.05$). Default parameter solutions for the SVM-PCA and NB led to poorer decoding solutions. Lastly, no significant reduction in inter-subject variance was measured between default and optimized parameter performance (Mood, $p > 0.05$).

Figure 7 presents the best-decoded test performance for every user. The classifier output signal $p[k]$ varied noticeably in terms of quality and temporal dynamics throughout the cohort. Effective detection occurred with high and sufficiently sustained p throughout the action with manageable levels of baseline noise (when $y[k] = 0$) and action noise (when $y[k] = 1$), such as observed in PD03L, PD04L, PD02L, and PD01R. In other instances, p adopted a more impulsive behavior, such as in PD02R or PD02L, making a full capture of the action difficult. Although the upper threshold prevented high baseline noise from triggering the detection, such as in PD03L, PD04L, or PD01R, it could not contend with very large spurious impulses outside of the action (when $y[k] = 0$) as those found in PD04L, PD05R, PD06R, and PD06L. Conversely, spurious drops would also occur within the action (when $y[k] = 1$) such as in PD05L, triggering the lower threshold.

3.3. Decoder parameterization

Optimal parameter distributions were tested using ANOVA. The test results suggested that the thresholds (T and r_T) did not depend on either the classifier or the extractor ($p > 0.01$) but rather on the intrinsic signal properties of individual recordings. The number of channels (N_c) depended on the choice of classifier (ANOVA, $p < 0.01$) but not of the extractor. *Post-hoc* testing suggested that fewer channels were selected using the NB classifier rather than the LR or LDA ($p < 0.05$). The number of overlaps between windows (n_o) was dependent on both estimator and extractor choices (ANOVA, $p < 0.01$). The STFT adopted a significantly larger extraction window length (L) than the ARMA, IIR, and HJ extractors; the LR, than the LDA and NB classifiers (*post-hoc* tests, $p < 0.05$). Likewise, the choice of feature window length (N_w) depended on the extractor (ANOVA, $p < 0.01$) but not on the classifier selection. Notably, the STFT, IIR, and HJ required significantly larger N_w than the ARMA extractor (*post-hoc* tests, $p < 0.05$). **Figure 8** presents distributions for the decoder analysis window (τ_d) and the total number of features employed for decoding (N_f , section 2.3). τ_d and N_f depended on the choice of extractor and classifier (ANOVA, $p < 0.01$). Because of the smaller optimum N_w and n_o , the ARMA extractor and the NB classifier required significantly smaller windows than the STFT extractor and LR classifier (*post-hoc* tests, $p < 0.05$). Likewise, because of smaller

