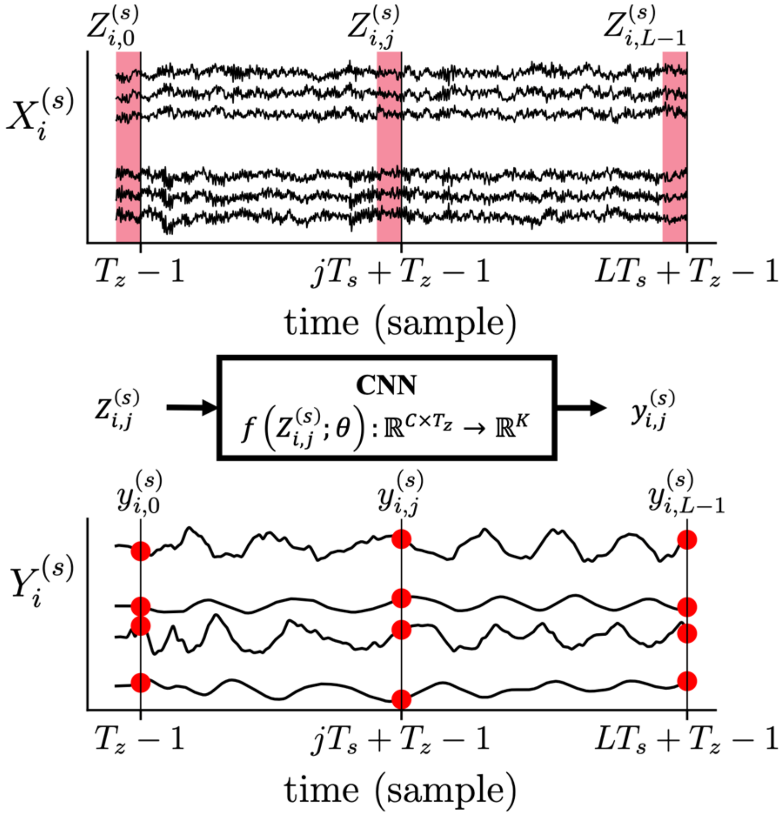
Supplementary Materials: Decoding movement kinematics from EEG using an interpretable convolutional neural network

Davide Borra (0000−0003−3791−8555), Valeria Mondini (0000-0001-7680-6199), Elisa Magosso (0000−0002−4673−2974), Gernot R. Müller-Putz (0000-0002-0087-3720)

Supplementary Section 1: Continuous trajectory decoding from the EEG

To perform continuous decoding of 2-D positions and velocities, the CNN input was a buffer of EEG data (‘EEG chunk’), consisting of T_z time samples of the multi-variate input time series recorded during each trial, extracted with a stride of T_s samples. This procedure is reported in the Supplementary Fig. 1 and described in Sections 2.1 and 2.2 of the manuscript.



Supplementary Figure 1 – Schematic of the extraction of EEG chunks and trajectory values from EEG trials. Each red shaded area (top) denotes the buffer of EEG data forming the EEG chunk, and its four associated red dots (bottom) denote the kinematic values.

Supplementary Section 2: Hyper-parameter search and hyper-parameter sensitivity analysis

