nature biomedical engineering

Article

https://doi.org/10.1038/s41551-024-01297-1

Enhanced control of a brain–computer interface by tetraplegic participants via neural-network-mediated feature extraction

Received: 10 January 2023

Accepted: 27 October 2024

Published online: 06 December 2024

Check for updates

Benyamin Haghi[1,8] ✉, Tyson Aflalo[2,8] ✉, Spencer Kellis[2,3,4], Charles Guan[2], Jorge A. Gamez de Leon[2], Albert Yan Huang[1], Nader Pouratian[5,6], Richard A. Andersen[2] & Azita Emami[1,7] ✉

To infer intent, brain–computer interfaces must extract features that accurately estimate neural activity. However, the degradation of signal quality over time hinders the use of feature-engineering techniques to recover functional information. By using neural data recorded from electrode arrays implanted in the cortices of three human participants, here we show that a convolutional neural network can be used to map electrical signals to neural features by jointly optimizing feature extraction and decoding under the constraint that all the electrodes must use the same neural-network parameters. In all three participants, the neural network led to offline and online performance improvements in a cursor-control task across all metrics, outperforming the rate of threshold crossings and wavelet decomposition of the broadband neural data (among other feature-extraction techniques). We also show that the trained neural network can be used without modification for new datasets, brain areas and participants.

Brain–computer interfaces (BCIs)—technologies that communicate directly with the brain—can improve the quality of life of millions of participants with brain circuit disorders[1]. Motor BCIs are among the most powerful examples of BCI technology: ongoing clinical trials implant microelectrode arrays into motor regions of tetraplegic participants. Movement intentions are decoded from recorded neural signals into command signals to control a computer cursor or a robotic limb[2–7]. However, these systems fail to deliver the precision, speed, degrees of freedom and robustness of control enjoyed by motor-intact individuals. To enhance the overall performance of the BCI systems and to extend the lifetime of the implants, newer approaches for recovering functional information of the brain are necessary.

Part of the difficulty of improving BCI control is the unconstrained nature of the design problem. Such design can be fundamentally modelled as a data science problem: the mapping from brain activity to motor commands must be learned from data[5,8,9] and must find adequate solutions to the unique challenges of neural interfaces, such as limited and costly training data, low signal-to-noise ratio (SNR) predictive features, complex temporal dynamics, nonlinear tuning curves, neural instabilities and the fact that solutions must be optimized for usability, not offline prediction[10–18]. These properties have made end-to-end solutions (for example, mapping 30 kHz sampled array recordings to labelled intention data) intractable. Therefore, most BCI systems separate the decoding problem into two distinct phases: (1) transforming

[1]Department of Electrical Engineering, California Institute of Technology, Pasadena, CA, USA. [2]Division of Biology and Biological Engineering, California Institute of Technology, Pasadena, CA, USA. [3]Neurostimulation Center and Neurosurgery, Keck School of Medicine, University of Southern California, Los Angeles, CA, USA. [4]Blackrock Microsystems, Salt Lake City, UT, USA. [5]UT Southern Medical Center, Dallas, TX, USA. [6]UCLA Neurosurgery, University of California Los Angeles, Los Angeles, CA, USA. [7]Andrew and Peggy Cherng Department of Medical Engineering, California Institute of Technology, Pasadena, CA, USA. [8]These authors contributed equally: Benyamin Haghi, Tyson Aflalo. ✉e-mail: benyamin.a.haghi@caltech.edu; taflalo@caltech.edu; azita@caltech.edu

https://doi.org/10.1038/s41551-024-01297-1

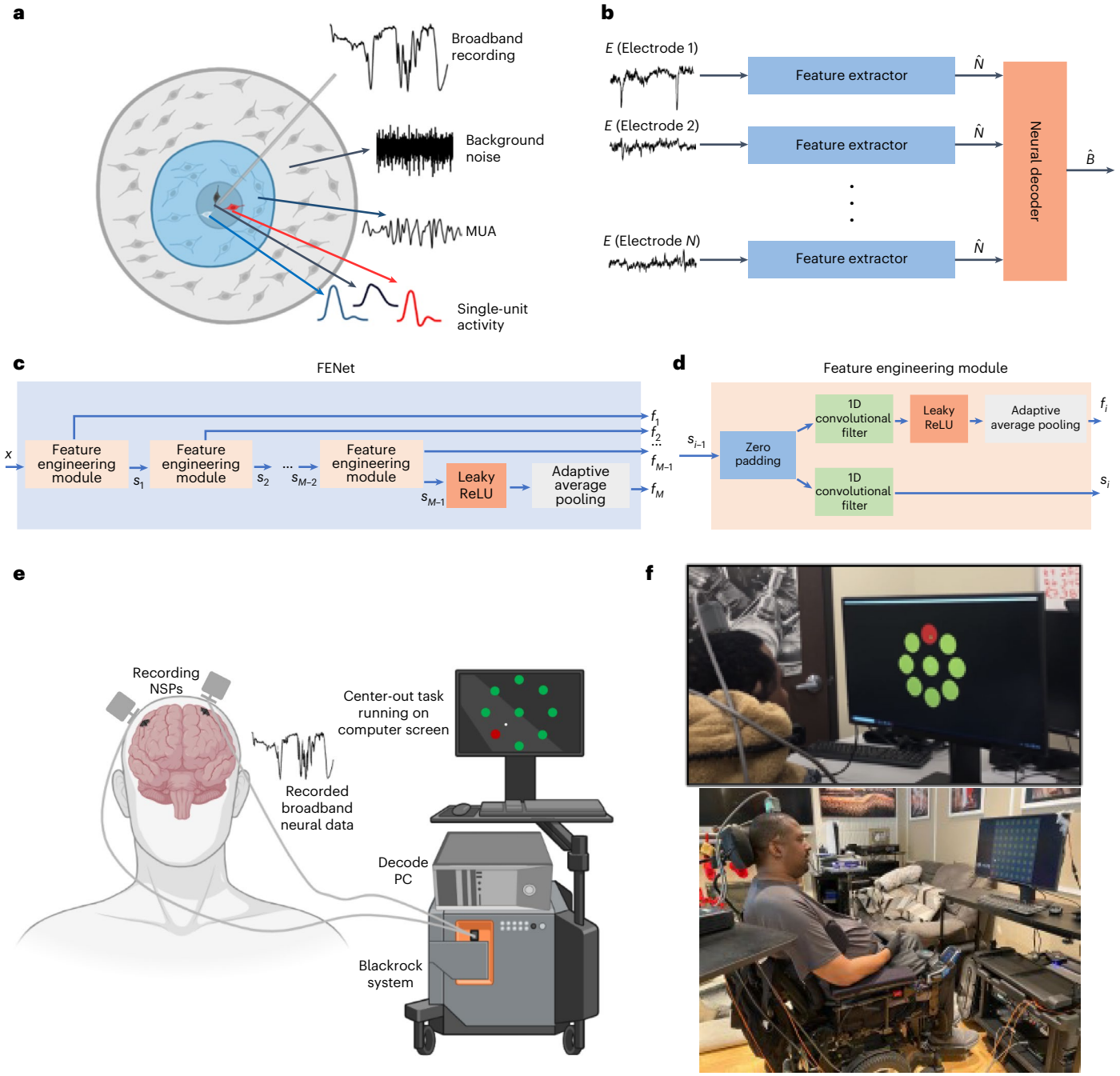


Fig. 1 | Overview of the study's methodology. a, Each single electrode records the broadband data that consist of various neural activities (for example, somata, dendrites, axons and so on). Neurons close to the electrode will generate stronger single-unit activities compared with the neurons far from the recording electrode, which record MUAs. The electrode records noise as the distance of the neurons increases. **b**, Schematic architecture enabling FENet training illustrating separate processes for feature generation ($\boldsymbol{E}\rightarrow\hat{\boldsymbol{N}}$) and neural decoding ($\hat{\boldsymbol{N}}\rightarrow\hat{\boldsymbol{B}}$). In blue, feature extractors estimate neural activity $\hat{\boldsymbol{N}}$ from recorded electrical activity E. This system has fixed parameters for all the electrodes, tasks, sessions and participants. In orange, a neural decoder estimates the behaviour $\hat{\boldsymbol{B}}$ from estimates of the neural activity $\hat{\boldsymbol{N}}$. The session-specific neural decoder is learned for each session. **c**, FENet implementation including $M-1$ back-to-back feature engineering modules, leaky ReLU and adaptive average pooling. **d**, A single FENet feature engineering module with zero padding, 1D convolutional filters, a leaky ReLU activation function and adaptive average pooling. **e**, The overall architecture of the closed-loop BCI system. First, the data are recorded from the Utah microelectrode arrays (NeuroPort, Blackrock Microsystems) implanted on the surface of the brain by using two neural signal processors (NSPs). Then, neural signal processors send the recorded neural data to the Blackrock microsystem to preprocess the raw data. After the pre-processing, the decode PC extracts the appropriate neural features and decodes the neural features to the computer cursor movements for the under-study task. **f**, A research participant controls a cursor in a BCI in centre-out task (top) on a unit circle and a grid task (bottom) which includes 64 targets on an 8-by-8 square. For each trial, a computer-generated target appears randomly in red.

electrical signals recorded from implanted electrode arrays into neural features and (2) learning parameters that map neural features to control signals. Despite the increasing number of decoding methodologies, including those incorporating neural networks as decoders, current BCI systems rely on conventional feature extraction approaches such as spike band powers, threshold crossings (TCs) and wavelets (WTs)[4,17,19–30]. However, most of these feature extraction techniques, including TCs and WTs, are likely suboptimal as they use simple heuristics or were

https://doi.org/10.1038/s41551-024-01297-1

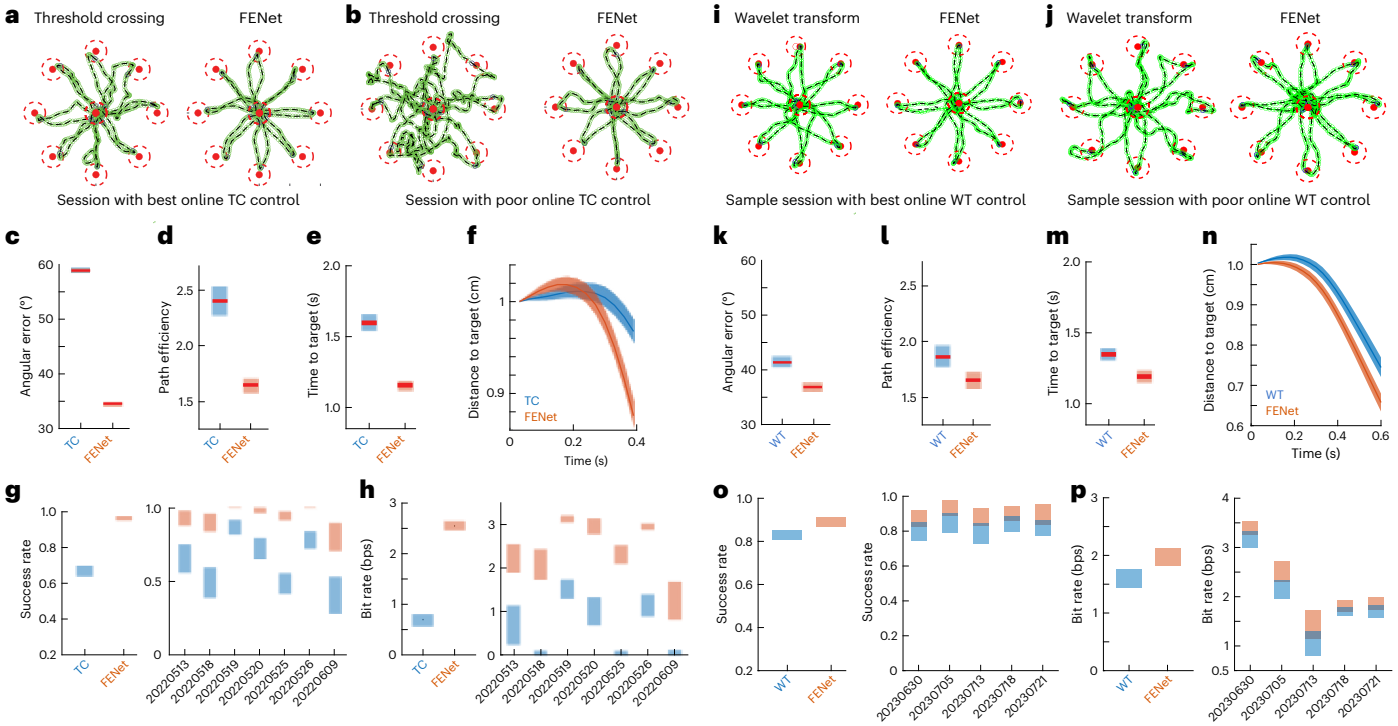


Fig. 2 | Closed-loop performance evaluation for JJ. Online trajectories comprising one movement out and back to each of eight targets in a centre-out paradigm. This figure illustrates the effectiveness of a linear decoder when it performs on FENet features, compared with the TC features (left panel) and the WT transform features (right panel). At the top, left and right figures show trajectories using the other feature extraction technique and FENet-based features, respectively. Trajectories were sampled from the same experimental run, as part of an interleaved-block design. There is high daily variability in control quality using TCs. In addition, due to the considerable instability encountered during the operation of WTs, we had to manually introduce bias in both the *x* and *y* directions to prevent the cursor from going beyond the screen after prolonged use. **a**,**i**, Results from the best recent day using TCs (**a**), WTs (**i**) and FENet. **b**,**j**, Results for another experimental session with poor TCs (**b**) and WTs (**j**) performances. TC control quality has degraded substantially, while FENet has largely preserved performance. Furthermore, the control achieved through the use of WTs has shown inconsistent stability between sessions 5 years after implantation. **c**,**d**,**e**,**k**,**l**,**m**, The averaged angular error (**c**,**k**), path efficiency (**d**,**l**) and time to target (**e**,**m**) over the closed-loop sessions as the closed-loop control metrics, comparing FENet with TC (**c**,**d**,**e**) and WT (**k**,**l**,**m**). Instantaneous angular error captures the angle between the vector pointing towards the target and the instantaneous velocity of the cursor. Path efficiency is measured as the total distance travelled en route to the target normalized by the straight-line distance from the starting location to the target. Distance to target (mean ± 95% confidence interval) was used to quantify cursor responsiveness to the participant's intent. Here, latency from target onset to goal-directed movements is shorter for FENet-based features compared with TCs and WTs. **f**,**n**, The averaged distance to target for the centre-out task, comparing FENet with TC (**f**) and WT (**n**). To account for variations in trial lengths, the figure depicts the average distance to the target across multiple trials, generated using the duration of the smallest trial as a reference. Consequently, the figure does not extend in time until the target has been reached. **g**,**h**,**o**,**p**, The success rate (**g**,**o**), and the bit rate (**h**,**p**) (Methods) within an 8-by-8 grid task (for FENet versus TCs (**g**,**h**), $t = -11.850$, $P < 0.0001$; for FENet versus WTs (**o**,**p**), $t = -4.252$, $P < 0.0001$). Success was measured as the ability to move the cursor to and hold a target (0.5 s hold time) within 4 s. bps, bits per second.