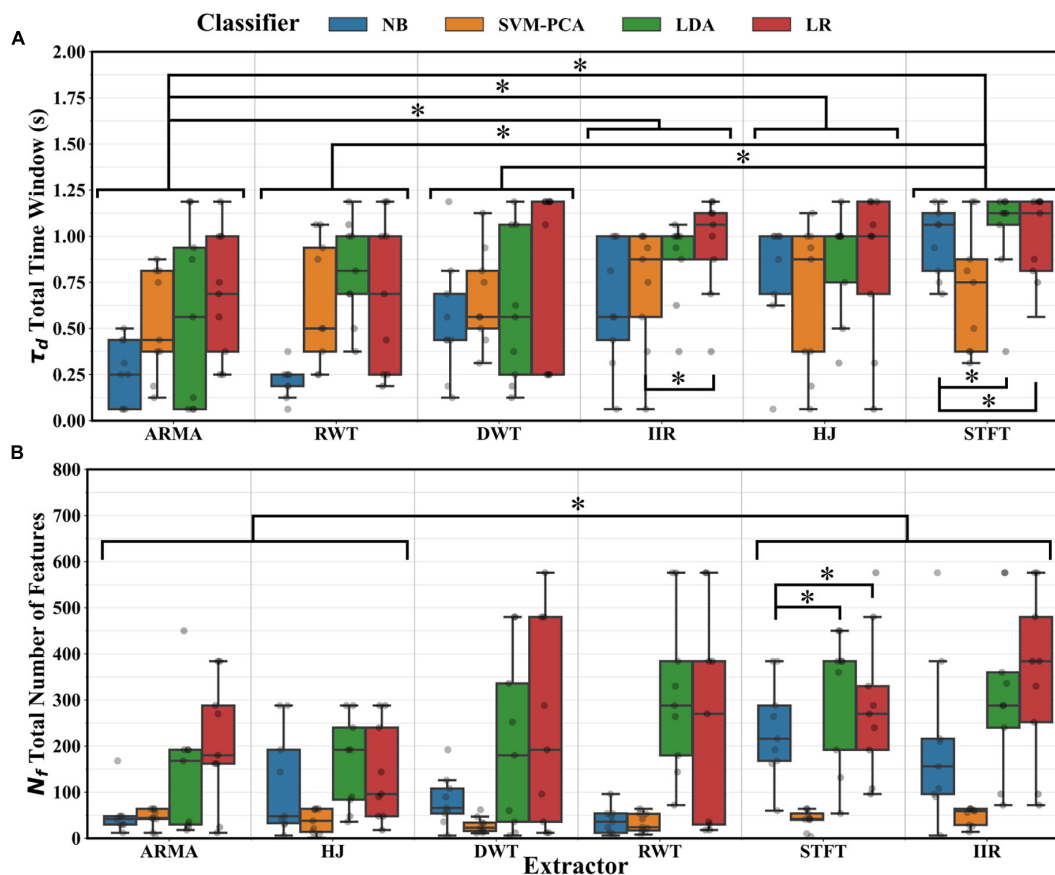


FIGURE 8

(A) Distributions across participants of the total decoder analysis window τ_d using optimal parameters. Significant reductions (*, Tukey test, $p < 0.05$) were found between decoders using the ARMA, RWT, or DWT extractors and those using the IIR, HJ, or the STFT extractors. Significant differences (*, Tukey test, $p < 0.05$) were also found between decoders using the NB or LR and those using other classifiers. (B) Distributions of the total number of features N_f used for decoding. Significant reduction (*, Tukey test, $p < 0.05$) were found between decoders using the ARMA or HJ and those using the STFT or IIR extractors. Setting aside the SVM-PCA, significant reduction (*, Tukey test, $p < 0.05$) were also found between decoders using the NB and those using the LDA or LR classifiers.

N_w , N_c or N_i , the decoders using the ARMA and HJ extractors or the NB classifier required significantly fewer features than the STFT extractor or LR classifier (*post-hoc* tests, $p < 0.05$). Lastly, there was no dependence between optimal extractor parameters and classifier choice and no dependence between classifier parameters and extractor choice.

The index distributions yielded from the sensitivity analysis are presented in **Figure 9** for each parameter. Considering the subset of common parameters (n_o , N_c , N_w , r_T , T), ANOVA testing suggested that the importance indexes varied depending on the parameter type and the classifier selected ($p < 0.01$), but not on the extractor ($p > 0.05$). *Post-hoc* testing showed that the upper and lower thresholds in the final decoding stage (T and r_T) were significantly more important than all other parameters ($p < 0.05$). The window overlap (n_o) was significantly more important than the number of channels selected (N_c) and the window length (N_w) ($p < 0.05$), despite clear preferential settings depending on the selected classifier or extractor. *Post-hoc* tests also showed that the NB and the LDA attributed more importance to those parameters than the LR classifier ($p < 0.05$). *Post-hoc* tests were performed for extractor combinations with the LR classifier with all available parameters. Aside from T , r_T , the regularization constant, C ,

was the most important parameter ($p < 0.05$), with the unique exception of the ARMA λ parameter. In contrast, with the NB classifier, all parameters had shared similar levels of importance ($p > 0.05$), except for n_o , which was more important with ARMA and IIR pairing ($p < 0.05$). The remaining parameters had comparable importance with the LDA and SVM-PCA classifiers. Regularization parameters did not hold as much importance as they did with LR classifiers. Lastly, extractor-specific parameters other than λ (the model order, N_{AR} , N_{MA} , for ARMA, the lattice parameters, θ_1 , θ_2 for DWT and the window shape factor, β_K , for the STFT) were never significantly more important than other parameters (omitted some exceptions, **Figure 9**).

3.4. Computational benchmark for feature extraction

Figure 10 presents a benchmark to evaluate the computational time taken to process features for a single decoder step for each method. All of them were sufficiently fast to operate in real-time. A large overhead was observed when features were extracted

