Supplementary Material

The Portil∞p: a deep learning-based open science tool for closed-loop brain stimulation

Nicolas Valenchon, Yann Bouteiller, Hugo R. Jourde, Emily B.J. Coffey and Giovanni Beltrame

APPENDIX A
MODEL-BASED EXPLANATION OF SLEEP SPINDLES

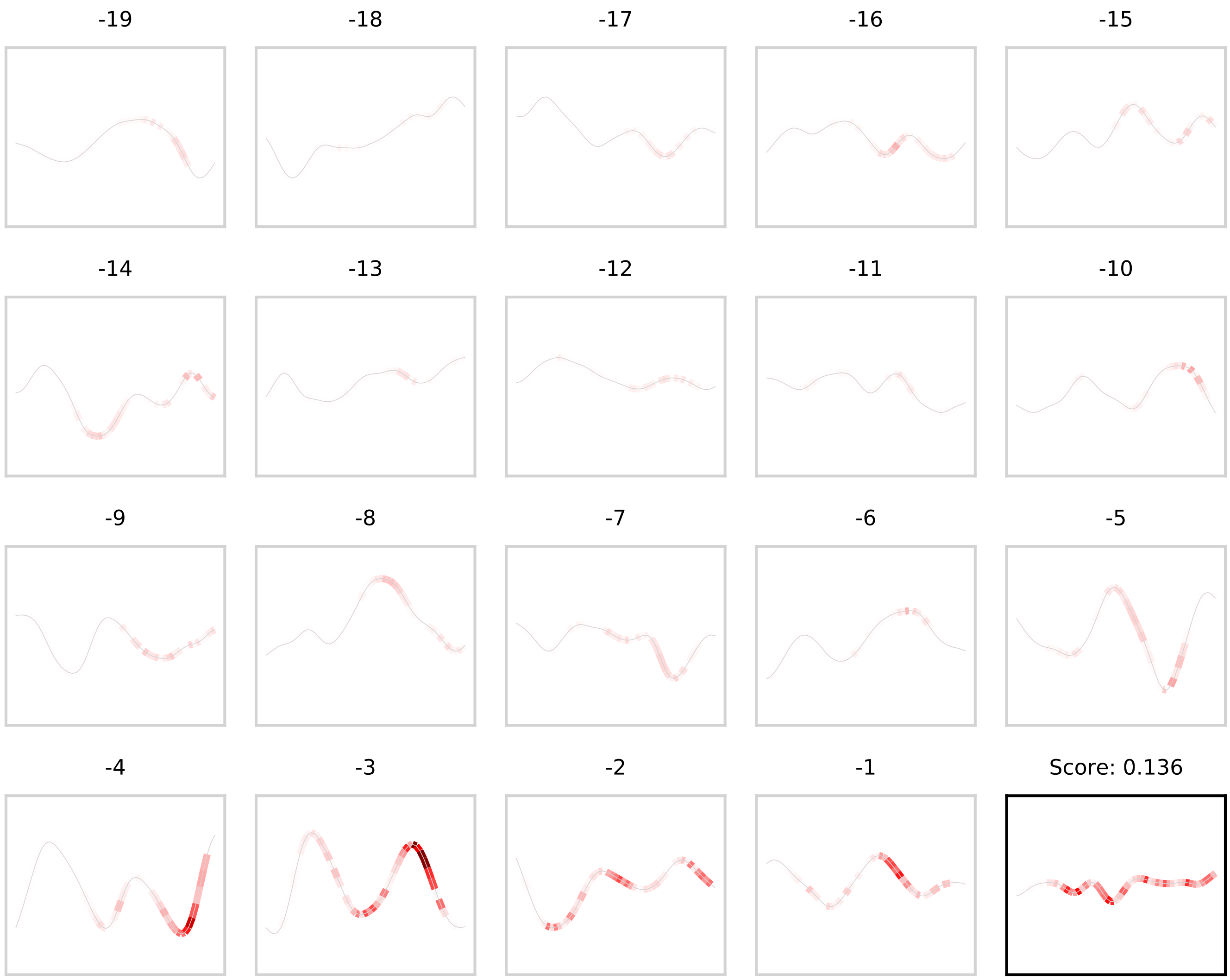

Fig. 1. Integrated gradients (classifier ANN output: 0.136). The *integrated gradients* algorithm enables exploring why the model takes a given decision (the more a portion of the signal is represented in red, the highest its influence on the current output of the ANN). Grey windows are past inputs, whereas the black window is the current input: the past influences the current output due to the RNN. Here, the model finds that it is looking at the aftermath of a sleep spindle. With our time dilation and window size, a small portion of the window overlaps from one sample to the next. We see that this portion (at the left-hand side of each window) is in fact ignored by the model. Therefore, it is probably possible to shrink our model even more, although PMBO did not find this. In the future, this type of visualizations might also help experts better understand what sleep spindles are by revealing unknown influences.