developed in other domains and simply applied to the neural signals. Therefore, these methods may perform sub-optimally compared with the data-driven methods that may better account for the specific biophysical processes giving rise to the dynamics of interest in the raw electrical recordings. The process of learning an optimal mapping from raw electrical recordings to neural features has not been explored.

In this Article, we report the development of an algorithm to optimize the information content of neural features and demonstrate improvements in human participants participating in intracortical BCI clinical trials. We designed our algorithm with several considerations: (1) the new method should easily drop into current decoding pipelines; (2) the method should generalize across electrodes, participants, brain areas and implant duration without parameterization; (3) the method should run real time on standard computers and ultimately be deployable in low-power application-specific integrated circuits; (4) the method should not substantially increase the complexity or amount of training data required for the subsequent decoding algorithm that maps the extracted neural features to the participant's intent. To fulfil these requirements, we developed FENet (for 'feature-extraction network'), a compact one-dimensional (1D) convolutional network that is specifically trained to extract the informative neural features from the broadband neural recordings for the BCI applications. The aim was to develop a constrained end-to-end training architecture. This architecture was structured to maximize the amount of information contained in the extracted neural features while abstracting away the parametric relationship between the extracted features and the decoded participant behaviour (Fig. 1b).

In addition, a retrospective analysis over years of recordings showed that FENet generates a higher-magnitude peak-to-trough within tuning curves and achieves improved trial separability compared with other feature extraction techniques over the entire lifetime of the array. FENet demonstrated a significant improvement in cross-validated coefficient of determination (R^2) compared with TCs, as the current standard feature extraction techniques in lab works with human participants[4,6,22,26,27,31], and WTs, which have also demonstrated performance improvements in our offline analysis and in the recent studies on BCIs[20,29], across multiple participants and through the lifetime of the arrays. Furthermore, FENet generalized well across cortical brain regions, participants and tasks, demonstrating its ability to serve as a drop-in replacement for other feature extraction techniques.

Finally, the population-level analysis demonstrated that FENet preserves the representational structure and temporal dynamics of sorted neural populations and, thus, provides an accurate measure of brain activity. Due to the inherent variability in absolute performance of BCI systems across participants, labs, tasks, and implant sites and age, we use within-participant comparisons to assess the efficacy of FENet. This approach aligns with current recommendations for evaluating BCI performance and underscores the importance of considering participant-specific factors when interpreting results[5,32–35]. Taken together, FENet can improve the efficacy of implantable electrode systems while delivering improved performance and ease of use.

Fluctuations in electrical activity recorded at an electrode come from a diversity of sources[36] (Fig. 1a). Typically, a neural decoding pipeline starts with extracting a particular neural feature of interest, which has historically been the number of neural spikes per unit of time. However, recent work has shown that alternative ways of processing broadband electrical recordings (for example, WT decompositions or power) can improve the information content of extracted features[19,20,29]. We hypothesized that a custom-tailored algorithm built around the statistics of neural signals may enable further improvements in extracting informative neural features. In general, the BCI problem can be formulated as learning a mapping from electrical fluctuations E to behaviour B. However, as mentioned above, the decoding problem is made tractable by splitting the problem into two stages: first mapping E to estimates of neural activity N and, then, from N to B (Fig. 1b). However, as we have no direct knowledge of N, we attempt to recover $\hat{N}$, the estimate of neural state that optimizes estimates of the behavioural state. To accomplish this, we adopt the constrained end-to-end architecture of Fig. 1b. We fix parameters mapping E to $\hat{N}$ across all electrodes and recording sessions, while allowing the mapping between the estimate of the neural activity $\hat{N}$ to behaviour $\hat{B}$ (for example, cursor velocity) to be electrode and session dependent. This approach assumes that the same transfer function can be applied to all electrodes and is independent of the relationship between the neural state and the behaviour. Sharing weights across electrodes reduces the number of parameters, improves interpretability and encourages solutions that generalize to new electrodes with distinct tuning properties. We designed FENet as a multi-layer 1D convolutional architecture for our feature extraction module (the mapping from E to $\hat{N}$; Fig. 1c,d and Extended Data Fig. 1b,c) while using a linear mapping that decodes the estimates of neural activity, $\hat{N}$, to the behaviour, $\hat{B}$. The use of a linear mapping was designed to encourage maximum learning within our feature extraction network.

We trained our feature extraction network (FENet) on data collected from electrode arrays implanted in motor and posterior parietal cortices of paralysed humans participating in the BCI clinical trials. Training data consisted of broadband recordings sampled at 30 kHz rate recorded while participants attempted movements in a centre-out task (Methods and Fig. 1e,f). The amount of training data, hyperparameters, the importance of FENet extracted features per electrode and the computational cost of FENet per evaluation for different FENet architectures were explored in Extended Data Figs. 6 and 7.

Results

FENet improves closed-loop control

We developed FENet to improve the closed-loop control of external devices. Figure 2 compares BCI-controlled cursor movements using FENet-based neural features, threshold-based neural waveform crossings (TCs) and WT transform for participant JJ (Methods). TCs represent the current standard for closed-loop control and is the method that underlies best-in-world closed-loop control performance[4,6,22,26,27,31,34]. WTs have also demonstrated performance improvements in recent studies on BCIs[20,29]. JJ was instructed to guide a BCI-controlled cursor towards visually cued targets on a computer screen. Testing was done in both a 'centre-out' environment, in which targets alternated between a central location and one of the eight pseudo-randomly chosen peripheral locations (Fig. 1e,f, top), and a 'grid' environment, in which the target was pseudo-randomly chosen from an 8-by-8 grid of targets (Fig. 1f, bottom). The data used to train the linear decoders mapping neural features to behaviour were collected either in an open-loop setting or using interleaved blocks of closed-loop data that included use of both FENet features and TCs/WTs to minimize the chances that training data would bias performance in favour of FENet or TCs/WTs. All experiments were double-blind using block-interleaved scheduling (see Extended Data Fig. 1a for a schematic illustrating the training and testing protocols).

Neural decoders using FENet-based features outperformed TC-based and WT-based features across all metrics. The difference in performance is visually evident when viewing the two approaches in our interleaved-block design (Supplementary Videos 1–4) or when visualizing the trajectories across movements (Fig. 2a,b,i,j). FENet-based features improved cursor trajectories as measured by reduced instantaneous angular error, improved path efficiency and reduced time to target (Fig. 2c,d,e,k,l,m). Furthermore, FENet improved the responsiveness of the cursor to the participant's intent, decreasing the latency between target onset and the time the cursor first moved towards the target (Fig. 2f,n). Improvements on both fronts resulted in substantial improvements in overall task performance during the grid task, including success rate and bit rate (for FENet versus TCs, $t = -11.850$, $P < 0.0001$, and for FENet versus WTs, $t = -4.252$, $P < 0.0001$) (Fig. 2g,h,o,p). Finally, as part of our double-blind experimental design, we asked the participant to report which of the two methods he preferred. In every instance, the participant reported a strong preference for the FENet-based decoder.

Baseline performance with TCs was poor during testing. This poor performance was the consequence of substantial degradation in the quality of the neural signals over the lifetime of the recording arrays[37] (Extended Data Fig. 2). Furthermore, the performance of WTs showed instability, necessitating manual adjustment of bias in both the x and y directions to enable participants to maintain stable control over the cursor and successfully complete the trials while using WTs. As shown in the next section, we find that FENet can improve the performance across the lifetime of the array (even when TCs and WTs produce excellent performance) and across the participants.

FENet provides improved open-loop decoding performance

Direct comparison in closed-loop testing is ideal, but opportunities for such testing are relatively limited. To increase the scope of comparison across time and feature extraction techniques, we evaluated the ability of FENet to reconstruct the movement kinematics using previously collected neural data recorded from implanted electrode arrays. In particular, we used data collected during an 'open-loop' paradigm, in which the participant attempted movements as cued by a computer-controlled cursor performing the centre-out task. Given that FENet is a neural network, and neural networks have the potential to overfit, the data that we used to train the FENet was 100% separate from the validation and the test data. Figure 3a,b shows the reconstruction performance of a linear decoder operating on TCs, WTs and FENet extracted features. These figures also compare the performance of FENet with other types of feature, including multi-unit activities (MUA)[20,38], high-frequency local field potentials (HFLFP)[4,34] and the combination of FENet and TCs with HFLFP. As FENet seeks to provide a new solution to the feature extraction process, we held the feature decoding stage constant across all feature extraction techniques to minimize confounds to interpretation. Comparisons were made for two human participants, JJ and EGS, on 54 recorded sessions spanning 2019 to 2022 for JJ and on 175 recorded sessions spanning 2014 to 2018 for EGS. Figures 3b and 4 upper panels show that for JJ, FENet improves the average cross-validated coefficient of determination (R^2) (Methods) of TCs ($t = -19.368$, $P < 0.0001$) and WTs ($t = -17.338$, $P < 0.0001$)