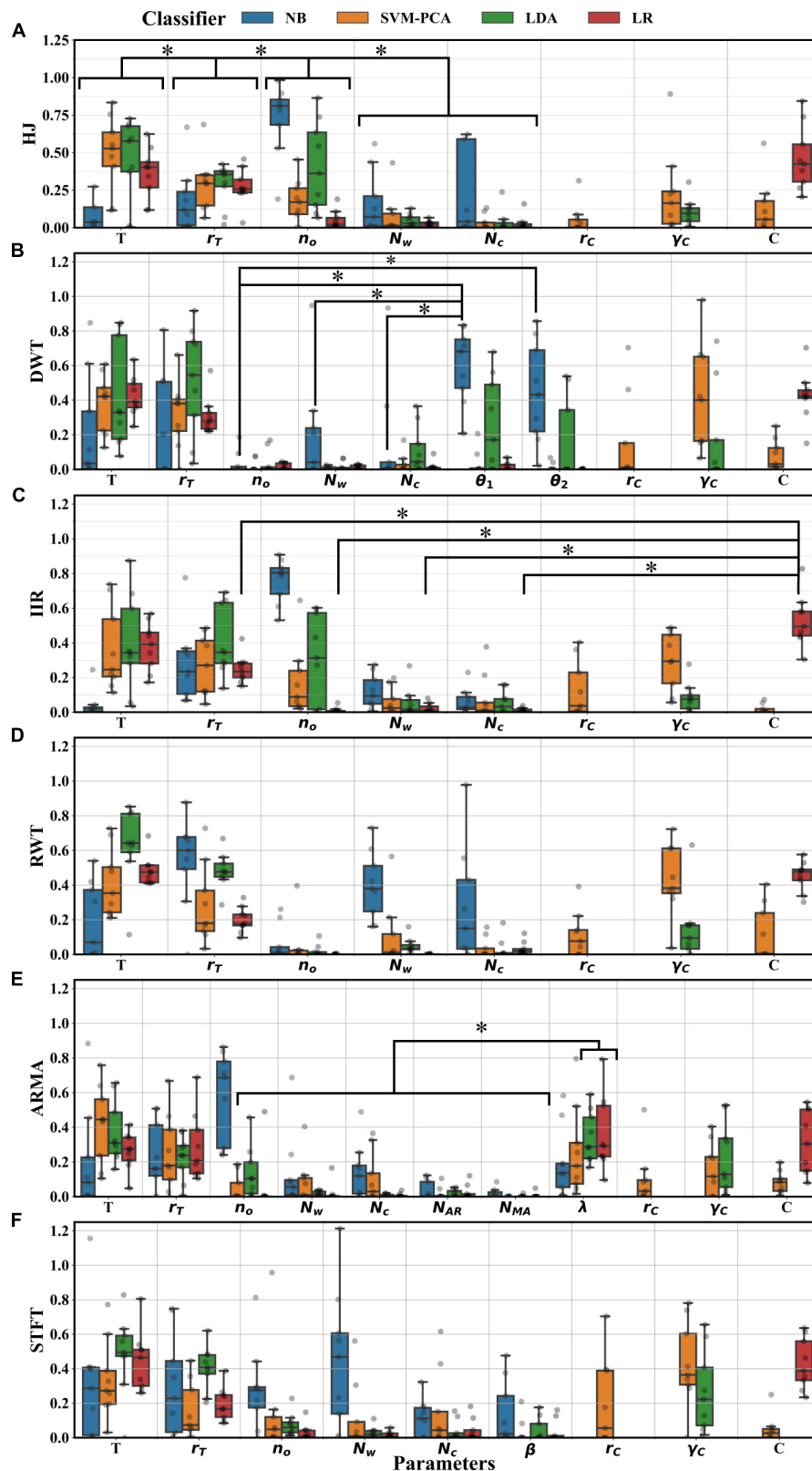


FIGURE 9

Distributions of sensitivity index for every tuned parameter across participants. Index resulted from a Sobol sensitivity analysis of each BO surrogate model. (A) Index distribution for decoders using the HJ; (B) DWT; (C) IIR; (D) RWT; (E) ARMA; and (F) STFT extractors. Significant decoder reduction (*, Tukey test, $p < 0.05$) in importance were found between T , r_T , n_o and lastly N_c , N_w . In most extractor pairing, significant differences (*, Tukey test, $p < 0.05$) were found between most parameters and the regularization constant C of the LR classifier (shown in C). Significant differences (*, Tukey test, $p < 0.05$) were also found between the importance of λ when using the LR and LDA classifier and other parameter (B). When paired with the NB, significant differences (*, Tukey test, $p < 0.05$) were found between lattice parameter θ_1 , θ_2 , and other parameters (E).

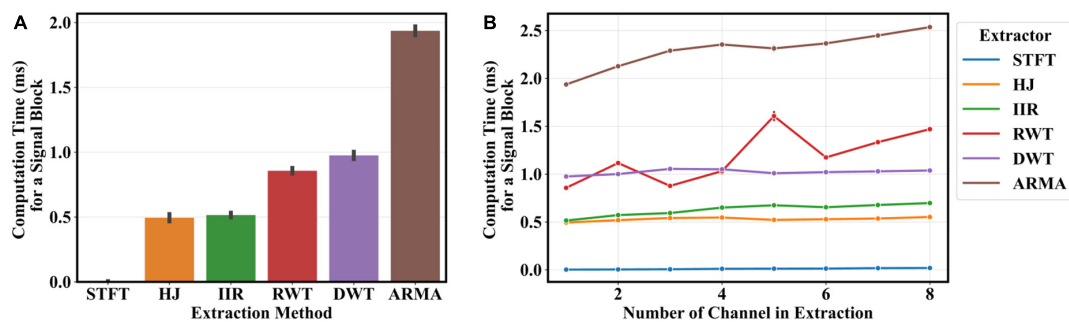


FIGURE 10

Computational time benchmark for every feature extraction method (using an Intel® Core™ i7-7700K 4.20 GHz); each method was evaluated using a 100 s long, 512 Hz (f_i), Gaussian random noise signal. (A) The extraction was repeated 30 times using default setting parameters and a single channel signal. The average time taken to complete a single computational step (process a block of $R = 32$ signal points) at 16 Hz (f_o) was estimated. (B) The benchmark was reproduced by incrementally adding channels to a total of 8. Each method benchmark is significantly inferior to the refreshing time (62.5 ms) between decoder output updates. All methods were custom implemented using the NumPy library, except the STFT which used a specialized function from the SciPy library. Error-bars indicate the SD taken across the 30 extractions.

from a single channel, most likely associated with slow array manipulation in python at each time-step. More complex methods such as ARMA and DWT were more penalized than others (more linear transformation for the ARMA and multi-level processing for the DWT extractor). The DWT should be, in practice, far more efficient (Farina et al., 2007). Nonetheless, extraction processing time was found to scale favorably as multi-dimensional signals were introduced. More demanding processes, such as the RWT, ARMA and, to an extent, the IIR extractors, had a notable increase in computational time with every added channel. Other methods, such as the DWT and HJ were more computationally efficient. The benchmark proposed in this study is specific to the programming language, operating system and processor. Different hardware and software optimization considerations should ultimately be made based on the target deployment device and the selected method.

4. Discussion

This investigation demonstrates the benefits of automatic parameter tuning using BO for STN-LFPs asynchronous decoding. The parameterization of a relatively complex decoding pipeline optimized a task-relevant objective function, and BO significantly increased task performance from default parameter settings and substituted part of the expert knowledge required for calibration. In addition, BO allowed us to systematically evaluate and compare different combinations of feature extraction and classification methods for decoding. It also helped to uncover the parameters that had the most impact on the decoding performance, such as the classifier thresholds and the window overlaps. In particular, the classifier thresholds were configured differently depending on the individual properties of the STN-LFPs signals.