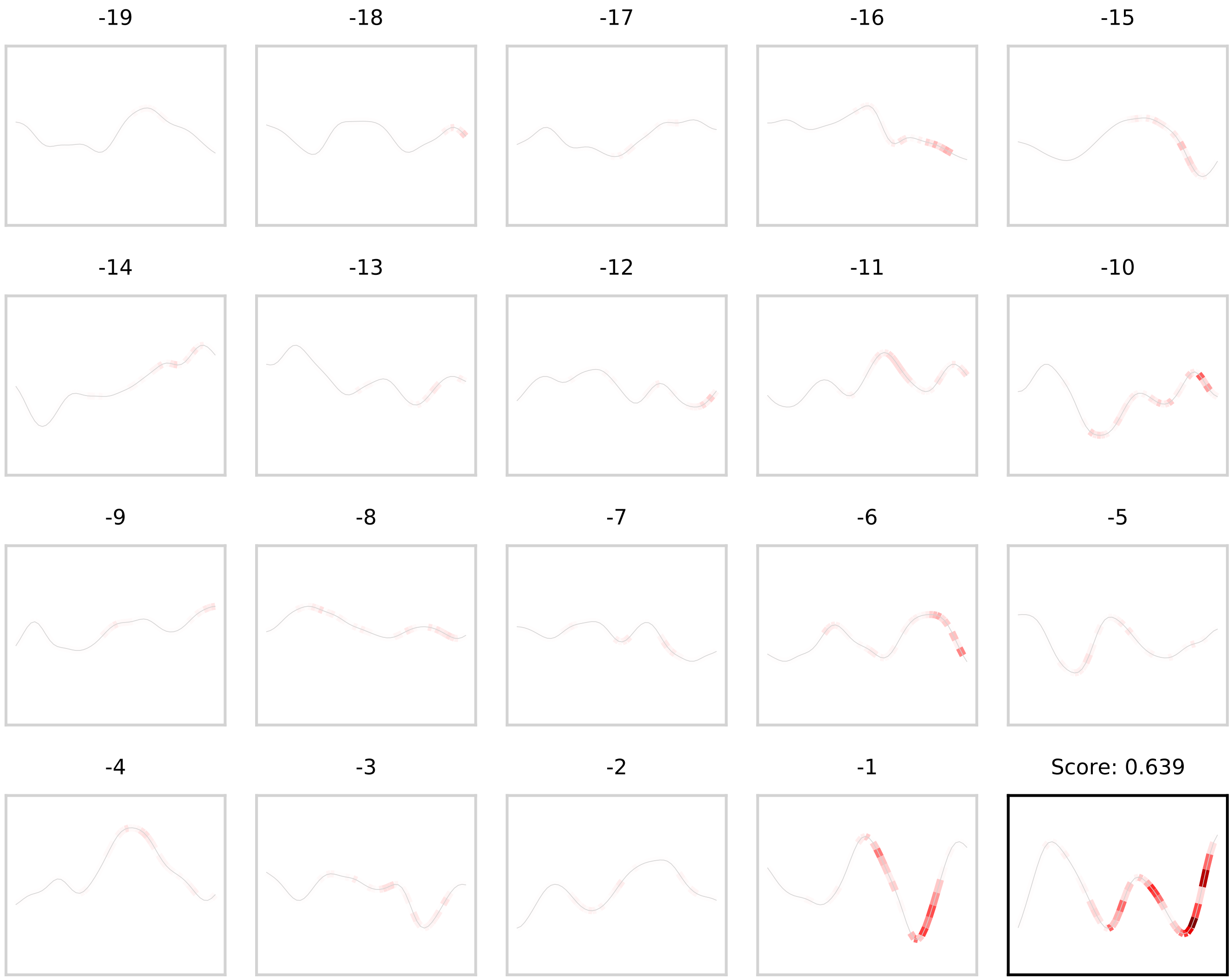

Fig. 2. Integrated gradients (classifier ANN output: 0.639). The current window is within an actual sleep spindle. The model mainly focuses on the spindle itself, but also a few events that happened further back in time, to make its decision.

APPENDIX B
STIMULATION VISUALIZATIONS

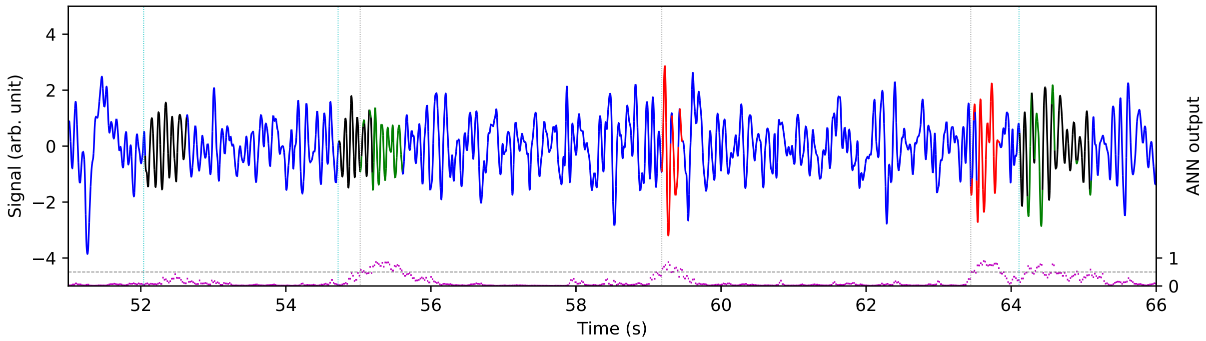

Fig. 3. Different stimulation failure modes (classifier with threshold 0.5). Blue: no sleep spindle and no detection. Black: sleep spindle not detected. Green: sleep spindle correctly detected. Red: detection where the signal is not a spindle. Vertical blue: beginning of a spindle. Vertical grey: stimulation. Magenta: ANN output. Horizontal gey: detection threshold. This figure illustrates typical ‘failure’ cases of our final sleep spindle stimulating device. False negative: the first spindle is missed because the threshold is too big. True positive: the second spindle is correctly stimulated. False positive: a part of the signal not labeled as a sleep spindle by MODA is detected as a spindle by the device and stimulated. Almost true positive: the sleep spindle is stimulated in advance (NB: we count this case as a false positive).