To select the optimal hyper-parameters defining the architecture and to train the architecture, Bayesian optimization [1] was performed while training within-subject models. The searched hyper-parameters were ${F_0}^{ISS}$, ${K_0}^{ISS}$, ${D_1}^{ISS}$, the number of parallel branches each learning features at a different time scale (N_s), ${K_1}^{DST}$, use of batch normalization [2], p, and the learning rate. The meaning of these symbols is described in the Sections 2.2 and 2.3 of the manuscript. Batch normalization [2] normalizes the network intermediate outputs, speeding up training, reducing the influence of a specific parameter initialization scheme, and introducing a regularizer effect. CNNs widely exploit this normalization for motor classification problems (e.g., [3–6]) from the EEG. To evaluate whether this technique could be useful also for trajectory decoding, batch normalization was applied to the output of each convolutional layer before the activation function, as suggested in [2]. The following procedure was performed to select the kernel sizes of the multi-scale temporal feature extractor. At first, the kernel size of the largest scale was set the same as the searched value of ${F_0}^{ISS}$, denoting the temporal kernel size of the first layer. Then, depending on the number of scales selected, the kernel size of the i-th parallel branch ($1 \leq i \leq N_s$) is defined automatically as $(1, {F_0}^{ISS}[1]/i)$, $1 \leq i \leq N_s$, taking the nearest odd number, e.g., when ${F_0}^{ISS} = (1{,}51)$, then kernel sizes of the multiple scales are set to $(1{,}51)$ and $(1{,}25)$ if $N_s = 2$, or $(1{,}51)$, $(1{,}25)$, $(1{,}17)$ if $N_s = 3$. Bayesian optimization was performed for 100 iterations using the validation loss as metric to minimize, tree-structured Parzen estimator as surrogate function and expected improvement as selection function. The optimal configuration of the ICNN was selected as the most frequent one across the subject-specific Bayesian-optimized models and corresponds to the one described in Table 1 of the manuscript. Details on the source space are reported in Supplementary Table 1.

Supplementary Table 1 – Searched hyper-parameters of MS-Sinc-ShallowNet: distributions and admitted values sampled during Bayesian optimization. Curly brackets denote discrete admitted values, while square brackets denote interval of admitted values.

Hyper-parameter	**Distribution**	**Values**
${F_0}^{ISS}$	uniform	{(1,25), (1,51)}
${K_0}^{ISS}$	uniform	{8,16,32,64}
${D_1}^{ISS}$	uniform	{1,2,4}
N_s	uniform	{1,2,3}
${K_1}^{DST}$	uniform	$\{{K_0}^{ISS} \cdot {D_1}^{ISS}$, 1, 2, 4}
Use of batch norm.	uniform	{False, True}
p	uniform	{0, 0.25, 0.5}
Learning rate	log-uniform	[1e-5, 1e-2]

Then, the main hyper-parameters of the proposed ICNN were investigated by changing one hyper-parameter at a time and evaluating the change in the performance, i.e., performing a sensitivity analysis on the hyper-parameters, as performed to study hyper-parameters in [3,5,7]. Hereafter, the ICNN with the baseline hyperparameters defined in Table 1 of the manuscript, will be referred as ‘baseline’ architecture. Then, we changed the value of one hyper-parameter at a time of the baseline architecture, realizing a ‘variant’ architecture. Both architectural (i.e., parameters affecting the overall architecture design) and training hyper-parameters were investigated (i.e., parameters influencing the training). These were:

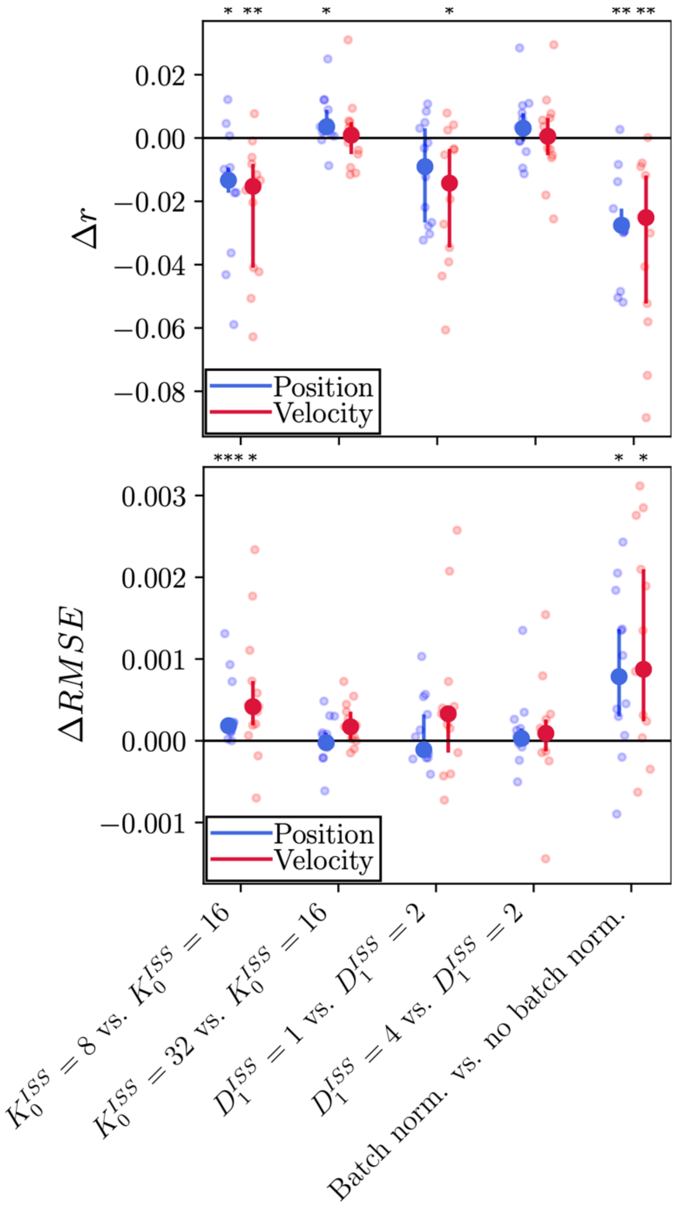
i. The number of trainable band-pass filters in Block 1 (${K_0}^{ISS}$). In the baseline architecture, 16 band-pass filters were learned. To investigate designs using less or more filters, two variant architectures were developed, by setting ${K_0}^{ISS} = 8$ or ${K_0}^{ISS} = 32$. Of course, using less or more filters is associated to a reduced or increased model size, respectively.