https://doi.org/10.1038/s41551-024-01297-1

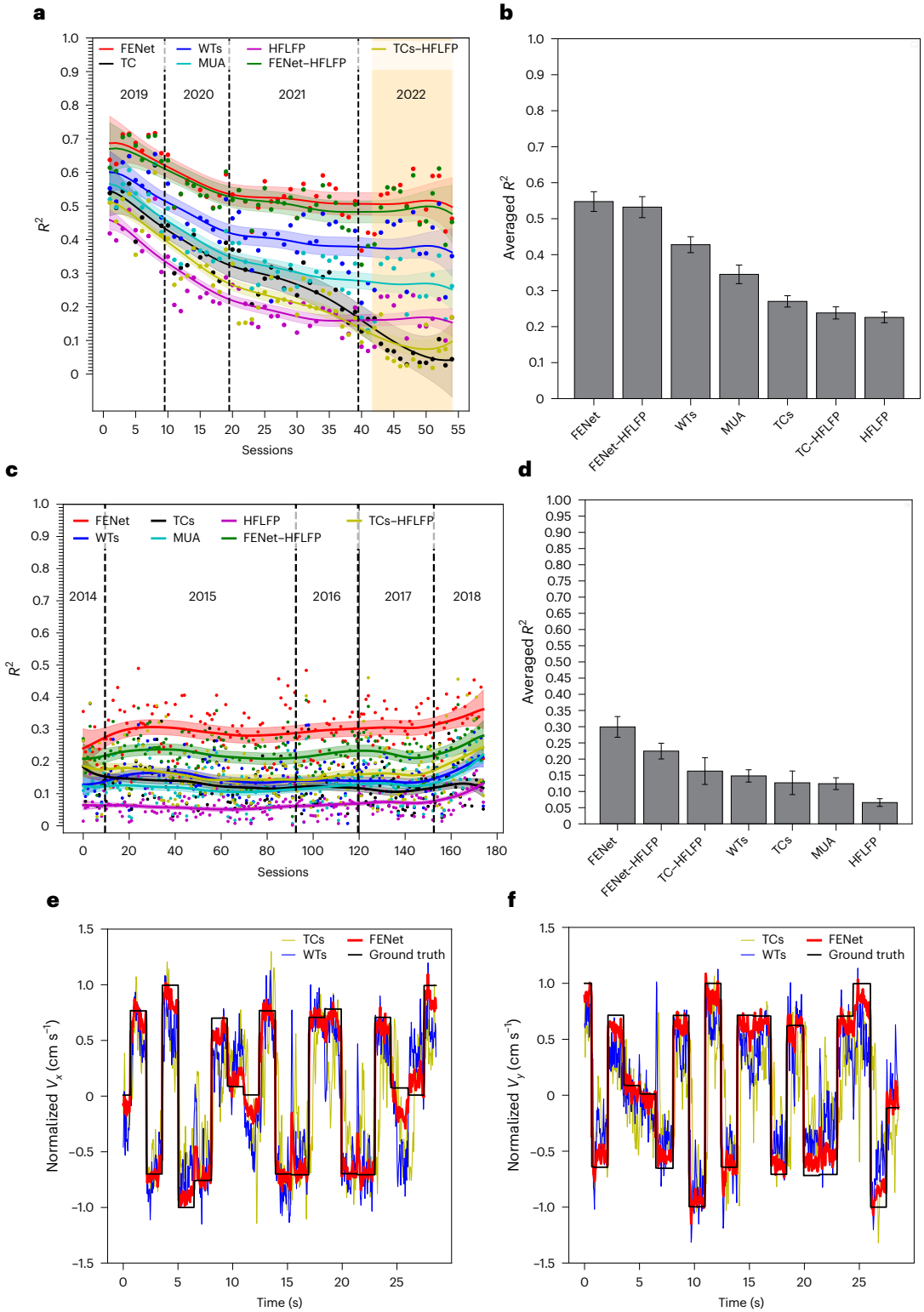


Fig. 3 | Open-loop multi-electrode performance. Performance for FENet, TC events, Debaucheries WTs, MUA, HFLFP and the combination of FENet and TCs with HFLFP (FENet–HFLFP and TCs–HFLFP). **a**,**b**, The research participant JJ over 54 recorded sessions spanning from 2019 to 2022 (shaded region shows the closed-loop sessions) per session performance (**a**) and averaged performance (**b**). **c**,**d**, The research participant EGS over 175 recorded sessions spanning from 2014 to 2018 per session performance (**c**) and averaged performance (**d**). The dashed lines separate the sessions of different years. The band in each time series shows the range of its 95% confidence interval of a locally estimated scatterplot smoothing[84,85] fit. **e**,**f**, Single experimental session 20190507 example (fourth session in **a**) of reconstructed instantaneous velocity of participant JJ showing reconstructions from FENet, WTs and TCs for horizontal (V_x) (**e**) and vertical (V_y) (**f**) dimensions. The black line shows the ground-truth target velocity, and the coloured lines show the reconstruction of the feature extraction techniques.

https://doi.org/10.1038/s41551-024-01297-1

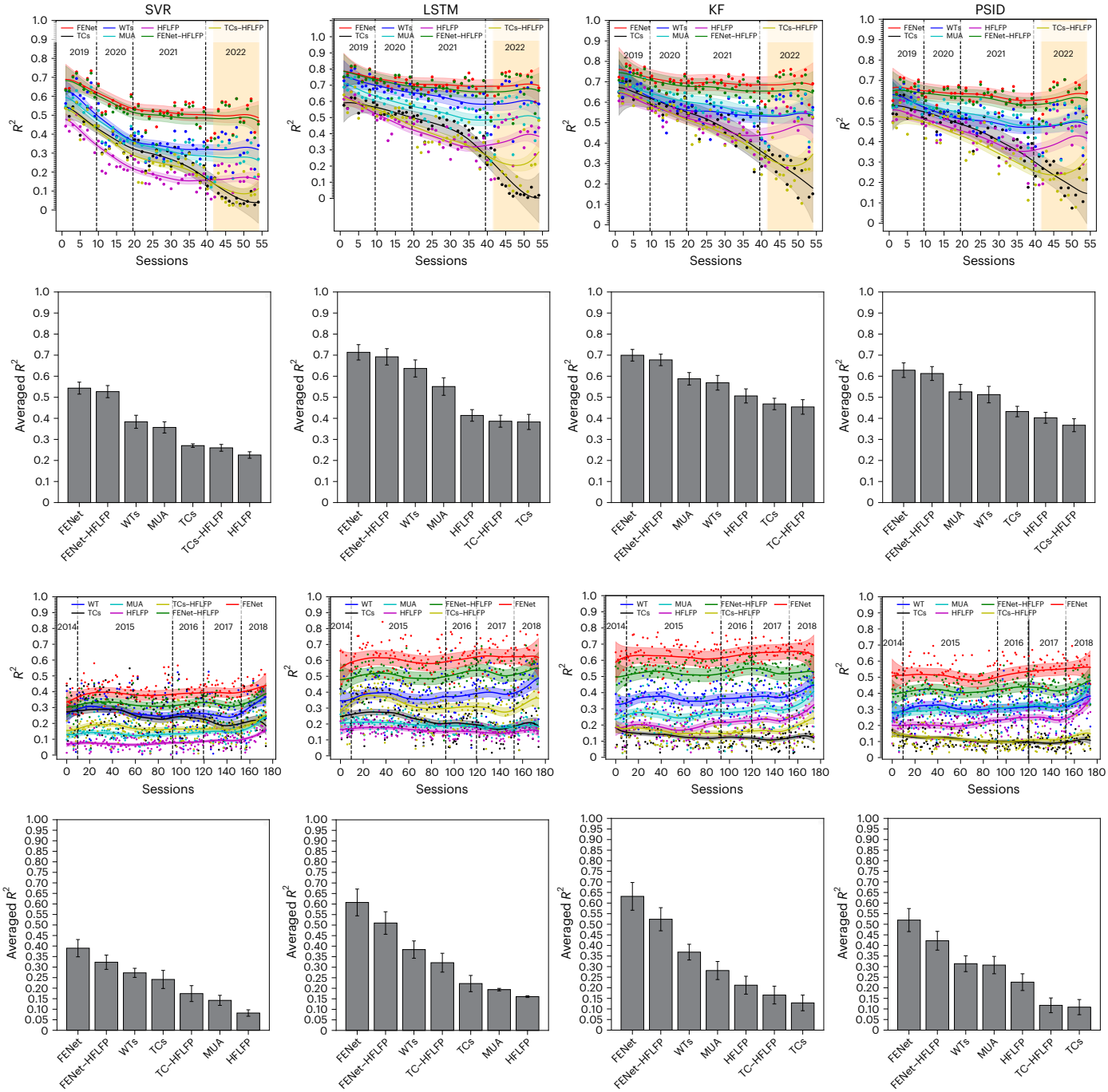


Fig. 4 | FENet features improve off-line decoding performance across all tested decoding algorithms. The upper two rows of panels and lower two rows of panels show the decoder performance for participants JJ and EGS, respectively, in a centre-out task. Performance is shown for five total decoding algorithms, each applied to features generated by seven feature extraction techniques. The analysed feature extraction techniques include FENet (red), WT transform with db20 mother WTs (blue), TCs (black), MUA (cyan), HFLFP (purple), the combination of FENet and HFLFP (green) and the combination of TCs and HFLFP (yellow). The tested decoding algorithms are SVR (first column), LSTM (second column), Kalman filter (KF; third column) and PSID (fourth column). The performance of every decoding algorithm was improved when trained on FENet features versus features generated using the other feature extraction techniques.

from 0.27 and 0.43 to 0.55, respectively. Figures 3d and 4 lower panels show that for EGS, FENet improves the average cross-validated R^2 value of TCs ($t = -39.012$, $P < 0.0001$) and WTs ($t = -28.281$, $P < 0.0001$) from 0.13 and 0.15 to 0.30, respectively. Figures 3a,c and 4 show that these improvements were found for each individual recording session as well. Figure 3e,f shows example reconstructions of the cursor velocity in x and y directions for a session recorded for JJ in 2019 and highlights how FENet reduces both trial-to-trial variability (FENet in red line is closer to ground truth for each trial repetition) and within-trial variability (FENet in red line demonstrates less variability within each trial). Extended Data Fig. 3a,b shows that FENet does not rely on the low-frequency (<250 Hz) local field potentials to achieve its enhanced decode performance. As designed, FENet improves population decoding by increasing the behavioural information content of almost every electrode (Extended Data Fig. 4a–c). It is worth noting that, although FENet improves the R^2 between neural features and kinematics

compared with WTs and TCs, Extended Data Fig. 4d–j shows that FENet reports similar tuning preferences to TCs and WTs at the same electrodes.

To ensure that improvements in our feature extraction method generalize across feature decoding methods, we have also included the performance of additional feature decoders, namely, support vector regression (SVR)[39–41], long short-term memory (LSTM)[42] recurrent neural network (RNN), Kalman filter[5,24] and preferential subspace identification (PSID)[43]. Figure 5 provides a comprehensive evaluation of the performance of these decoders operating on different feature extraction techniques. The comparisons were conducted using optimized parameters for each decoder, ensuring fair evaluations rather than relying on standard configurations (Methods). As we see in this figure, FENet improves decoding R^2 compared with the other feature extraction techniques for all the tested decoders.

We evaluated the open-loop results with FENet using neural data and behaviour binned at a fine temporal resolution (30 ms bins) and without smoothing the extracted features. This was motivated by our primary goal that FENet be maximally useful for closed-loop control where smoothing decreases the responsiveness of the closed-loop system by using potentially outdated neural information. However, recognizing that FENet could also be used for slow-timescale applications, we tested how FENet performed against TCs when smoothing the extracted features from a larger window size. Extended Data Fig. 5a,b shows the robustness of FENet against the change in the window size used to update the feature extraction process in our trajectory tasks. This analysis aimed to assess the impact of varying feature extraction window sizes on the performance of decoders using the extracted features. Throughout this study, FENet remained trained on data partitioned into 30 ms bins. However, we expanded the window size to ascertain whether FENet showed superior performance compared with TCs during inference. Furthermore, the utilization of a more extensive history of broadband data for feature extraction with larger window sizes introduces a smoothing effect in the decoding process. Consequently, we observe that both feature extraction techniques demonstrate improved decoder performance owing to this inherent smoothing effect.

To assess and understand the effectiveness of the extracted features obtained through diverse feature extraction techniques, we conducted a rigorous analysis using offline data of the best electrode from three sample sessions labelled as 20190314, 20200928 and 20210312 (Extended Data Fig. 8). The data of each session consisted of eight distinct trials, each corresponding to a unique target location in a centre-out task. To differentiate between these targets, we designated the location where $x > 0$ and $y = 0$ as Target0. Our analysis focused primarily on the feature values derived from the top electrode recorded during these sessions. To identify the top electrode within a session, we organized electrodes based on their individual electrode R^2 values, indicating the linear predictability of kinematics for each electrode using each distinct feature extraction technique. Subsequently, we randomly chose three sample sessions spanning 2019, 2020 and 2021 from those where the index of the top electrodes remained consistent across all feature extraction techniques. The results of our analysis demonstrated the remarkable preservation of the fundamental tuning characteristics of the neurons across the various feature extraction techniques used. It is worth noting that FENet showed substantial improvement in the amplitude of the preferred versus anti-preferred directions in the tuning curves, thus improving the ability to distinguish individual trials. These findings indicate that FENet provides a more robust and distinctive representation of the neural activity, thereby enhancing the performance for decoding neural signals.

Interpreting machine-learning algorithms, especially deep learning, in medical applications is a considerable challenge[44]. To enhance our understanding of the differential characteristics of FENet in comparison to WTs, MUA and HFLFP feature extraction techniques, we turn to the existing literature on interpretability in deep learning[45–48]. A key aspect worth exploring is the utilization of filter shapes that show discernibly distinct frequencies (Extended Data Fig. 9). Specifically, we examined the gain, or the amplification capability, of a sample set of FENet trained convolutional filters across its feature engineering modules. In contrast to the other filters, FENet showed a unique characteristic of dynamically amplifying specific frequency bands during its training process. FENet's training mechanism takes into account the encoded information within each frequency band, allowing it to selectively enhance relevant features within different frequency ranges. This ability to dynamically amplify distinct frequency bands sets FENet apart from conventional filters such as WTs, MUA and HFLFP. This deviation from conventional approaches indicates that FENet operates in a manner fundamentally distinct from these conventional feature extraction techniques such as WTs, MUA and HFLFP. While the exact nature of this divergence requires further investigation, it is evident that FENet functions by adaptively adjusting its filters based on the specific frequency information, which shows a more nuanced and refined approach of feature extraction and leads to improved performance in analysing neural data.

To gain more insight into the specific regions of input data that receive more attention from FENet during its prediction process, we present two illustrative examples of single-electrode input samples obtained from FENet and WTs (Extended Data Fig. 9, lower panel). These samples were collected during a specific session identified as 20190625. To highlight the segments of higher importance in the predictions made by the linear decoder, we use colour-coded visual representations. To accurately depict the most relevant sections of the input signals, we calculated the average Shapley value[45,46] (Methods) across all samples. Subsequently, we selectively coloured the samples whose Shapley values surpassed this calculated average threshold. In addition, a horizontal line is included in the figures to denote the threshold used for extracting features associated with TCs from each input sample. The presented figures demonstrate that FENet, following its training, not only leverages spike information (TCs) and WT transforms but also shows superior capabilities in identifying local patterns within the input data. Furthermore, FENet demonstrates an exceptional proficiency in accurately tracking rapid and abrupt changes present in the input signals compared with WTs. These empirical findings collectively suggest that FENet possesses the remarkable ability to capture more intricate and localized information, which can result in enhanced feature extraction capabilities when compared with the conventional WT and TC approaches.