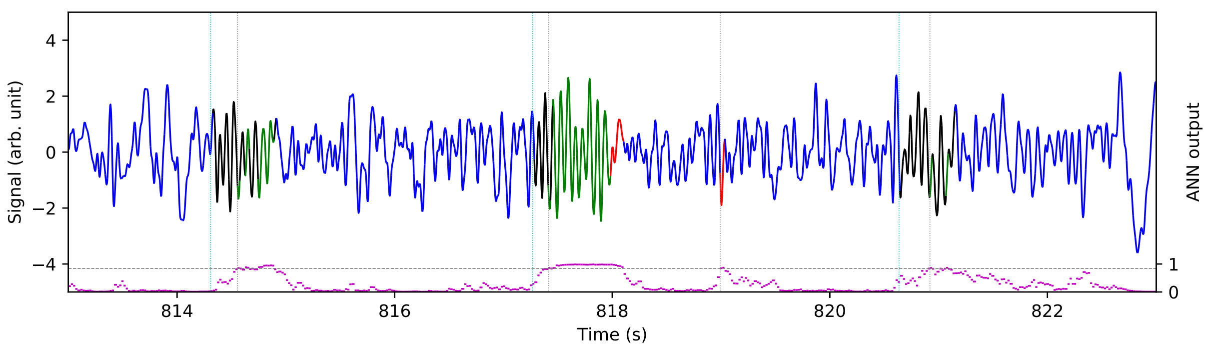

Fig. 4. Classifier with threshold 0.84, success example. Same color code as Figure 3. Increasing the detection threshold from 0.5 to 0.84 removes most false positive stimuli (vertical grey not following vertical blue).

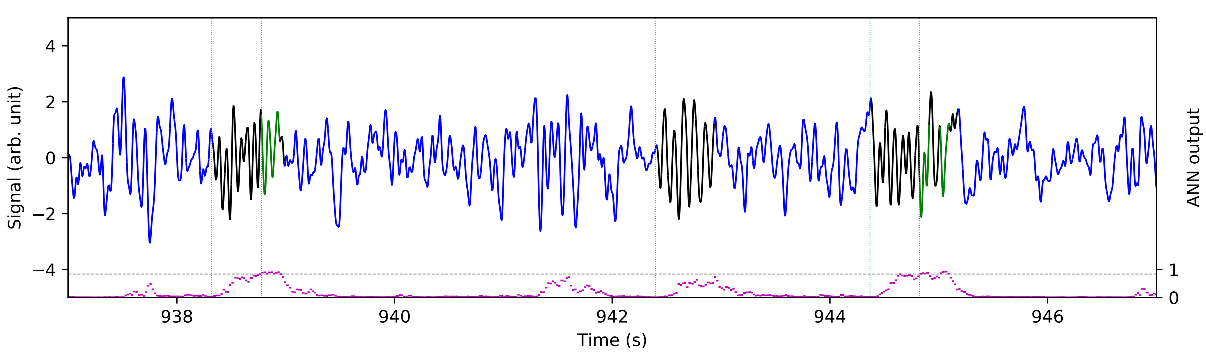

Fig. 5. Classifier with threshold 0.84, failure example. Same color code as Figure 3. Increasing the detection threshold comes with more false negative stimuli (vertical blue not followed by vertical grey).

Appendix C
PMBO hyperparameters

Table I. Hyperparameters used for PMBO

hyperparameter	selected value
range cost hardware	1000-80000
noise type 1	0,25
noise type 2	0,1
m	200
meta network type	MLP
# layers meta network	3
hidden size meta network	200
optimizer meta network	SGD
learning rate meta network	0.05
weight decay meta network	0.01

Two types of noise are used in our implementation of PMBO to foster exploration of the hyperparameter space:

- noise type 1: portion of the m sampled models that are sampled randomly in the whole hyperparameter space, instead of in a Gaussian around the last completed experiment.
- noise type 2: portion of the time when a model is sampled randomly in the m models, instead of being selected by its Pareto efficiency.

The ANN used for our meta model is a simple Multi-Layer Perceptron (MLP) of 3 fully connected layers. The hyperparameters we use in PMBO are summarized in Table I.