ii. The number of spatial filters tied to each band-pass filter in Block 1 (${D_1}^{ISS}$). In the baseline architecture, 2 spatial filters were learned for each band-pass filter. To investigate designs using less or more spatial filters, two variant architectures were developed, by setting ${D_1}^{ISS} = 1$ or ${D_1}^{ISS} = 4$. Of course, using less or more filters is associated to a reduced or increased model size, respectively.

iii. The inclusion of batch normalization [2]. In the baseline architecture, batch normalization was not adopted. To evaluate whether this technique could be useful also for trajectory decoding, batch normalization was included in the architecture, as specified at the beginning of Supplementary Section 2.

Performance metrics were computed for these variant conditions as described in Section 2.6 of the manuscript. The performance difference between each of the previously described variants and the baseline architecture was computed. Then, for brevity the performance difference was averaged across x- and y-axis components, and the effect of each hyper-parameter was studied for position and velocity, separately. The performance of the baseline MS-Sinc-ShallowNet was compared with the performance obtained with each variant MS-Sinc-ShallowNet design with pairwise comparisons (10 total tests). Pairwise comparisons were performed using Wilcoxon signed-rank tests and false discovery rate correction at $\alpha = 0.05$ (Benjamini–Hochberg [8]) to correct for multiple tests.

Correlation and RMSE differences between variant ICNNs and the baseline ICNN are reported in Supplementary Fig. 2, together with the results of the statistical analysis.