4.1. Performance comparison and study limitations

The optimum performance obtained in this study is comparable to the results of past studies that have employed

similar asynchronous and real-time constraints to decode STN-LFPs (Mace et al., 2014; Niketeghad et al., 2018). The limitation remains that stable detection can vary significantly across participants. We note, however, that BO did not reduce inter-subject variance, indicating that lower-performing cases may be due to a lack of information related to the decoding target in the recorded signals. Moreover, the performance increase can be achieved from the simultaneous optimization of the extractor and classifier, indicating that default parameters may not capture a priori expertise knowledge. A clear benefit of this approach was observed principally in the NB classifier. Nonetheless, end-to-end decoder optimization may not have been the most efficient approach in the proposed architecture and decoding task context. The tuning problem could potentially be broken down into several stages, e.g., one for feature extraction (optimizing some desirable attributes of X) and another for state estimation (optimizing \bar{g}). Separating these two stages would also be more practical to implement estimators with more computational intensive fitting such as the kernel SVM (and avoid resorting to PCA to compress X).

Furthermore, the decoding of movement in this investigation provides a basis to apply the method to other DBS targets, state types (e.g., tremor, gait, etc.), and BCI tasks to further gauge the impact of tuning. Online testing is needed to perform decoding sequentially with stimulation in an adaptive configuration (He et al., 2021), continuous tuning (Grado et al., 2018), and for deployment, accounting for the fixed computational and memory budget of an embedded device (Zhu et al., 2020). Finally, findings in this study can further guide algorithm selection, and a BO approach could help ease deployment by rethinking the task objective with these budgeting constraints in mind.

4.2. Improving the tuning method

The optimization process involved a range of parameterization trade-offs. For instance, channel ranking or other decomposition strategies are not a strict guarantee of improved performance through a reduction in classifier overfitting (Shah et al., 2018).

Except for the NB, the BO algorithm kept adding channels despite diminishing returns in performance gain (Flint et al., 2012). A multi-objective approach or optimization constraints could narrow the set of reasonable tuning solutions. Penalty for delayed detection, output chattering, or model complexity could be introduced (Mace et al., 2014). Parameters such as the number of channels selected (N_c) and the window length (N_w) could then become more relevant. Further practical improvements could also be made to the BO algorithm implementation by introducing robust convergence criteria (Bashshati et al., 2016) and more adaptive acquisition functions (Grado et al., 2018).

4.3. Tuning and usage of the proposed feature extractors

For feature extraction, parameters influencing the temporal smoothness of the feature signals, such as the number of overlaps between feature extraction windows n_o and the ARMA forgetting factor λ were important to decoding. A second-order Kalman filter (Chisci et al., 2010; Yao et al., 2020) could be tuned using BO and applied either in the feature space for further smoothing or at the classifier output to help stabilize the state transition. Reducing the decoder output frequency f_o could also help reduce higher-frequency noise at the cost of a slower refreshing rate. An intermediate cost-function measuring signal-to-noise ratio could be employed to tune extractor parameters in the first optimization stage. Feature extractors overall delivered comparable levels of performance. With hindsight, testing other signal features that were not purely spectral-based (Bakstein et al., 2012; Wang et al., 2018) or different combinations of feature bands might have been more intriguing (Tan et al., 2016). The BO algorithm could have been used to select the combination of features better suited to decoding.

Although the extractor was not critically associated with the detection performance, this work suggests that the choice of extractor still influences extraction and feature window lengths and the number of features used for classification, elements important to the decoder design. Some extractors may thus be better suited to some decoding tasks than others. For instance, the ARMA extractor was among the more performant and instantaneous methods. It can also be helpful if connectivity features (Mamun et al., 2015) or instantaneous phase estimation (Foffani et al., 2005) are of interest. However, it comes at the cost of a specialized implementation and a more computationally intensive process. The STFT, in comparison, is a practical, efficient and versatile extractor. Since the window shape only had a limited influence on decoding, using, for instance, a recursive formulation of the exponential window (Grado et al., 2017) could help minimize windowing lag. Although, the performance with the RWT did not supersede that of the STFT, it has the benefit of combining a narrow-band filter in a compact IIR form with a complex phase estimator, which could be useful to isolate a single or few specific sub-band phase and power. Otherwise, the additional computational burden of a large number of relatively long complex filters associated with a full-signal decomposition does not seem justify. Smaller filter banks, such as the IIR extractor, are relevant to embedded applications (Zhu et al., 2020). The mutual information-based channel selection scheme could be extended to incorporate

only informative power bands. Selected individual features across contacts is here more practical as only the relevant filters can be activated on a given channel greatly reducing the computational burden. Given the comparatively short τ_d for the DWT and its computational efficiency, BO could permit the optimization of longer wavelets which could help match the performance of other extractors. The packet transform could introduce a sparser and more adaptive signal representation (Mamun et al., 2015; Luo et al., 2018). Lastly, the HJ parameters were not sufficiently informative and should only be used as complementary features for movement decoding (Shah et al., 2018).