FENet generalizes across time, brain areas and participants

For FENet to have maximum public impact, it should work across participants, in any implanted region of the brain, for any subset of electrodes and for the duration of the implant recordings. In other words, although FENet was trained using a particular set of participants and brain areas, the resulting solution should apply more generally to any situation in which the functional state of the brain must be inferred from electrical recordings. To understand how well FENet generalizes to the unseen data, we split our training data in various ways (in time, brain area, participant and electrode subset) and compared performance within and across our data splits. Extended Data Fig. 10 demonstrates that FENet generalizes within and across splits. For example, in Extended Data Fig. 10a, we show that decode performance on data collected from participant JJ in 2022 is similar whether FENet was trained on the same 2022 data or any previous year of the implant. This is remarkable given the notable changes in the quality of electrical recordings over this time span (for example, see Extended Data Fig. 2). It is worth noting that, in all cases, generalization performance was substantially better than TCs or WTs applied to the same dataset (Fig. 3b,d). These results suggest that FENet can generalize across different time periods, brain areas, participants and electrodes (Extended Data Fig. 6f).

https://doi.org/10.1038/s41551-024-01297-1

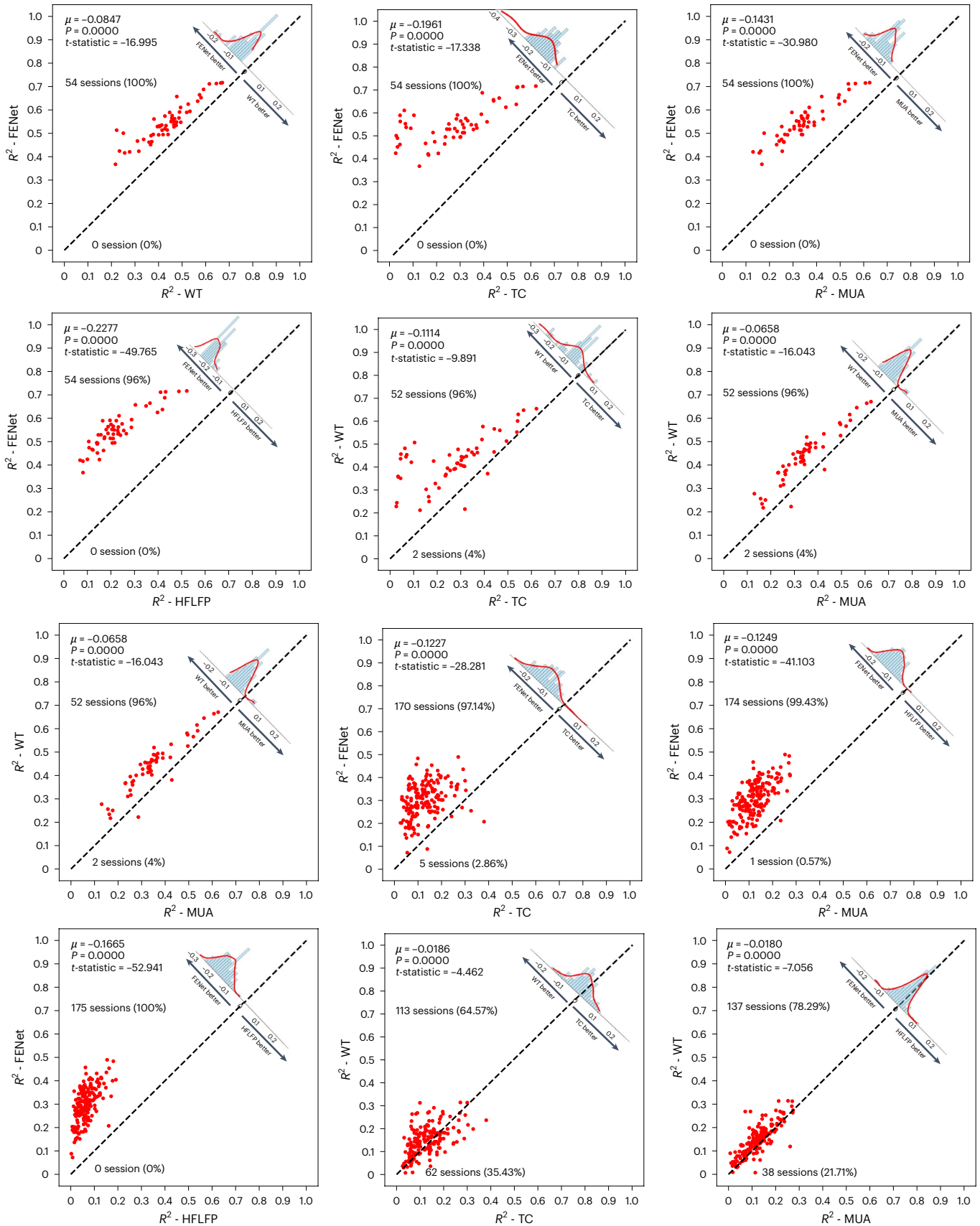


Fig. 5 | Comparative performance of feature extraction techniques. Comparison of the cross-validated R^2 of linear decoder operating on one feature extraction technique versus the other technique for participant JJ (top two rows of panels) and participant EGS (bottom two rows of panels). Red dots denote the sessions. The dashed line shows $y = x$. The percentage of dots on each side of $y = x$ shows the number of sessions in favour of the corresponding feature extraction technique. The t-test statistics have been calculated to show the confidence level of the reported statistics. According to these figures, the linear decoder operating on the FENet-based features provides superior performance in terms of R^2 compared with other feature extraction techniques for both participants.

FENet generalizes across tasks

FENet substantially improved our ability to decode instantaneous cursor velocity in the centre-out and grid trajectory tasks. We next demonstrated that FENet could serve as a drop-in solution to improve the information content of neural features in a different task. To this end, we chose to apply FENet to a previously published 'finger flexion grid' task dataset[49] based on the three characteristics of the dataset: (1) Intended BCI movements may be confounded with overt movements (for example, of the head and eyes) as the participant orients to a target. The finger-grid task explicitly dissociates overt movements from the neural signals of interest by randomizing the cue location. (2) The populations of the sorted units collected during the finger-grid task exhibited representational structure that dynamically changed through time. The ability of FENet to recapitulate these representational dynamics, with improved SNR, would further validate that FENet can be dropped into any neuroscience and neuroengineering processing chains. (3) In the finger-grid task, we test the ability to decode movements of each finger, which demonstrates that FENet generalizes to additional variables of interest to neural prosthetics. Finally, the finger-grid dataset was collected from participant NS, and thus, the successful application of FENet would demonstrate generalization of FENet to a new participant.

Supplementary Fig. 1a shows a schematic representation of the finger-grid task. In response to a visual cue, participant NS immediately attempted to press the corresponding finger, as though pressing the key to a keyboard. Movements were cued by having a cursor move randomly across a 4-by-3 grid of letters. The participant oriented her head and eyes to each position on the board after which she attempted the instructed movement. Supplementary Fig. 1b shows that FENet features improved our ability to distinguish individual finger movements, here captured as the cross-validated Mahalonobis (crossNobis) distance[49,50] between fingers. It is worth noting that the relative magnitude and timing of FENet encoding of the location of the spatial cue (Supplementary Fig. 1c) was much smaller than what we found for digit encoding (Supplementary Fig. 1b). This suggests that FENet features are not unduly influenced by factors associated with overt movements such as head or cue position and instead maintain the specificity of populations of sorted neurons. Finally, a comparison of Extended Data Fig. 1d,e shows that FENet preserves the representational structure and dynamics of populations of sorted neurons. Taken together, these results demonstrate that FENet can improve decoding of a dataset from a new participant with electrodes implanted in different brain regions, while maintaining the specificity and preserving the detailed representational structure of sorted single neurons.

Similar to the case of the cursor control task (for example, Extended Data Fig. 5a,b), we tested how FENet performed against TCs when smoothing the extracted features. Extended Data Fig. 5c shows that the relative benefit of FENet is diminished with increasing smoothing windows, although it maintains a benefit over TCs.

Discussion

Implantable electrode-based BCIs promise to restore autonomy to paralysed individuals if they are sufficiently robust and long lasting to overcome the inherent risks associated with brain surgery. Unfortunately, the breakdown of materials in the hostile environment of the body[37,51] and inherent stochasticity of the quality of information available at individual electrodes[52,53] provide a substantial hurdle for the safety and efficacy of implantable solutions. Advances in material sciences, minimally invasive delivery and design modifications provide one path to overcome these limitations, but they may take many years to receive US Food and Drug Administration (FDA) approval and may not improve baseline decoding quality. Here we present an algorithmic solution, demonstrating FENet's ability to extend the lifetime of implanted electrode arrays and to generally improve the performance of the BCI system using a simple drop-in solution.

Multiple aspects of single-unit, multi-unit and population-level neural behaviour can be detected from a single electrode. Examples include the waveforms of single-neuron action potentials, multi-unit hash, local field potentials and synchronous population responses[36] (Fig. 1a). Intuitively, it is tempting to think that the activity of well-isolated single neurons would be the most informative features; however, within the constraints of current recording set-ups, empirical evidence has contradicted this intuition[54,55]. With FENet, our goal was to create a structured learning problem so that we could discover the best transformation using neural and behavioural data. At the same time, we imposed several constraints to facilitate a solution that would be generalizable. For example, we baked in the constraint that each electrode uses the same transformation and that this transformation be independent of behaviour. This constraint was essential for our goal to find a generalizable solution, a goal that reflects an underlying principle of our approach that the biophysical elements giving rise to the electrical activity at a given electrode are consistent across electrodes, brain areas, individuals and time. Simplifying the argument, all human brains are made up of neurons and the associated interconnective matter, and all neurons in human brains generate action potentials. Furthermore, the waveform of an action potential is predominantly a function of the relative location of the electrode tip and the neuron, not the neuron's role in behaviour[56]. We assume that the precise spatiotemporal features of broadband data report the relative activity of the local neural ensemble and not details of the behaviour per session. Therefore, in a similar vein to conventional feature extraction methods that apply uniform operations to individual electrodes, FENet uses a consistent feature extraction process across all electrodes. Furthermore, it is crucial to consider the bandwidth limitations of implantable BCIs. Even with systems that currently record high-sample-rate broadband data, transmitting such data off-device for continuous model retraining remains impractical due to bandwidth and power constraints. This scenario underscores the critical advantage of FENet, which can be directly deployed without requiring further training on newly collected broadband data. This capability paves the way for robust BCI performance across diverse participants and implants.

FENet was designed to be a feature extraction method that could easily be dropped into current decoding pipelines with existing decoding approaches and thus must generalize across electrodes, participants, brain areas and time, despite potential recording non-stationarities. It is possible that neural networks that are electrode specific could improve performance, but this would occur at the cost of generalizability and ease of use. Furthermore, given recording non-stationarities and limited access to training data, the practical ability to train electrode-specific realizations of FENet is non-trivial. Consequently, this limitation constrains the potential benefits of hyper-specific solutions tailored to individual electrodes. Therefore, FENet is designed to be agnostic to the specific number and configuration of electrodes within different BCI systems, making it readily adoptable by users, particularly those who prefer to avoid setting up their own training protocols.

In this study, we conducted an offline analysis to compare FENet's performance with multiple other feature extraction techniques using different linear and nonlinear decoders. To expand the scope of comparison across different time periods and feature extraction techniques, we assessed FENet's capability to reconstruct movement kinematics using previously recorded neural data from implanted electrode arrays. A retrospective analysis over years of recordings showed that FENet significantly improved the cross-validated R^2 and the SNR of extracted neural features compared with the other feature extraction techniques across multiple participants and through the lifetime of the arrays. Furthermore, FENet generalized well across cortical brain regions, participants and tasks, demonstrating its ability to serve as a drop-in replacement for other feature extraction techniques. Moreover, the population-level analysis demonstrated that

FENet preserves the representational structure and temporal dynamics of sorted neural populations and, thus, provides an accurate measure of brain activity. Consequently, according to the results of our offline analysis, we evaluated and compared the performance of TCs, as the current standard for closed-loop control[4,6,22,26,27,31,34], and WT transforms, which have also demonstrated performance improvements in our offline analysis and in the recent studies on BCIs[20,29], against FENet features in our closed-loop analysis. Neural decoders using FENet-based features outperformed TC-based and WT-based features across all metrics. In addition, FENet enhanced the cursor's responsiveness to the participant's intent, reducing the time it took for the cursor to move towards the target after its onset. These improvements in both aspects led to substantial enhancements in overall task performance during the closed-loop sessions. While we reported both the open- and closed-loop performances for participant JJ, our evaluation of the presented feature extraction techniques was limited to the recorded open-loop neural data from participants EGS and NS in specific tasks, as their participation in the clinical trial concluded, and their electrodes were explanted. Nevertheless, the principles derived from analysing the recorded open-loop neural data from EGS and NS will hold relevance for future participants with electrodes in the same cortical areas. Taken together, FENet can improve the efficacy of implantable electrode systems while delivering improved performance and ease of use.