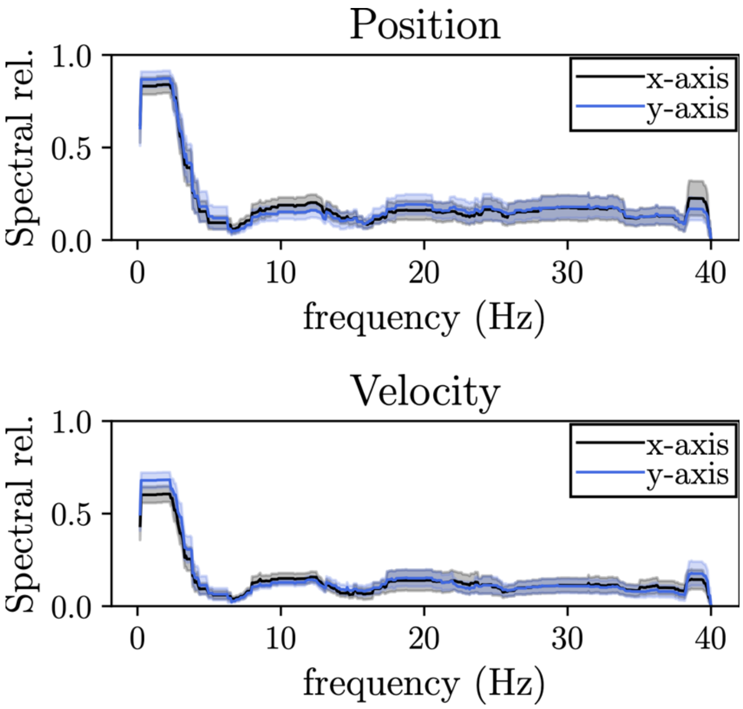


Supplementary Figure 2 – Sensitivity analysis on the main ICNN hyper-parameters. Difference of the Pearson's correlation coefficients (Δr) and of the RMSE ($\Delta RMSE$) between each MS-Sinc-ShallowNet variant and the baseline MS-Sinc-ShallowNet, both trained in within-subject (WS) strategy. Smaller dots represent the performance difference for each subject, while bigger dots represent the median of each distribution and whiskers represent the 25th and 75th percentile. Significant corrected (Benjamini–Hochberg [8]) p-values are reported (*p<0.05, **p<0.01, ***p<0.001).

In addition to significant higher RMSE ($p < 0.05$), significantly lower correlations were obtained when using a lower number of band-pass filters in Block 1 ($p < 0.05$), and when including batch normalization ($p < 0.01$), for both position and velocity. Lower correlations ($p < 0.05$) were also found when using less spatial filters in Block 1, but with a significant worsening in performance only for velocity component. Using more filters in Block 1 (both band-pass and spatial filters) only slightly increased the performance, with a small but significant ($p < 0.05$) improvement for position correlations.

Supplementary Section 3: Spectral relevance comparison across directions

In this study, the spectral relevance was computed for each decoded trajectory, i.e., p_x, p_y, v_x, v_y, see Eq. B.4, and is reported in Supplementary Fig. 3 for each coordinate. A permutation cluster test with 5000 permutations and by using threshold-free cluster enhancement [9] was performed between the relevance values in the x- and y-axes, separately for position and velocity. This was done to test whether there are significant differences between the two different directions. No significant differences were obtained, highlighting that the spectral relevance between the two directions was comparable.



Supplementary Figure 3 – Spectral relevance for position and velocity. The spectral relevance is reported in its mean value (tick line) and standard error of the mean (shaded area) across subjects.

Supplementary references

[1] J. Snoek, H. Larochelle, R.P. Adams, Practical Bayesian Optimization of Machine Learning Algorithms, in: F. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 25, Curran Associates, Inc., 2012: pp. 2951–2959. http://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf.

[2] S. Ioffe, C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, in: F. Bach, D. Blei (Eds.), Proceedings of the 32nd International Conference on Machine Learning, PMLR, Lille, France, 2015: pp. 448–456.

[3] R.T. Schirrmeister, J.T. Springenberg, L.D.J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, T. Ball, Deep learning with convolutional neural networks for EEG decoding and visualization, Human Brain Mapping. 38 (2017) 5391–5420. https://doi.org/10.1002/hbm.23730.

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

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

[6] D. Borra, S. Fantozzi, E. Magosso, EEG Motor Execution Decoding via Interpretable Sinc-Convolutional Neural Networks, in: J. Henriques, N. Neves, P. de Carvalho (Eds.), XV Mediterranean Conference on Medical and Biological Engineering and Computing – MEDICON 2019, Springer International Publishing, Cham, 2020: pp. 1113–1122. https://doi.org/10.1007/978-3-030-31635-8_135.

[7] D. Borra, S. Fantozzi, E. Magosso, A Lightweight Multi-Scale Convolutional Neural Network for P300 Decoding: Analysis of Training Strategies and Uncovering of Network Decision, Frontiers in Human Neuroscience. 15 (2021) 655840. https://doi.org/10.3389/fnhum.2021.655840.

[8] Y. Benjamini, Y. Hochberg, Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing, Journal of the Royal Statistical Society. Series B (Methodological). 57 (1995) 289–300.

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