4.4. Tuning and selecting state estimators for decoding

The double-thresholds system played an important role in providing stable state transitions. It was observed that the upper threshold was generally set above spurious events to prevent false-positive triggering. A higher triggering level delayed detection, especially if the transient response of the feature extractor and classifier combination was slow. A lower second threshold, close to the baseline activity of the classifier output, was preferable. It aligned the detection and the action ending better and prevented a spurious transition if the classifier output suddenly dipped during an action. BO could thus help calibrate more complex state machines required for adaptive DBS (He et al., 2021).

Aside from the NB classifier, there was no substantial difference between the classifier types in terms of performance. However, not every classifier was best suited for the proposed detection task. Fitting the SVM-PCA classifier was more computationally demanding, given the temporal longitudinal aspect of the dataset. Adjusting f_o could help in this regard (Golshan et al., 2016, 2018, 2020). In addition, the SVM algorithm does not yield prediction probabilities. Given the importance of thresholding for state transition, those cannot be omitted and thus have to be estimated in an additional *post hoc* process (Platt, 1999). LR and the LDA classifiers may be thus preferable with the LR was more easily calibrated through regularization. Ensemble methods, using a set of classifiers rather than a single unit (in particular decision trees), have proven to outperform other machine-learning methods when deployed on small size, complex, and noisy BCI datasets (Gruenwald et al., 2019; Zhu et al., 2020) and could be easily implemented using the proposed method. The method could also be applicable to neural-network based decoders (Ahmadi et al., 2019; Golshan et al., 2020) specifically tuning of decoder specific parameters such as selected channels, thresholds or window convolution size.

5. Conclusion

We have introduced a decoding architecture based on BO to tune different parameters in the feature extraction, channel selection, classification and state-transition stages of an asynchronous decoder to identify movement signatures from STN-LFPs signals recorded in the deep brain. This architecture allows self-tuning in a computationally efficient implementation

for real-time deployment. Although asynchronous decoding from STN-LFPs remains highly subject-specific, we demonstrate that BO improves decoding performance across participants. In addition, BO provided in-depth details regarding the tuning process and the relevance of each parameter which could help inform future decoder design iterations. For asynchronous decoding, time-frequency signal extraction (regardless of the chosen approach) should render smoother feature signals, and state estimators should be robust to spurious activity change. An increased number of features or decoding channels should be budgeted relative to incremental performance gains and model complexity. Optimal individual parameters are essential, especially where substantial inter-subject variability is present, to best adapt to the characteristics of the neural signals. The method presented in this study could also be applied to decoding other bulkier movements, arousal, and pathological states to construct feedback mechanisms for adaptive DBS systems and serve other BCI applications. Future work should include further investigation of other DBS-recorded LFPs signals and further online testing.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: <https://data.mrc.ox.ac.uk/mrcbndu/datasets/search>.

Ethics statement

The studies involving human participants were reviewed and approved by the South Central – Oxford C Research Ethics Committee 18/SC/0006. The patients/participants provided their written informed consent to participate in this study.

Author contributions

HT and SH gathered the data used in the computational experiment. TM designed and implemented the computational

experiment, analyzed the experimental results, and wrote the manuscript. SH, HT, and RV reviewed and edited the manuscript. All authors contributed to the article and approved the submitted version.

Funding

This work was supported by the Medical Research Council (MR/P012272/1 and MC/UU/12024/1), the Rosetrees Trust, the National Institute of Health Research Oxford Biomedical Research Centre, the Engineering and Physical Sciences Research Council (EP/L016737/1), EPSRC Doctoral Training Centre for Neurotechnology at Imperial College London, and the UK Dementia Research Institute (NIHR202133). SH was supported by BRAIN Non-Clinical Post-Doctoral Fellowship (HMR04170).

Acknowledgments

We express our warmest gratitude to the research participants in this study.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

- Ahmadi, N., Constandinou, T. G., and Bouganis, C.-S. (2019). "Decoding hand kinematics from local field potentials using long short-term memory (LSTM) network," in *2019 9th international IEEE/EMBS conference on neural engineering (NER)*, (Piscataway, NJ: IEEE), 415–419. doi: 10.1109/NER.2019.8717045
- Anzak, A., Tan, H., Pogoyan, A., Foltynie, T., Limousin, P., Zrinzo, L., et al. (2012). Subthalamic nucleus activity optimizes maximal effort motor responses in Parkinson's disease. *Brain* 135, 2766–2778. doi: 10.1093/brain/awr183
- Bakstein, E., Burgess, J., Warwick, K., Ruiz, V., Aziz, T., and Stein, J. (2012). Parkinsonian tremor identification with multiple local field potential feature classification. *J. Neurosci. Methods* 209, 320–330. doi: 10.1016/j.jneumeth.2012.06.027
- Bashashati, H., Ward, R. K., and Bashashati, A. (2016). User-customized brain computer interfaces using Bayesian optimization. *J. Neural Eng.* 13:026001. doi: 10.1088/1741-2560/13/2/026001
- Benabid, A. L., Costecalde, T., Eliseyev, A., Charvet, G., Verney, A., Karakas, S., et al. (2019). An exoskeleton controlled by an epidural wireless brain-machine interface in a tetraplegic patient: A proof-of-concept demonstration. *Lancet Neurol.* 18, 1112–1122. doi: 10.1016/S1474-4422(19)30321-7
- Bergstra, J., and Bengio, Y. (2012). Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* 13, 281–305. doi: 10.5555/2188385.2188395
- Brown, P. (2003). Oscillatory nature of human basal ganglia activity: Relationship to the pathophysiology of Parkinson's disease. *Mov. Disord.* 18, 357–363. doi: 10.1002/mds.10358
- Cai, Y., Chow, M., Lu, W., and Li, L. (2010). "Evaluation of distribution fault diagnosis algorithms using ROC curves," in *IEEE PES General Meeting, PES 2010*, (Piscataway, NJ: IEEE), 1–6. doi: 10.1109/PES.2010.5588154
- Camara, C., Isasi, P., Warwick, K., Ruiz, V., Stein, J., Aziz, T., et al. (2015). Resting tremor classification and detection in Parkinson's disease patients. *Biomed. Signal Proc. Control* 16, 88–97. doi: 10.1016/j.bspc.2014.09.006
- Cassidy, M., Mazzone, P., Oliviero, A., Insola, A., Tonali, P., Lazzaro, V. D., et al. (2002). Movement-related changes in synchronization in the human basal ganglia. *Brain* 125, 1235–1246. doi: 10.1093/brain/awf135