It is worth noting that, across all testing conditions, FENet improved results when analysis was done at fine temporal scale. However, in some cases, the benefits of FENet were reduced as smoothing was applied to the data. Thus, FENet seems to considerably reduce high-frequency within-trial variability, yet may have less impact on reducing trial-to-trial variability (Extended Data Fig. 5d), depending on experimental set-up. Reducing the high-frequency variability is critical for real-time BCIs, situations in which the behaviour unfolds over very rapid timescales, when looking for precise estimates of timing or when attempting to infer network dynamics. Trial-to-trial variability can be captured by how similar the neural response is when repeating the same experimental trial. The trial-to-trial variability is an accurate report of how behavioural factors have changed the activity measured at the single-electrode level. Reducing the trial-to-trial variability is also important; however, the underlying reasons for trial-to-trial variability are less clear. For instance, measured differences may be accurate reflections of the underlying state of the network that, for example, reflect task-irrelevant features of the participant's behavioural state[57]. To the degree that trial-to-trial differences are driven by behavioural factors, no algorithm measuring the activity from a single electrode can reduce this variability, although populations of such electrode recordings can reduce variability through estimates of latent variables[22].

It is important to note that FENet was designed to maintain a small computational footprint in comparison to ultradeep RNN feature extraction techniques and other convolutional network designs. This was achieved by extracting features from single electrodes using the same trained parameters for all electrodes. We deliberately constrained the architecture to an algorithm with complexity that allows for computation within 5 ms in closed-loop BCIs. The presented version of FENet, based on the WT with db20 mother WT architecture described in the paper, consists of only 560 learnable parameters. This notably reduces its size compared with more complex deep-network alternatives. In addition, we conducted experiments by swapping hyperparameters of FENet, demonstrating that we can achieve comparable benefits and performance even with a smaller architecture. Exact hardware details will be explored further in our upcoming works, which will focus specifically on hardware implementation.

Our primary focus within this study was to test techniques that can be applied in a causal real-time system. To explore the effect of applying a non-causal filter when extracting TCs[58], we evaluated the decoder's performance when using FENet as the feature extraction technique versus TCs preceded by a non-causal filter that was applied over the data twice—once forward and once in reverse. Supplementary Fig. 2 shows the performance of the decoder on eight sessions. This figure highlights the relative strengths of different feature extraction approaches in enhancing decoder performance, with FENet emerging as the most effective technique. The non-causal filtering applied to TCs improves performance but is still notably less effective than FENet in terms of decoding accuracy.

Traditionally, BCI systems can trade off speed and accuracy depending on the design preferences. The ability of FENet to improve on both sets of metrics in parallel represents a considerable advance in BCI design. It is worth noting that these advantages come with little or no cost in either computational or experimental performance. FENet preserves the representational structure of sorted neural populations and therefore should be applicable to any subsequent decoding scheme. Moreover, FENet remarkably improved a human clinical BCI participant's ability to use brain signals to control a computer cursor in the closed-loop control compared with TCs and WTs. This performance increase was clinically important: before FENet, the clinical participant requested surgical reimplantation to improve the quality of neural recordings that had degraded substantially since the initial implantation. With FENet, the participant was satisfied with the quality of his neural control. Thus, FENet can extend the functional lifetime of the implanted electrodes, mitigating the need for revision surgeries and thus improving commercial viability. It is important to clarify that our reference to improved performance specifically pertains to the feature extraction component, where the participant serves as their own control. Furthermore, it is crucial to acknowledge that the performance of a BCI system can be influenced by various factors, both within and outside of our control. These factors may include the nature of the participant (human or non-human primate), implant site and age, recording yield, task and the specific decoder used[5,32–35]. Recognizing this inherent heterogeneity in BCI performance across participants, tasks and labs, we adopt a within-participant experimental set-up to evaluate FENet. We observed enhanced performance when using features extracted by FENet with improvements across all datasets included in this study (3 human participants, 192 total electrodes and many hours of neural data representing multiple years of implantation). We found that FENet generalized well between three participants, three brain regions, closed- and open-loop settings and up to 5 years of recordings. This provides preliminary confidence that FENet provides a generalizable improvement to current feature extraction methods. However, it remains possible that FENet will not improve performance across all participants, tasks and array technologies. We therefore provide our code in a public repository in the hope that additional clinical sites will test and ultimately improve FENet.

Methods

Human participants and neural recordings

We conducted our FDA- and Institutional Review Board-approved BCI study with a 54-year-old (referred to as JJ) and a 32-year-old (referred to as EGS) tetraplegic (C5-C6) male human research participants for trajectory tasks. Participant JJ has Utah microelectrode arrays (NeuroPort, Blackrock Microsystems) implanted in the hand-knob of motor cortex and superior parietal lobule of the posterior parietal cortex (PPC)[59]. Participant EGS has Utah electrode arrays implanted near the medial bank of the anterior intraparietal sulcus and in Brodmann area 5 (refs. 8,60,61). We collected open-loop data over 54 sessions for JJ and over 175 sessions for EGS in our open-loop analysis. Broadband data were sampled at 30,000 samples per second from the two implanted Utah microelectrode arrays (96 electrodes each). For the finger-grid task, we recorded the single- and the multi-neuron activities from a tetraplegic 62-year-old female human participant with a complete C3–C4 spinal cord injury (referred to as NS)[49]. We recorded nine sessions of the broadband neural activity from a Utah microelectrode array implanted in the left (contralateral) PPC at the junction of the

post-central and intraparietal sulci[59,61,62]. This region is thought to specialize in the planning and monitoring of grasping movements[60,63–65]. In this work, although we have reported the open- and closed-loop performances for participant JJ, we have only evaluated the presented feature extraction techniques on the recorded open-loop neural data of EGS and NS in the trajectory and the finger-grid tasks, respectively, as EGS and NS have completed their participation in the clinical trial and have had the electrodes explanted. However, the principles learned from the analysis on the recorded open-loop neural data from EGS and NS will be relevant to the future participants with electrodes in the same cortical areas.

Behavioural tasks

Data were collected while participants performed various 2D control tasks, including centre-out[5,19,28], grid and finger-grid[49] tasks using pseudo-random interleaving of targets to ensure balanced statistical sampling of movement directions[8]. In the centre-out task, a cursor moves in two dimensions on a computer screen from a central target outward to one of the eight targets located around a circle and back to the centre (Fig. 1e,f, top). We define a trial to be one trajectory, either from the central location outward to the peripheral targets or from the peripheral targets back to the centre target. In the grid task, the target appears in a random location in an 8-by-8 squared grid on the computer screen, and the cursor moves starting from the old target to the newly appeared target (Fig. 1f). Cursor movement kinematics are updated every 30 ms for JJ and every 50 ms for EGS. For the purposes of this study, we extracted trajectories from 200 ms after target presentation to 100 ms before the cursor overlapped the target. This segment of time captures a window where the participant's intent should be well defined, after reacting to the presented target and before possibly slowing down as the cursor approaches the target[8]. Neural features were regressed against cursor velocity, which, for simplicity, was modelled as constant amplitude. Each of these tasks was conducted in either open-loop, in which the cursor movements were fully generated by the computer and the participant did not directly control the cursor's position but instead imagined control over a visually observed, computer-controlled cursor, or closed-loop, in which the cursor movements were under the participant's full control with 0% assistance from the computer.

For the finger-grid task, a text cue (for example, 'T' for thumb) was displayed to the participant on a computer screen in each trial. Then, the participant immediately attempted to press the corresponding finger of the right hand[49,59] (Supplementary Fig. 2). To model the multi-finger tasks, we considered the muscle model and somatotopy model[50]. The muscle activation model posits that the representational structure should align with the coactivation patterns observed in muscle activity during individual finger movements. Conversely, the somatotopy model suggests that the representational structure should correspond to the spatial arrangement of the body, wherein neighbouring fingers exhibit similar representations. Although somatotopy typically pertains to physical spaces resembling the body, in this context, we use the term broadly to encompass encoding spaces that resemble the body[49].

Preprocessing the broadband neural data

To reduce the effect of high-frequency noise, which has not been removed by the recording hardware, we applied common average referencing (CAR) to the recorded broadband neural data as the first step of the preprocessing[66]. To apply CAR, we used principal component analysis to remove the top two principal components across each electrode before transforming the remaining principal components back to the time domain. After applying the CAR to the recorded broadband data, we applied an 8-order elliptical high-pass filter with the cut-off frequency of 80 Hz, pass-band ripple of 0.01 dB and the stop-band attenuation of 40 dB to the common-average-referenced neural data to exclude the low-frequency variations in the broadband neural activities. Given that we estimate FENet features in a window duration of 30 ms, the theoretical lower frequency limit is approximately 33.33 Hz. However, to accurately estimate frequency content, we need at least two cycles of the lowest frequency within our window. Setting a high-pass cut-off at 80 Hz ensures more than two cycles within the 30 ms window, providing a more reliable frequency estimation.

FENet pipeline

To train FENet, we created a two-stage optimization problem, which transformed broadband signals into the movement kinematics within a BCI cursor control paradigm: In the first stage, broadband activity is transformed into neural features by using FENet as a 1D convolutional neural network. In the second stage, an analytic linear mapping is trained to predict the movement kinematics from the resulting neural features. The two-stage joint optimization enforces that the feature extraction process generates informative features while being independent of the relationship between the neural activity and the cursor kinematics. As each electrode records a relatively independent 1D temporal signal, we use 1D convolutional filters in our feature extractor architecture to take in single-electrode broadband samples and output M features (that is, the instantaneous states of the various information sources on the electrode). Suppose that $x \in \mathbb{R}^S$ represents a 1D time series consisting of S samples of the broadband neural data recorded from one electrode, which has been sampled at the sampling frequency of F_s Hz. FENet can be represented as a function $\mathcal{F}_\psi$: $\mathbb{R}^S \to \mathbb{R}^{M\times N}$, which maps the input waveform to an M-dimensional neural feature space. $M < S$ shows the number of extracted features, and N is the number of electrodes. ψ corresponds to the feature extraction (in this case, FENet) parameters. The decoder can be represented by $g_\theta(.)$, in which g is parameterized by θ. Then, the supervised optimization problem that should be solved to find the parameters of the FENet and the decoder will be as follows:

$$\psi^*, \theta^* = \mathrm{argmin}_{\psi,\theta} E_{(x,y)\in D} \mathcal{L}\left(g_\theta(\mathcal{F}_\psi(x)), y\right). \quad (1)$$

where (x, y) are the samples in the labelled dataset, D. $\mathcal{L}$ represents the loss function, which in our regression problem is the mean squared error (m.s.e.) between the correct and the predicted movement kinematics of the cursor velocity. According to our assumption that the generative process that produces the broadband neural activity is statistically similar across electrodes, we design FENet such that it learns a single set of parameters ψ for all the electrodes. Thus, the same instantiation of FENet, as defined by the parameter set ψ, is applied independently to broadband data recorded from all electrodes.

The architecture of FENet in the BCI system is shown in Extended Data Fig. 1b. As a nonlinear feature extractor, FENet consists of a set of 1D convolutional filters, nonlinear activation functions and pooling layers[15–17]. Let $x \in \mathbb{R}^S$ denote the input of the FENet with size $1 \times S$, where S is the number of input data samples (for example, 30 ms of the recorded broadband neural data, which includes 900 samples for JJ and NS, and 50 ms of the recorded broadband neural data, which includes 1,500 samples for EGS). The input x is passed into $M-1$ back-to-back feature engineering modules (Fig. 1c). In each feature engineering module, the input data of the ith feature engineering module, s_{i-1}, is padded with zeros, and the zero-padded data are passed through the two separate temporal 1D convolutional filters. The output of the upper filter is downsampled by stride 2 and is passed through a leaky rectified linear unit (ReLU) nonlinear activation function. The leaky ReLU activation is designed to find the absolute value of its input with the parameter $\alpha = -1$ in the negative side. Then, the output of the current filter is passed through an adaptive average pooling layer to summarize extracted temporal patterns into a single feature, f_i. We pass the output of the lower filter, s_i, to the next feature engineering module. This process is repeated to find the output feature vector. We

pass the output of the lower filter of the last feature engineering module of FENet to a leaky ReLU activation and an adaptive average pooling layer to append this single extracted feature to the feature vector as well. Therefore, the upper convolutional filter in each feature engineering module generates one of the FENet extracted features, and the lower convolutional filter of each module extracts more abstract features from its input to be used as the input of the next feature engineering module. Finally, we use batch normalization as a regularization technique, which standardizes the output of the last layer of FENet to zero mean and unit variance for the training examples equal to the batch size. Batch normalization helps the used optimization algorithm by keeping inputs closer to the normal distribution during the training process[48]. FENet is unique as it is parameterized using an architecture that jointly optimizes the feature extraction and feature decoding stages of the neural decoding process, while constraining the feature extraction algorithm to use the same parameters for all the electrodes used in the training set. The constraint of sharing parameters across electrodes will keep the number of learnable parameters small in FENet architecture. Moreover, FENet is trained to receive a single neural electrode of broadband data as its input and extracts the signal's most informative features automatically. This process can be repeated for all recording electrodes to estimate the current state of a neural population independent from the decoder.