Appendix D
Model hyperparameters

Table II. Hyperparameters used to train the final model

Hyperparameter	Selected value	PMBO
Training		
optimizer	AdamW	
# epochs max	150	
epochs before early stopping	20	
early stopping running average factor	0,1	
batches per epoch	1000	
batch size	256	X
dropout on first layer	False	
dropout factor	0,5	
adam learning rate	0,0005	X
adam weight decay	0,01	
balancing mode	oversampling	
type of training	classification	
sequence length	50	
Architecture		
window size (s)	0,216	X
time dilation (s)	0,168	X
# CNN layers	3	X
# CNN channels	31	X
stride convolution	1	X
kernel size convolution	7	X
dilation convolution	1	X
stride max pooling	1	X
kernel size max pooling	1	X
dilation max pooling	1	X
RNN layers	1	X
RNN hidden size	7	X

The hyperparameters we use in our final model are listed in Table II. Hyperparameters that were chosen by PMBO are marked with a cross under the PMBO column.

Appendix E
Best threshold for classifiers and regressors

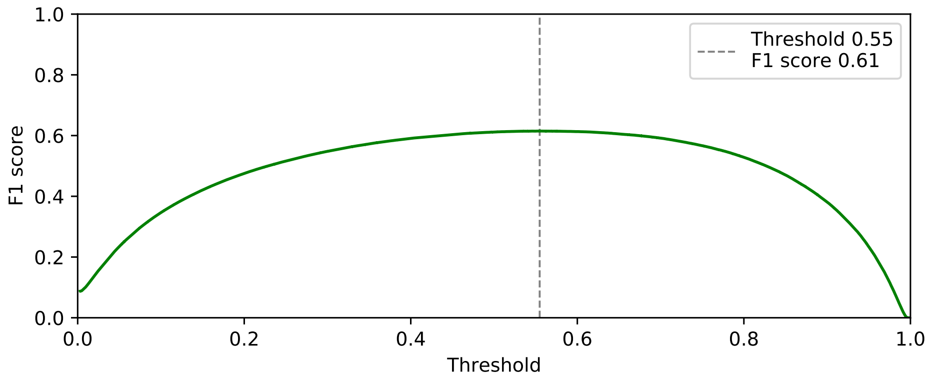

Fig. 6. F1 score evolution with threshold variation on classification

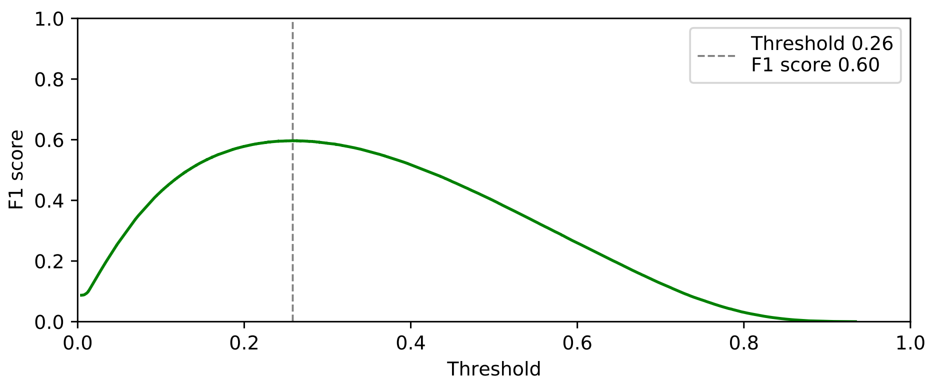

Fig. 7. F1 score evolution with threshold variation on regression

Appendix F
Detection delay distributions

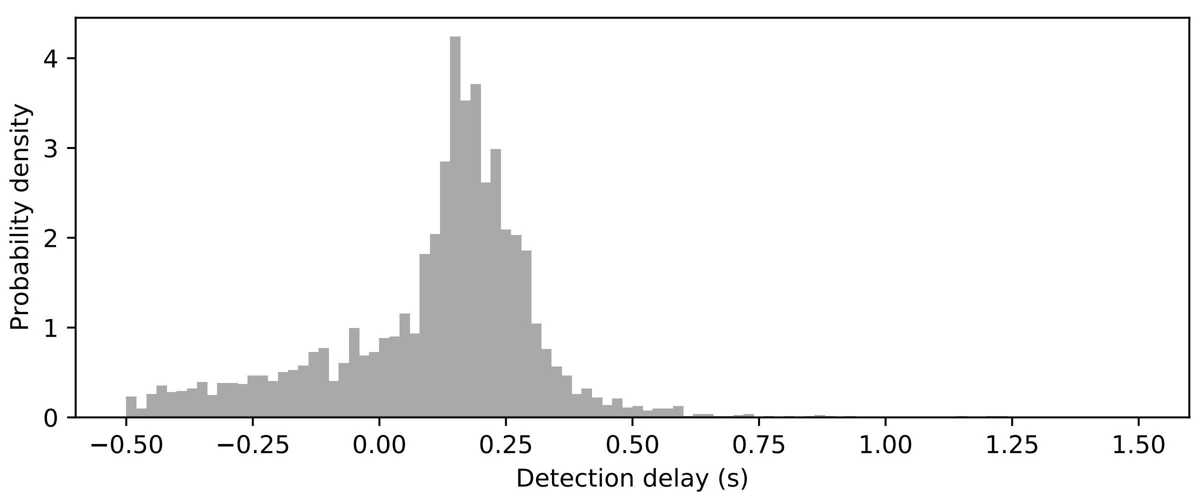

Fig. 8. Stimulation histogram for 1 input network classification with a 0.25 threshold

APPENDIX G
2-INPUT NETWORK ARCHITECTURE

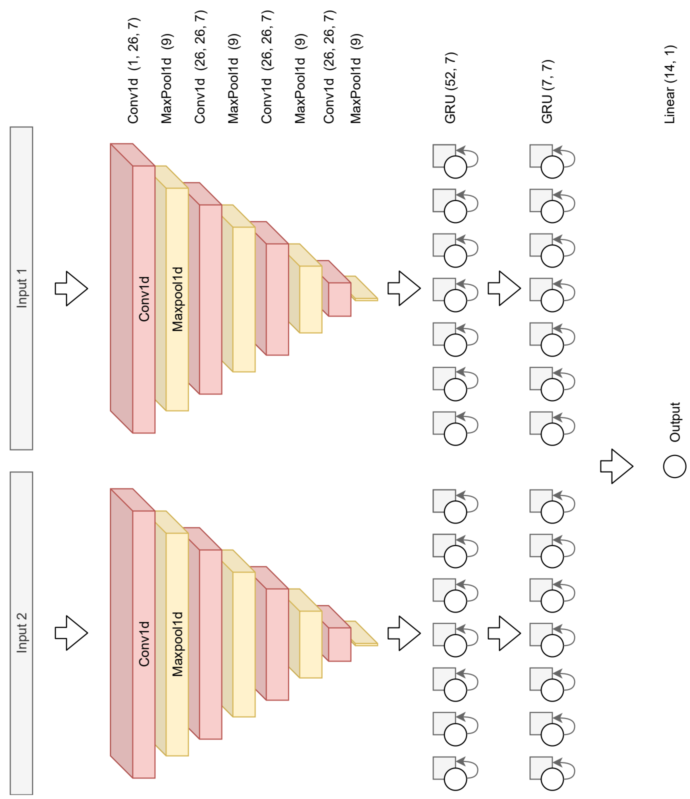

Fig. 9. 2-input neural network architecture. This architecture consists of two sequences of CNN and RNN. The first sequence processes the cleaned raw signal (input 1), whereas the second processes the envelope (input 2). The latent features from both branches are concatenated and fed to a fully connected layer to yield the output of the ANN.