- Chen, Y., Gong, C., Hao, H., Guo, Y., Xu, S., Zhang, Y., et al. (2019). Automatic sleep stage classification based on subthalamic local field potentials. *IEEE Tran. Neural Syst. Rehabil. Eng.* 27, 118–128. doi: 10.1109/TNSRE
- Chisci, L., Mavino, A., Perferi, G., Sciandrone, M., Anile, C., Colicchio, G., et al. (2010). Real-time epileptic seizure prediction using AR models and support vector machines. *IEEE Trans. Biomed. Eng.* 57, 1124–1132. doi: 10.1109/TBME.2009.2038990
- Farina, D., Nascimento, O. F., Lucas, M., and Doncarli, C. (2007). Optimization of wavelets for classification of movement-related cortical potentials generated by variation of force-related parameters. *J. Neurosci. Methods* 162, 357–363. doi: 10.1016/J.JNEUMETH.2007.01.011
- Flint, R. D., Lindberg, E. W., Jordan, L. R., Miller, L. E., and Slutzky, M. W. (2012). Accurate decoding of reaching movements from field potentials in the absence of spikes. *J. Neural Eng.* 9:046006. doi: 10.1088/1741-2560/9/4/046006
- Foffani, G., Bianchi, A., Baselli, G., and Priori, A. (2005). Movement-related frequency modulation of beta oscillatory activity in the human subthalamic nucleus. *J. Physiol.* 568, 699–711. doi: 10.1113/jphysiol.2005.089722
- Golshan, H. M., Hebb, A. O., and Mahoor, M. H. (2020). LFP-Net: A deep learning framework to recognize human behavioral activities using brain STN-LFP signals. *J. Neurosci. Methods* 335:108621. doi: 10.1016/j.jneumeth.2020.108621
- Golshan, H. M., Hebb, A. O., Hanrahan, S. J., Nedrud, J., and Mahoor, M. H. (2018). A hierarchical structure for human behavior classification using STN local field potentials. *J. Neurosci. Methods* 293, 254–263. doi: 10.1016/j.jneumeth.2017.10.001
- Golshan, H. M., Hebb, A. O., Hanrahan, S. J., Nedrud, J., Mahoor, M. H., et al. (2016). “A Multiple Kernel Learning approach for human behavioral task classification using STN-LFP signal,” in *2016 38th Annual international conference of the IEEE engineering in medicine and biology society (EMBC)*, (Piscataway, NJ: IEEE), 1030–1033. doi: 10.1109/EMBC.2016.7590878
- Grado, L. L., Johnson, M. D., and Netoff, T. I. (2017). The sliding windowed infinite fourier transform. *IEEE Signal. Proc. Magaz.* 34, 183–188. doi: 10.1109/MSP.2017.2718039
- Grado, L. L., Johnson, M. D., and Netoff, T. I. (2018). Bayesian adaptive dual control of deep brain stimulation in a computational model of Parkinson’s disease. *PLoS Comput. Biol.* 14:e1006606. doi: 10.1371/journal.pcbi.1006606
- Gruenewald, J., Kapeller, C., Kamada, K., Scharinger, J., Guger, C., et al. (2017). “Optimal bandpower estimation and tracking via Kalman filtering for real-time Brain-Computer Interfaces,” in *International IEEE/EMBS conference on neural engineering, NER*, (Shanghai: IEEE), 605–608. doi: 10.1109/NER.2017.8008424
- Gruenewald, J., Znobishchev, A., Kapeller, C., Kamada, K., Scharinger, J., and Guger, C. (2019). Time-variant linear discriminant analysis improves hand gesture and finger movement decoding for invasive brain-computer interfaces. *Front. Neurosci.* 13:901. doi: 10.3389/fnins.2019.00901
- Harris, C., Millman, K., van der, W. S., Gommers, R., Virtanen, P., et al. (2020). Array programming with NumPy. *Nature* 585, 357–362. doi: 10.1038/s41586-020-2649-2
- He, S., Baig, F., Mostofi, A., Pogoyan, A., Debarros, J., Green, A. L., et al. (2021). Closed-Loop deep brain stimulation for essential tremor based on thalamic local field potentials. *Mov. Disord.* 36, 863–873. doi: 10.1002/mds.28513
- He, S., Syed, E., Torrecillos, F., Tinkhauser, G., Fischer, P., Pogoyan, A., et al. (2019). “Beta Oscillation-targeted neurofeedback training based on subthalamic lfps in parkinsonian patients,” in *International IEEE/EMBS conference on neural engineering, NER, 2019-March(0)*, (San Francisco, CA: IEEE), 81–84. doi: 10.1109/NER.2019.8717176
- Head, T., Kumar, M., Nahrstaedt, H., Louppe, G., Shcherbatyi, I., et al. (2021). *scikit-optimize/scikit-optimize*. Honolulu: Zenodo. doi: 10.5281/ZENODO.5565057
- Herman, J., and Usher, W. (2017). SALib: An open-source python library for sensitivity analysis. *J. Open Source Softw.* 2:97. doi: 10.21105/joss.00097
- Hirschmann, J., Schoffelen, J. M., Schnitzler, A., and Gerven, M. A. (2017). Parkinsonian rest tremor can be detected accurately based on neuronal oscillations recorded from the subthalamic nucleus. *Clin. Neurophysiol.* 128, 2029–2036. doi: 10.1016/j.clinph.2017.07.419
- Hjorth, B. (1970). Technical EEG analysis based on contributions on time domain properties. *Electroencephalogr. Clin. Neurophysiol.* 28, 306–310. doi: 10.1016/0013-4694(70)90143-4
- Hunter, J. (2007). Matplotlib A 2D Graphics Environment. *Comput. Sci. Eng.* 9, 90–95.
- Lemaître, G., Nogueira, F., and Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *J. Mach. Learn. Res.* 18, 1–5. doi: 10.5555/3122009.3122026
- Little, S., Pogoyan, A., Neal, S., Zavala, B., Zrinzo, L., Hariz, M., et al. (2013). Adaptive deep brain stimulation in advanced Parkinson disease. *Ann. Neurol.* 74, 449–457. doi: 10.1002/ana.23951
- Luo, H., Huang, Y., Du, X., Zhang, Y., Green, A. L., Aziz, T. Z., et al. (2018). Dynamic neural state identification in deep brain local field potentials of neuropathic pain. *Front. Neurosci.* 12:237. doi: 10.3389/fnins.2018.00237
- Mace, M., Yousif, N., Naushahi, M., Abdullah-Al-Mamun, K., Wang, S., Nandi, D., et al. (2014). An automated approach towards detecting complex behaviours in deep brain oscillations. *J. Neurosci. Methods* 224, 66–78. doi: 10.1016/J.JNEUMETH.2013.11.019