Generation of other features

We have extracted the features from the 30 ms bins of recorded broadband neural data for JJ and NS and from 50 ms bins for EGS without any post hoc offline steps. To extract WT features[19,20], we used a db20 mother WT with seven scales on moving windows (no overlap) of the time series recorded from each electrode. A db20 mother WT was selected as it contains filters with length 40 and can model WT high-pass and low-pass filters more accurately compared with other Daubechies WT families[20]. Moreover, in our preliminary experiments, we found that it outperformed other WT variants for the datasets tested in the current study[19] (such as db4, Haar). The mean of absolute-valued coefficients for each scale was calculated to generate $M = 8$ time series per electrode, including seven detailed coefficients and one approximation coefficient generated by the WT high-pass filters and the final stage WT low-pass filter, respectively. To generate TC features, we set a threshold for the neural data at −3.5 times the root mean square of the noise of the broadband signal, independently computed for each electrode, after band-pass filtering the broadband signal between 250 Hz and 5 kHz. We did not sort the action potential waveforms[21]. TC events were counted using the same intervals as WTs and FENet. To derive the MUA features, the raw broadband neural data underwent a bandpass filtering process (a third-order Butterworth filter) with a frequency range of 300 to 6,000 Hz. Following this, customized root mean square values were calculated to generate the MUA signal for each bin[38]. To generate the HFLFP features, the raw broadband neural data from each electrode underwent a second-order band-pass filtering process using a Butterworth filter with low and high cut-off frequencies set at 150 Hz and 450 Hz. The power of the filter's output was then calculated and used as the HFLFP feature for each electrode[4,34]. For FENet–HFLFP and TC-HFLFP, we simply concatenate the corresponding features together to generate a larger feature matrix that include both types of extracted feature.

Preprocessing of the generated features for open- and closed-loop analysis

During our offline analysis, we intentionally refrained from applying smoothing to the features under investigation. Smoothing techniques have the potential to enhance the R^2 of decoder output in ways that do not generalize to online control. This is because smoothing works by averaging across time, effectively introducing control delays that render the cursor uncontrollable or undesirably sluggish. Therefore, we opted to evaluate performance in the absence of smoothing. By contrast, during closed-loop control, we used exponential smoothing[58] as a preprocessing step for the extracted features. This was done to mitigate abrupt changes and jitters for improved stability[67]. With the participant in the loop, we could ensure that the level of smoothing was not burdensome to the participant. Finally, we also evaluated off-line performance with smoothing (either explicitly or implicitly in the architecture of the decoding algorithm) to ensure that improvements with FENet were not restricted to high-frequency components of the signal that could easily be removed by subsequent signal processing.

Given the flexibility of FENet's and WTs' design to accommodate varying numbers of feature extraction levels, the resulting impact on the number of features extracted from each electrode necessitates the reduction of dimensionality. This reduction is essential to prevent overfitting of the decoder during individual sessions. To address this concern while maintaining the single-channel architecture of the feature extraction technique, we used partial least squares regression[68] (PLSR). Specifically, PLSR was independently applied to the features extracted from each channel. The objective was to condense the eight extracted features obtained from each electrode into a smaller set of features, specifically two features in this case.

Training and inference for FENet

The architecture of the BCI system is shown in Extended Data Fig. 1b. The input of the system is the broadband neural data with the dimension of $B \times N \times S$, where B is the batch size, N is the number of input neural electrodes and S is the number of samples of the broadband neural data in a specific time interval. To update the network parameters during training, we randomly picked one training session and passed a batch of the associated broadband activities to the FENet to extract neural features. According to our experiment, the best performance was achieved when we set the batch size to be equal to the length of a session. Moreover, we use one training session for each update cycle as it is the only way that simultaneously acquired neural recordings can be associated with corresponding cursor kinematics. The same FENet parameters are applied to all the N neural electrodes. The output of the FENet is a feature matrix with the dimension of $B \times (N \times M)$, where M is the number of the generated neural features per electrode. This feature generation process is the first stage of the two-stage optimization process. To reduce the dimension of the FENet output per channel to avoid overfitting of the consequent decoder, we applied M electrode-specific PLSR[68] to the M FENet-generated features of each neural electrode to reduce the M features to K, in which $K \leq M$. We then used the output of the FENet, which was applied on a single session at the current iteration, to train an analytical linear decoder, which learns to map the extracted neural features to the movement kinematics of the computer cursor for the current single session analytically by the following formula[23,69]:

$$P = U\beta + \varepsilon \tag{2}$$

$$\beta = (U^T U)^{-1} U^T P \tag{3}$$

where P is the $B \times 2$ kinematics matrix, U is the $B \times K$ extracted neural feature matrix, β is the linear decoder coefficient and ε is the regression error. As predicting the velocity of the cursor movements in a BCI system is more stable and smoother than predicting the cursor position[70], we first predict the cursor velocity by using the decoder. Then, to find the position of the cursor movements, we integrate the predicted velocity patterns of the cursor in x and y directions[26,27]. After the linear decoder predictions, we froze the trained linear decoder parameters and performed backpropagation[71] to only update FENet weights. We repeated this whole process to train FENet and linear decoder parameters per system update, which happened per session.

For the symmetric replication of the feature engineering modules of the FENet, we designed FENet to have a hierarchical and symmetric

architecture similar to the db20 WT transform. As the FENet architecture is inspired by the WT transform architecture, we initialized the FENet convolutional filters with db20 mother WT filters to guarantee the convergence of the FENet by a more accurate initial condition at the beginning of training[19,20]. We used seven back-to-back feature engineering modules in the FENet architecture (Fig. 1c). We set the length of each feature engineering module's convolutional filter to 40, similar to the length of db20 filters. The convolutional filters kernel sizes and the strides of the filters were set to 1 and 2 for all the convolutional filters, respectively. To compensate for the left and the right edge effect of the convolutional filters' inputs during the convolution operation, we padded 39 zeros to both sides of the inputs at the first block of the feature engineering modules, which is one less than the filter length to make sure the first convolution only covers the first sample of each input. To tune the network parameters and to train the network, we have used the open-loop neural data recorded from 11 sessions of the first year of JJ's implantation. Extended Data Fig. 6c shows the amount of training data needed to train FENet. To train FENet, we did cross-validation by dividing the training sessions into train and validation sessions, holding three of the sessions out for validation, while training the network on the remaining eight sessions. For training the linear decoder after FENet generated the features per session, we applied the 10fold cross-validation on each session. To avoid overfitting, we used early stopping to stop the training when the validation loss on the left-out validation sessions started to increase[48]. We also used dropout, which has been shown to reduce overfitting in neural networks[72]. To control the range of the values of the network weights, we applied weight decay L_2 regularization on all the weights of the network and batch normalization on the output features as other regularization techniques for the stability of training[48]. We optimized the m.s.e. between the predicted and the ground-truth movement kinematics by using the Adam optimizer[73] to update the learnable parameters of the FENet. The learning rate α starts at $\alpha = 0.1$, which is divided by 2 every 10 epochs using a linear scheduler. The value for the drop-out has been set to 0.2 for all the layers. To avoid overfitting of the linear decoder, the batch size was set to be equal to the length of the input session, which is around 20 (s.d. ±3) times greater than the dimensionality of the FENet-generated features but can differ from session to session. We applied early stopping as another regularization technique, which avoids overfitting by stopping the training process if validation loss does not decrease after 20 epochs.

We conducted parameter sweeps using Bayesian optimization[74] on the FENet model to assess the importance and impact of each hyperparameter in the model's architecture. The results indicate a correlation between the R^2 values and the parameter values (Extended Data Fig. 8). Our sweeps based on using Bayesian optimization shows that the strides of the initial layers emerge as the most influential parameters, with smaller strides yielding higher performance. One possible interpretation of this phenomenon is that smaller strides enable convolutional kernels to encompass a broader range of patterns within the input. By contrast, larger strides restrict the coverage between successive kernel movements, leading to a reduced capacity for filters to learn diverse patterns[48]. In addition, we observed that the kernel size becomes more crucial in later layers compared with the initial layers. This suggests that the inputs to later layers summarize information from multiple samples in the preceding layers. Consequently, the network becomes more sensitive to kernel size when combining richer features with different kernel sizes, as these layers combine samples providing less abstract information than deeper layers.

Our training architecture assumes that the neural activity is informative of movement kinematics. As FENet is trained on single electrodes, to remove the noisy and non-informative electrodes during training, we trained FENet on the top 25, 50 and 75 electrodes with the highest cross-validated R^2 values after sorting the neural electrodes according to the R^2 values of the TCs with respect to the cursor movement kinematics (Extended Data Fig. 6a,b). According to our analysis, the top 50 electrodes out of 192 recorded electrodes per session were providing the highest averaged performance on the validation data and therefore were used during training of the FENet. In theory, pre-selecting electrodes based on TCs performance could bias results to favour TCs in comparisons. Despite this, TCs were consistently outperformed by both the FENet and the WTs, as shown in the closed- and open-loop results (Figs. 2 and 3). To ensure that there is no feature bias favouring well-tuned electrodes compared with the other electrodes, we divided the top 75 electrodes of each session into three equal groups based on the sorted cross-validated R^2 values: top 25, middle 25 (mid 25) and bottom 25 (down 25) electrodes. In each experiment, we train FENet on the top 25, mid 25 or down 25 electrodes of the training sessions separately. Then, we train and test the linear decoder when it operates on FENet features extracted from the top 25, mid 25 or down 25 electrodes of the test session. Extended Data Fig. 6b shows the cross-validated averaged R^2 of each experiment. The linear decoder operating on FENet trained on the top 25 electrodes of the training sessions achieves higher averaged R^2 when it is tested on the top 25, the mid 25 or the down 25 electrodes. Therefore, using informative features to train the FENet is an integral aspect of the training process.

During the inference, we froze the trained FENet, and to be consistent with the training, we applied electrode-specific PLSR[68] to the M FENet-generated features of each neural electrode to reduce the M features to K, in which $K \leq M$ (Extended Data Fig. 1b). We set $M = 8$ and $K = 2$ in our experiments according to the analysis on the number of partial least squares coefficients (PLSs) needed for regression (Extended Data Fig. 6d,e). PLSR maps the input features to a lower-dimensional space by defining an analytic linear transformation between its inputs and its lower-dimensional outputs, which maximizes the covariance between the neural data and the kinematics. Then, we trained an analytical linear decoder based on the top two PLS-generated neural features to minimize overfitting that can occur when too many predictor variables are used relative to the amount of training data.

To evaluate the impact of PLSR on the performance of the linear decoder operating on FENet, we conducted a rigorous analysis using data from all 54 sessions of participant JJ (Extended Data Fig. 3c,d). Our evaluation involved a comparison of FENet's performance with and without the application of PLSR, specifically applied to the top 40 electrodes within these sessions. The selection of these top 40 electrodes was motivated by the goal of mitigating potential overfitting issues that may arise in the linear decoder, particularly in scenarios where PLSR is not used. The results depicted in this figure provide compelling evidence demonstrating that FENet shows the ability to effectively capture informative features from the vast neural data, irrespective of the presence or absence of PLSR. In addition, the application of PLSR plays a vital role in reducing the dimensionality of the extracted features. This dimensionality reduction step is crucial as it helps prevent overfitting of the decoders, particularly when working with limited neural data obtained from human participants within each session. These findings highlight the robustness and efficacy of FENet as a feature extraction technique in neural decoding tasks. Furthermore, they underscore the importance of using dimensionality reduction methods such as PLSR, which can enhance the performance and generalizability of the linear decoder by mitigating the risk of overfitting when working with limited neural data.

To assess the significance of each extracted feature by FENet for every electrode, we used the Shapley value[45] as a measure of importance. The Shapley value allows us to determine the contribution of each input feature in the decoding process when using a linear decoder. The computation of the Shapley value involves comparing the decoder's output with and without the inclusion of a specific feature. The discrepancy between these two cases reflects the contribution of the feature to the decoding process. This calculation is repeated for all possible combinations of features per electrode, and the Shapley value for a given feature is determined by averaging these contributions across all possible combinations, considering the number of combinations

that include the feature. In this manner, we can evaluate the incremental contribution of each feature to the decoder's output while considering the interactions between features. Features with higher Shapley values are deemed more important as they make a greater contribution to the output variable compared with other features. Extended Data Fig. 7 presents the relative Shapley values of the eight FENet-extracted features. These values represent the average contribution of each feature to the decoding process, calculated and averaged across all electrodes and sessions, using offline data recorded from human participant JJ during the centre-out task. Extended Data Fig. 7 illustrates that the features extracted at the initial stage play a more crucial role in predicting outcomes through the linear decoder.

As an alternative to the PLSR for dimensionality reduction, to combine the dimensionality reduction technique with the feature extraction process, we have replaced the PLSR with a single fully connected layer as the last layer of the FENet, which maps $M = 8$ FENet-generated features to $K = 1$ feature per electrode (Extended Data Fig. 1c). Extended Data Fig. 1e,f compares the performance of the FENet when the dimensionality of the FENet convolutional filters outputs is reduced by using the PLSR or the added fully connected layer as the last layer of the FENet for dimensionality reduction. According to these figures, the performance of the decoder stays almost the same independent from these two dimensionality reduction techniques. Combining the feature extraction and the dimensionality reduction processes will make the usage of FENet architecture easier, while there is less control on the number of extracted features per electrode compared with using the PLSR for dimensionality reduction.

Decoders

To evaluate the performance of different feature extraction techniques, we passed them to different types of decoder, including the linear decoder[23,69], SVR[39–41], LSTM[42], Kalman filter[5,24] and PSID[43].

The linear decoder uses a standard linear regression model where we can predict kinematics ($\hat{y}$) from the extracted neural features (u) by using the following:

$$\hat{y} = b + \sum_{i=1}^{N} W_i u_i \tag{4}$$

We find the weights W_i and the bias term b through a least squares error optimization to minimize m.s.e. between the model's predictions and ground-truth kinematics during training. The parameters are then used to predict unseen kinematics given extracted neural features during the test.