- Mamun, K. A., Mace, M., Lutman, M. E., Stein, J., Liu, X., Aziz, T., et al. (2015). Movement decoding using neural synchronization and inter-hemispheric connectivity from deep brain local field potentials. *J. Neural Eng.* 12:056011. doi: 10.1088/1741-2560/12/5/056011
- Martineau, T., Martineau, T., He, S., Vaidyanathan, R., Brown, P., Tan, H., et al. (2020). “Optimizing time-frequency feature extraction and channel selection through gradient backpropagation to improve action decoding based on subthalamic local field potentials,” in *Proceedings of the annual international conference of the IEEE engineering in medicine and biology society, EMBS*, (Montreal, QC: IEEE), 3023–3026. doi: 10.1109/EMBC44109.2020.9175885
- Nandy, A., As Alahe, M., Nasim Uddin, S. M., Alam, S., Al Nahid, A., et al. (2019). “Feature extraction and classification of EEG signals for seizure detection,” in *1st International conference on robotics, electrical and signal processing techniques, ICREST 2019*, (Dhaka: IEEE), 480–485. doi: 10.1109/ICREST.2019.8644337
- Nicolas-Alonso, L. F., and Gomez-Gil, J. (2012). Brain computer interfaces, a review. *Sensors* 12, 1211–1279. doi: 10.3390/s120201211
- Niketghad, S., Hebb, A. O., Nedrud, J., Hanrahan, S. J., and Mahoor, M. H. (2018). Motor task detection from human STN using interhemispheric connectivity. *IEEE Trans. Neural Syst. Rehabil. Eng.* 26, 216–223. doi: 10.1109/TNSRE.2017.2754879
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in python. *J. Mach. Learn. Research* 12, 2825–2830.
- Platt, J. (1999). Probabilistic outputs for svm and comparison to regularized likelihood methods. *Adv. Large Marg. Class.* 10, 61–74.
- Ren, J., and Kezunovic, M. (2010). “Use of recursive wavelet transform for estimating power system frequency and phasors,” in *2010 IEEE PES transmission and distribution conference and exposition: Smart solutions for a changing world*, (New Orleans, LA: IEEE), 1–6. doi: 10.1109/TDC.2010.5484474
- Rieder, P., Gotze, J., Nosse, J. S., and Burrus, C. S. (1998). Parameterization of orthogonal wavelet transforms and their implementation. *IEEE Trans. Circ. Syst. II Anal. Digit. Signal Proc.* 45, 217–226. doi: 10.1109/82.661654
- Ross, B. C. (2014). Mutual information between discrete and continuous data sets. *PLoS One* 9:e87357. doi: 10.1371/journal.pone.0087357
- Saltelli, A. (2002). Making best use of model evaluations to compute sensitivity indices. *Comput. Phys. Commun.* 145, 280–297. doi: 10.1016/S0010-4655(02)00280-1
- Sand, D., Rappel, P., Marmor, O., Bick, A. S., Arkadir, D., Lu, B., et al. (2021). Machine learning-based personalized subthalamic biomarkers predict ON-OFF levodopa states in Parkinson patients. *J. Neural Eng.* 18:abfc1d. doi: 10.1088/1741-2552/abfc1d
- Seabold, S., and Perktold, J. (2010). “Statsmodels: Econometric and statistical modeling with python,” in *Proceedings Of the 9th Python in Science Conference*, (Austin, TX), 92–96.
- Shah, S. A., Tan, H., and Brown, P. (2016). “Decoding force from deep brain electrodes in Parkinsonian patients,” in *Proceedings of the annual international conference of the IEEE engineering in medicine and biology society, EMBS*, (Piscataway, NJ: IEEE), 5717–5720. doi: 10.1109/EMBC.2016.7592025
- Shah, S. A., Tan, H., Tinkhauser, G., and Brown, P. (2018). Towards real-time, continuous decoding of gripping force from deep brain local field potentials. *IEEE Trans. Neural Syst. Rehabil. Eng.* 26, 1460–1468. doi: 10.1109/TNSRE.2018.2837500
- Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. (2016). Taking the human out of the loop: A review of Bayesian optimization. *Proc. IEEE* 104, 148–175. doi: 10.1109/JPROC.2015.2494218
- Sobol, I. M. (2001). Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Math. Comput. Simul.* 55, 271–280. doi: 10.1016/S0378-4754(00)00270-6
- Tan, H., Debarros, J., He, S., Pogoyan, A., Aziz, T. Z., Huang, Y., et al. (2019). Decoding voluntary movements and postural tremor based on thalamic LFPs as a basis for closed-loop stimulation for essential tremor. *Brain Stimul.* 12, 858–867. doi: 10.1016/j.brs.2019.02.011
- Tan, H., Fischer, P., Shah, S., Vidaurre, D., Woolrich, M., and Brown, P. (2018). “Decoding movement states in stepping cycles based on subthalamic LFPs in Parkinsonian patients,” in *Proceedings of the annual international conference of the IEEE engineering in medicine and biology society, EMBS*, (Piscataway, NJ: IEEE), 1384–1387. doi: 10.1109/EMBC.2018.8512545
- Tan, H., Pogoyan, A., Anzak, A., Ashkan, K., Bogdanovic, M., Green, A. L., et al. (2013). Complementary roles of different oscillatory activities in the subthalamic nucleus in coding motor effort in Parkinsonism. *Exp. Neurol.* 248, 187–195. doi: 10.1016/j.expneurol.2013.06.010
- Tan, H., Pogoyan, A., Ashkan, K., Green, A. L., Aziz, T., Foltyniec, T., et al. (2016). Decoding gripping force based on local field potentials recorded from subthalamic nucleus in humans. *eLife* 5, 484–499. doi: 10.7554/eLife.19089
- Tarvainen, M. P., Hiltunen, J. K., Ranta-aho, P. O., and Karjalainen, P. A. (2004). Estimation of nonstationary EEG with kalman smoother approach: An application to event-related synchronization (ERS). *IEEE Trans. Biomed. Eng.* 51, 516–524. doi: 10.1109/TBME.2003.821029
- Tomek, I. (1976). An experiment with the nearest-neighbor rule. *IEEE Trans. Syst. Man Cybernet.* 6, 448–452.