SVR is the continuous form of support vector machines where the generalized error is minimized, given by the following function:

$$\hat{y} = \sum_{i=1}^{N} \left(\alpha_i^* - \alpha_i\right) k(u_i, u) + b \tag{5}$$

where α_i^* and α_i are Lagrange multipliers and k is a kernel function, where we use the radial basis function kernel in this paper. The Lagrange multipliers are found by minimizing a regularized risk function:

$$R_{\text{reg}}[f] = \frac{1}{2}||w||^2 + C\sum_{i=1}^{l} L_\varepsilon(y) \tag{6}$$

where $||w||^2$ represents the model complexity, and C is a constant that determines the trade-off between the ε insensitive loss function $L_\varepsilon(y)$. For SVR, we used a radial basis function kernel with C set to 1.

It is well known that simple RNN[75] units cannot remember long-term dependencies in sequential data because of the vanishing gradients problem. Another version of RNNs that is widely used in the literature are RNNs with LSTM units. By denoting $\circ$ as Hadamard product, the LSTM is defined as follows:

$$\begin{cases} f_k = \sigma(W_{fu}u_k + W_{fr}r_{k-1} + b_f) \\ i_k = \sigma(W_{iu}u_k + W_{ir}r_{k-1} + b_i) \\ o_k = \sigma(W_{ou}u_k + W_{or}r_{k-1} + b_i) \\ c_u = \tanh\left(W_{cu}u_k + W_{cr}r_{k-1} + b_c\right) \\ c_k = f_k \circ c_{k-1} + i_k \circ c_{k-1} \\ r_k = o_k \circ \tanh(c_k) \\ \hat{y}_k = W_{yr}r_k + b_y \end{cases} \tag{7}$$

r_k is the hidden state as in simple RNN; c_u is the output from the cell update activation function; c_k is the LSTM cell's internal state; f_k, i_k and o_k are the output matrices from the respective forget, input and output activation functions, which act as the LSTM's gates; W and b represent the weights and biases; and σ is the sigmoid function. Following our parameter sweeps, the settings for the number of layers, the number of recurrent nodes and the history of LSTM were determined as 1, 50 and 10, respectively.

The Kalman filter combines the idea that kinematics are functions of neural firings and the idea that neural activity is a function of movements, or the kinematics. This can be represented by two equations:

$$\begin{cases} \hat{y}_{k+1} = A_k\hat{y}_k + w_k \\ u_k = H_k\hat{y}_k + q_k \end{cases} \tag{8}$$

These represent how the system evolves over time as well as how neural activity is generated by the system's behaviour. The matrices A, H, Q and W can be found through a training process (where $q \sim \mathcal{N}(0, Q)$ and $w \sim \mathcal{N}(0, W)$. Using properties of the conditional probabilities of kinematics and neural data, we get a closed-form solution for maximizing the joint probability $p(Y_M, U_M)$. Using the physical properties of the problem, we get matrix A to be of the following form:

$$A = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & a & b \\ 0 & 0 & c & d \end{bmatrix} \tag{9}$$

where A_v is defined as follows:

$$A_v = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = V_2V_1^T(V_1V_1^T)^{-1} \tag{10}$$

V_1 consists of the velocity kinematics points except for the last time step, V_2 consists of the velocity kinematics points except for the first time step, and dt is the time step size used (in our case, 30 ms for JJ and NS, and 50 ms for EGS).

Furthermore, W is a zero matrix with the matrix $W_u = \frac{1}{N-1}(V_2 - AV_1)(V_2 - AV_1)^T$ in the bottom right corner. H and Q are given by the following:

$$\begin{cases} H = U^TY(YY^T)^{-1} \\ Q = \frac{1}{N}(U - HY)(U - HY)^{-1} \end{cases} \tag{11}$$

Then, we can use the following updated equations:

$$\begin{cases} \hat{y}_k^- = A\hat{y}_{k-1} \\ P_k^- = AP_{k-1}A^T + W \\ \hat{y}_k = \hat{y}_k^- + K_k(u_k - H\hat{y}_k^-) \\ P_k = (1 - K_kH)P_k^- \end{cases} \tag{12}$$

Here, P is the covariance matrix of the kinematics. K_k, the Kalman filter gain, is given by the following:

$$K_k = P_k^- H^T (H P_k^- H^T + Q)^{-1} \quad (13)$$

PSID models the brain's state as a high-dimensional latent variable influencing neural activity and behaviour. PSID, an algorithm built upon the Kalman filter equations, uses a dynamic linear state space model to describe the association between the latent state and the recorded neural activity (u_k) and behaviour (y_k). The model consists of a latent state $x_k \in \mathbb{R}^{n_x}$, which includes behaviourally relevant ($x_k^{(1)} \in \mathbb{R}^{n_1}$) and irrelevant ($x_k^{(2)} \in \mathbb{R}^{n_2}$) components as below:

$$\begin{cases} x_{k+1} = Ax_k + w_k \\ u_k = C_y x_k + v_k \\ y_k = C_z x_k + \varepsilon_k \end{cases}, \quad x_k = \begin{bmatrix} x_k^{(1)} \\ x_k^{(2)} \end{bmatrix} \quad (14)$$

PSID uses a two-stage identification approach. In the first stage, it directly learns the behaviourally relevant component ($x_k^{(1)}$) from training data without simultaneously learning the irrelevant component ($x_k^{(2)}$), which is optional in the second stage. This prioritization enables PSID to learn behaviourally relevant neural dynamics using low-dimensional states (only $x_k^{(1)}$). Similar to Kalman filter, the PSID model formulation includes noise terms (ε_k, w_k and v_k) representing behaviour dynamics that are not present in the recorded neural activity. The parameters of the model (A, C_y, C_z and noise statistics) are learned by PSID using training samples of neural activity and behaviour[43]. After the parameter sweep, we adjusted the latent space dimension to 10.

Open-loop evaluation measure

We reported the cross-validated coefficient of determination[76], R^2, as a measure of the strength of the linear association between the predicted and the ground-truth kinematics, respectively. The R_x^2 and R_y^2 have been computed independently in the x (horizontal) and y (vertical) dimensions using the definition of the coefficient of determination:

$$R^2 = \left(\frac{\sum_i (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_i (y_i - \bar{y})^2} \sqrt{\sum_i (\hat{y}_i - \bar{\hat{y}})^2}} \right)^2 \quad (15)$$

where y_i and $\hat{y}_i$ are the ith ground-truth and prediction, respectively. R^2 is a real number that varies from 0 to 1. The larger the R^2 is, the better the performance. We found that results are qualitatively the same when analysing each dimension separately. Then, we calculated the combined R^2 value for both x and y directions to be the norm of the $[R_x^2, R_y^2]$ vector as below:

$$R^2 = \frac{1}{\sqrt{2}} \sqrt{(R_x^2)^2 + (R_y^2)^2} \quad (16)$$

The maximum for R^2 occurs when the predictions and the ground-truth are completely matched, in which R_x^2 and R_y^2 are both equal to 1.

To assess the performance on the finger-grid task[49], we used the framework of representational similarity analysis[77,78] and representational dynamics analysis[79]. Representational similarity analysis quantifies the neural representational structure by measuring the pairwise distances between the neural activity patterns associated with each finger. These distances are used to construct the representational dissimilarity matrix, which provides a concise summary of the representational structure. It is worth noting that these distances are independent of the original feature types, such as electrode or voxel measurements, enabling us to compare finger organizations across participants and different recording modalities[80]. In addition, we used representational dynamics analysis to explore the temporal evolution of the representational structure. This involved modelling the representational structure of finger movements at each time point as a non-negative linear combination of potentially predictive models.

Single-electrode evaluation

To compare the improvement of the predictability of each single electrode using different feature extraction techniques, we directly trained three distinct linear decoders, one per each of FENet, TC and WT features that were extracted from each single electrode. Then, we predicted the movement kinematics for each of these three decoders corresponding to the mentioned three single-electrode features. Finally, we compared the cross-validated R^2 values of the predictions for each single neural electrode, and we repeated this process for all the other electrodes of 11 sample recording sessions for JJ. Extended Data Fig. 4a–c shows the R^2 value of linear decoder operating on FENet, WTs and TCs as the feature extraction technique with respect to each other, in a series of pair-wise comparisons. The blue dots represent the electrodes that have had low R^2 values in both feature extraction techniques, whereas the red dots represent the electrodes with the high R^2 values in at least one of the reported feature extraction techniques. FENet improved single-electrode R^2 values compared with the TCs (binomial test, $P = 0$) and the WTs (binomial test, $P = 4 \times 10^{-8}$).

To compare the preferred tuning direction of the FENet features per channel, we trained three distinct linear decoders, one for each feature extraction technique (FENet, TCs, WTs) per channel. Then, we calculated the phase and the magnitude difference between the corresponding tuning vectors for each pair of feature extraction techniques (Extended Data Fig. 4d–j). Although the feature extraction techniques are inherently different, the activity of a similar electrode maintains its preferred direction independent of a specific feature extraction technique.

Closed-loop evaluation measures

We have used several metrics to evaluate the closed-loop decoding performance: success rate as the number of correct trials completed within a fixed amount of time, time required for the cursor to reach the target, the path efficiency as measured by the ratio of path length to straight-line length, the instantaneous angular error that captures the angle between a vector pointing towards the target and the instantaneous velocity of the cursor, accuracy (how well the cursor tracks participant intentions) and blinded queries to research participants to evaluate responsiveness (how quickly the cursor responds to participant intentions) (Fig. 2). In addition, for the grid task, we have included the bit rate in our findings. The calculation of the bit rate is outlined below[7,81] (Supplementary Video 2):

$$B = \frac{\log_2(N) \times \max(S_c - S_i, 0)}{t} \quad (17)$$

where N is the number of total targets on the screen, S_c is the number of completed trials (correct selections), S_i is the number of incomplete trials (incorrect selections), and t is the time elapsed in seconds. Moreover, we have evaluated the computational overhead by tracking how much time is required to compute each prediction's update. With this array of metrics, we could build a more complete picture of the performance and computational consequences of our design choices and their impact on the participants' user experience and preference.

Closed-loop testing

The ability to test the FENet using neural recordings during development and operation with a human participant during test and validation is critical to the success of FENet. Our testing of the feature extraction techniques included both data-driven measurements of performance and quantitative and subjective feedback provided by

human research participants during our double-blind testing. We have used the double-blind testing to capture quantifiable and subjective performance metrics of the algorithms being tested for each of the feature types (TCs, WTs and FENet). In each session, these two feature extraction techniques (hereafter techniques A and B) were selected for evaluation. One batch consisted of an open-loop training run with 64 trials to parameterize A and B, a single closed-loop re-training run with 64 trials to re-train A and B decoders, and two closed-loop runs per algorithm each with 96 trials (four total closed-loop runs, with A and B shuffled). Each run lasted approximately 3–5 min, for a total of 15–25 min per batch. We performed two batches in each session with at least a 10 min break between and alternate the starting algorithm. The participant and researchers had been told which algorithm was being used ('A' or 'B') but not what A or B were. After each batch, we queried the participant to capture subjective experience and preference in each session.

Computational cost of FENet

To determine the computational complexity of various FENet architectures, we quantify the total count of multiplicative and additive operations performed for the feature extraction within the network. Assume that S_i, k_i and s_i are input size, kernel size and stride of the ith feature engineering module of FENet, respectively. The size of the input for the ith feature engineering module of FENet can be calculated as below[48]:

$$S_i = \left\lfloor \frac{S_{i-1} + \max(k_i - s_i, 0) + (k_i - 1) - k_i}{s_i} \right\rfloor \quad (18)$$

where $\max(k_i - s_i, 0)$ and $(k_i - 1)$ represent the left and right paddings, respectively. Then, we can calculate the cost for all the FENet layers as follows:

$$\text{Cost} = \sum_{i=0}^{n-1} 2k_i S_i \quad (19)$$

Given that n represents the quantity of feature engineering modules within the FENet, it is necessary to consider the dual cost incurred by both the upper and lower branches of these modules. As such, the computational cost is effectively doubled to encompass the collective operations of these components.

Algorithmic implementation requirements

We used PyTorch (version 1.14.0), a deep-learning application programming interface for Python (version 3.10)[82], as the programmatical framework to train and operate neural networks. We configured PyTorch to use CUDA, a parallel computing platform and programming model developed by NVIDIA, which can accelerate many of the computations involved in training neural networks with commercially available graphics processing units (GPU)[83]. For offline training and evaluation of the FENet, we used a single Tesla V100 GPU[83], and for the closed-loop runs, we used a single NVIDIA GeForce RTX 3080 GPU[83].

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The behavioural and neurophysiological data are archived in the Division of Biology and Biological Engineering at the California Institute of Technology. The broadband neural data are confidential and hence cannot be publicly shared. The raw and analysed data generated during the study are available for research purposes from the corresponding authors on reasonable request.

Code availability

The codes used for training and inference of FENet are available via GitHub at https://github.com/BenyaminHaghi/FENet. Codes used for analysing and displaying the results presented in this study are available from the corresponding authors on reasonable request.