- Vaidyanathan, P. (1993). "Two Channel FIR Paraunitary QMF Banks," in *Multirate Systems and Filter Banks*, 1st Edn, ed. A. V. Oppenheim (Hoboken, NJ: Prentice Hall), 298–314.
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., et al. (2019). SciPy 1.0—fundamental algorithms for scientific computing in Python. *Nat. Methods* 17, 261–272. doi: 10.1038/s41592-019-0686-2
- Wang, T., Shoaran, M., and Emami, A. (2018). "Towards adaptive deep brain stimulation: LFP-feature analysis and classification," in *IEEE International conference on acoustics, speech and signal processing (ICASSP)*, (Piscataway, NJ: IEEE), 2536–2540.
- Waskom, M. (2021). seaborn: Statistical data visualization. *J. Open Source Softw.* 6:3021. doi: 10.21105/joss.03021
- Yao, L., Brown, P., and Shoaran, M. (2020). Improved detection of Parkinsonian resting tremor with feature engineering and Kalman filtering. *Clin. Neurophysiol.* 131, 274–284. doi: 10.1016/j.clinph.2019.09.021
- Zaker, N., Zhang, J. J., Hanrahan, S., Nedrud, J., and Hebb, A. O. (2016). "A new approach for automated detection of behavioral task onset for patients with Parkinson's disease using subthalamic nucleus local field potentials," in *Conference record - asilomar conference on signals, systems and computers*, (Piscataway, NJ: IEEE), 780–784. doi: 10.1109/ACSSC.2015.7421240
- Zhang, S., Connolly, A. T., Madden, L. R., Vitek, J. L., and Johnson, M. D. (2018). High-resolution local field potentials measured with deep brain stimulation arrays. *J. Neural Eng.* 15:046019. doi: 10.1088/1741-2552/aabdf5
- Zhang, S., Green, A., and Smith, P. P. (2013). "An automatic classifier of pain scores in chronic pain patients from local field potentials recordings," in *International IEEE/EMBS conference on neural engineering, NER*, (San Diego, CA: IEEE), 1194–1197. doi: 10.1109/NER.2013.6696153
- Zhu, B., Farivar, M., and Shoaran, M. (2020). ResOT: Resource-efficient oblique trees for neural signal classification. *IEEE Trans. Biomed. Circ. Syst.* 14, 692–704. doi: 10.1109/TBCAS.2020.3004544