References

1. Musallam, S., Corneil, B. D., Greger, B., Scherberger, H. & Andersen, R. A. Cognitive control signals for neural prosthetics. *Science* **305**, 258–262 (2004).
2. Jackson, A., Moritz, C. T., Mavoori, J., Lucas, T. H. & Fetz, E. E. The Neurochip BCI: towards a neural prosthesis for upper limb function. *IEEE Trans. Neural Syst. Rehabil. Eng.* **14**, 187–190 (2006).
3. Moritz, C. T., Perlmutter, S. I. & Fetz, E. E. Direct control of paralysed muscles by cortical neurons. *Nature* **456**, 639–642 (2008).
4. Gilja, V. et al. Clinical translation of a high-performance neural prosthesis. *Nat. Med.* **21**, 1142–1145 (2015).
5. Gilja, V. et al. A high-performance neural prosthesis enabled by control algorithm design. *Nat. Neurosci.* **15**, 1752–1757 (2012).
6. Jarosiewicz, B. et al. Virtual typing by people with tetraplegia using a self-calibrating intracortical brain-computer interface. *Sci. Transl. Med.* **7**, 313ra179 (2015).
7. Nuyujukian, P., Fan, J. M., Kao, J. C., Ryu, S. I. & Shenoy, K. V. A high-performance keyboard neural prosthesis enabled by task optimization. *IEEE Trans. Biomed. Eng.* **62**, 21–29 (2015).
8. Aflalo, T. et al. Decoding motor imagery from the posterior parietal cortex of a tetraplegic human. *Science* **348**, 906–910 (2015).
9. Hochberg, L. R. et al. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature* **442**, 164–171 (2006).
10. Golub, M. D., Yu, B. M., Schwartz, A. B. & Chase, S. M. Motor cortical control of movement speed with implications for brain-machine interface control. *J. Neurophysiol.* **112**, 411–429 (2014).
11. Inoue, Y., Mao, H., Suway, S. B., Orellana, J. & Schwartz, A. B. Decoding arm speed during reaching. *Nat. Commun.* **9**, 5243 (2018).
12. Ben Hamed, S., Schieber, M. H. & Pouget, A. Decoding M1 neurons during multiple finger movements. *J. Neurophysiol.* **98**, 327–333 (2007).
13. Churchland, M. M. & Shenoy, K. V. Temporal complexity and heterogeneity of single-neuron activity in premotor and motor cortex. *J. Neurophysiol.* **97**, 4235–4257 (2007).
14. Aflalo, T. N. & Graziano, M. S. Relationship between unconstrained arm movements and single-neuron firing in the macaque motor cortex. *J. Neurosci.* **27**, 2760–2780 (2007).
15. Aflalo, T. N. & Graziano, M. S. Partial tuning of motor cortex neurons to final posture in a free-moving paradigm. *Proc. Natl Acad. Sci. USA* **103**, 2909–2914 (2006).
16. Goodman, J. M. et al. Postural representations of the hand in the primate sensorimotor cortex. *Neuron* https://doi.org/10.1016/j.neuron.2019.09.004 (2019).
17. Degenhart, A. D. et al. Stabilization of a brain–computer interface via the alignment of low-dimensional spaces of neural activity. *Nat. Biomed. Eng.* **4**, 672–685 (2020).
18. Gallego, J. A., Perich, M. G., Chowdhury, R. H., Solla, S. A. & Miller, L. E. Long-term stability of cortical population dynamics underlying consistent behavior. *Nat. Neurosci.* **23**, 260–270 (2020).
19. Allahgholizadeh Haghi, B. et al. Deep multi-state dynamic recurrent neural networks operating on wavelet based neural features for robust brain machine interfaces. *Adv. Neural Inf. Process. Syst.* **32**, 14514–14525 (2019).

20. Zhang, M. et al. Extracting wavelet based neural features from human intracortical recordings for neuroprosthetics applications. *Bioelectron. Med.* **4**, 11 (2018).
21. Christie, B. P. et al. Comparison of spike sorting and thresholding of voltage waveforms for intracortical brain-machine interface performance. *J. Neural Eng.* **12**, 016009 (2015).
22. Pandarinath, C. et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nat. Methods* **15**, 805–815 (2018).
23. Wu, W. & Hatsopoulos, N. G. Real-time decoding of nonstationary neural activity in motor cortex. *IEEE Trans. Neural Syst. Rehabil. Eng.* **16**, 213–222 (2008).
24. Orsborn, A. L. et al. Closed-loop decoder adaptation shapes neural plasticity for skillful neuroprosthetic control. *Neuron* **82**, 1380–1393 (2014).
25. Wessberg, J. et al. Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature* **408**, 361–365 (2000).
26. Sussillo, D., Stavisky, S. D., Kao, J. C., Ryu, S. I. & Shenoy, K. V. Making brain–machine interfaces robust to future neural variability. *Nat. Commun.* **7**, 13749 (2016).
27. Sussillo, D. et al. A recurrent neural network for closed-loop intracortical brain–machine interface decoders. *J. Neural Eng.* **9**, 026027 (2012).
28. Shah, S. et al. Decoding kinematics from human parietal cortex using neural networks. In *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)* 1138–1141 (IEEE, 2019); https://doi.org/10.1109/NER.2019.8717137
29. Schwemmer, M. A. et al. Meeting brain–computer interface user performance expectations using a deep neural network decoding framework. *Nat. Med.* **24**, 1669–1676 (2018).
30. Willsey, M. S. et al. Real-time brain-machine interface in non-human primates achieves high-velocity prosthetic finger movements using a shallow feedforward neural network decoder. *Nat. Commun.* **13**, 6899 (2022).
31. Willett, F. R., Avansino, D. T., Hochberg, L. R., Henderson, J. M. & Shenoy, K. V. High-performance brain-to-text communication via handwriting. *Nature* **593**, 249–254 (2021).
32. Sadtler, P. T., Ryu, S. I., Tyler-Kabara, E. C., Yu, B. M. & Batista, A. P. Brain-computer interface control along instructed paths. *J. Neural Eng.* **12**, 016015 (2015).
33. Young, D. et al. Closed-loop cortical control of virtual reach and posture using Cartesian and joint velocity commands. *J. Neural Eng.* **16**, 026011 (2019).
34. Pandarinath, C. et al. High performance communication by people with paralysis using an intracortical brain-computer interface. *eLife* **6**, e18554 (2017).
35. Fan, J. M. et al. Intention estimation in brain-machine interfaces. *J. Neural Eng.* **11**, 016004 (2014).
36. Martinez, J., Pedreira, C., Ison, M. J. & Quian Quiroga, R. Realistic simulation of extracellular recordings. *J. Neurosci. Methods* **184**, 285–293 (2009).
37. Barrese, J. C. et al. Failure mode analysis of silicon-based intracortical microelectrode arrays in non-human primates. *J. Neural Eng.* **10**, 066014 (2013).
38. Stark, E. & Abeles, M. Predicting movement from multiunit activity. *J. Neurosci.* **27**, 8387–8394 (2007).
39. Awad, M. & Khanna, R. in *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers* (eds Awad, M. & Khanna, R.) 67–80 (Apress, 2015); https://doi.org/10.1007/978-1-4302-5990-9_4
40. Basak, D., Pal, S. & Patranabis, D. Support vector regression. *Neural Inf. Process. Lett. Rev.* **11**, 203–224 (2007).
41. Smola, A. J. & Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **14**, 199–222 (2004).
42. Gers, F. A., Schmidhuber, J. & Cummins, F. Learning to forget: continual prediction with LSTM. *Neural Comput.* **12**, 2451–2471 (2000).
43. Sani, O. G., Abbaspourazad, H., Wong, Y. T., Pesaran, B. & Shanechi, M. M. Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification. *Nat. Neurosci.* **24**, 140–149 (2021).
44. Cabitza, F., Rasoini, R. & Gensini, G. F. Unintended consequences of machine learning in medicine. *JAMA* **318**, 517–518 (2017).
45. Lundberg, S. M. & Lee, S.-I. A unified approach to interpreting model predictions. In *Proc. 31st International Conference on Neural Information Processing Systems* 4766–4777 (Curran Associates, 2017).
46. Zhang, Y., Tiňo, P., Leonardis, A. & Tang, K. A survey on neural network interpretability. *IEEE Trans. Emerg. Top. Comput. Intell.* **5**, 726–742 (2021).
47. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
48. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* (MIT Press, 2016); https://www.deeplearningbook.org/
49. Guan, C. et al. Stability of motor representations after paralysis. *eLife* **11**, e74478 (2022).
50. Ejaz, N., Hamada, M. & Diedrichsen, J. Hand use predicts the structure of representations in sensorimotor cortex. *Nat. Neurosci.* **18**, 1034–1040 (2015).
51. Cogan, S. F. Neural stimulation and recording electrodes. *Annu. Rev. Biomed. Eng.* **10**, 275–309 (2008).
52. Chestek, C. A. et al. Long-term stability of neural prosthetic control signals from silicon cortical arrays in rhesus macaque motor cortex. *J. Neural Eng.* **8**, 045005 (2011).
53. Downey, J. E., Schwed, N., Chase, S. M., Schwartz, A. B. & Collinger, J. L. Intracortical recording stability in human brain-computer interface users. *J. Neural Eng.* **15**, 046016 (2018).
54. Nason, S. R. et al. A low-power band of neuronal spiking activity dominated by local single units improves the performance of brain–machine interfaces. *Nat. Biomed. Eng.* **4**, 973–983 (2020).
55. Trautmann, E. M. et al. Accurate estimation of neural population dynamics without spike sorting. *Neuron* **103**, 292–308.e4 (2019).
56. Gold, C., Henze, D. A., Koch, C. & Buzsáki, G. On the origin of the extracellular action potential waveform: a modeling study. *J. Neurophysiol.* **95**, 3113–3128 (2006).
57. Stringer, C. et al. Spontaneous behaviors drive multidimensional, brainwide activity. *Science* **364**, eaav7893 (2019).
58. Masse, N. Y. et al. Non-causal spike filtering improves decoding of movement intention for intracortical BCIs. *J. Neurosci. Methods* **236**, 58–67 (2014).
59. Guan, C. et al. Decoding and geometry of ten finger movements in human posterior parietal cortex and motor cortex. *J. Neural Eng.* **20**, 036020 (2023).
60. Klaes, C. et al. Hand shape representations in the human posterior parietal cortex. *J. Neurosci.* **35**, 15466–15476 (2015).
61. Aflalo, T. et al. A shared neural substrate for action verbs and observed actions in human posterior parietal cortex. *Sci. Adv.* **6**, eabb3984 (2020).
62. Jafari, M. et al. The human primary somatosensory cortex encodes imagined movement in the absence of sensory information. *Commun. Biol.* **3**, 757 (2020).
63. Andersen, R. A., Aflalo, T. & Kellis, S. From thought to action: the brain–machine interface in posterior parietal cortex. *Proc. Natl Acad. Sci. USA* **116**, 26274–26279 (2019).
64. Orban, G. A. & Caruana, F. The neural basis of human tool use. *Front. Psychol.* **5**, 310 (2014).
65. Gallivan, J. P. & Culham, J. C. Neural coding within human brain areas involved in actions. *Curr. Opin. Neurobiol.* **33**, 141–149 (2015).

66. Ludwig, K. A. et al. Using a common average reference to improve cortical neuron recordings from microelectrode arrays. *J. Neurophysiol.* **101**, 1679–1689 (2009).
67. Willett, F. R. et al. Principled BCI decoder design and parameter selection using a feedback control model. *Sci. Rep.* **9**, 8881 (2019).
68. Geladi, P. & Kowalski, B. R. Partial least-squares regression: a tutorial. *Anal. Chim. Acta* **185**, 1–17 (1986).
69. Montgomery, D. C., Peck, E. A. & Vining, G. G. *Introduction to Linear Regression Analysis* (John Wiley & Sons, 1992).
70. Mulliken, G. H., Musallam, S. & Andersen, R. A. Decoding trajectories from posterior parietal cortex ensembles. *J. Neurosci.* **28**, 12913–12926 (2008).
71. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *Nature* **323**, 533–536 (1986).
72. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014).
73. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. Preprint at https://arxiv.org/abs/1412.6980 (2017).
74. Snoek, J., Larochelle, H. & Adams, R. P. Practical Bayesian optimization of machine learning algorithms. *Adv. Neural Inf. Process. Syst.* **25**, 2–6 (2012).
75. Mandic, D. P. & Chambers, J. *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability* (John Wiley & Sons, 2001).
76. Rao, C. R. *Linear Statistical Inference and Its Applications* (Wiley-Interscience, 2001).
77. Kriegeskorte, N., Mur, M. & Bandettini, P. Representational similarity analysis - connecting the branches of systems neuroscience. *Front. Syst. Neurosci.* **2**, 4 (2008).
78. Diedrichsen, J. & Kriegeskorte, N. Representational models: a common framework for understanding encoding, pattern-component, and representational-similarity analysis. *PLoS Comput. Biol.* **13**, e1005508 (2017).
79. Kietzmann, T. C. et al. Recurrence is required to capture the representational dynamics of the human visual system. *Proc. Natl Acad. Sci. USA* **116**, 21854–21863 (2019).
80. Kriegeskorte, N. et al. Matching categorical object representations in inferior temporal cortex of man and monkey. *Neuron* **60**, 1126–1141 (2008).
81. Yuan, P. et al. A study of the existing problems of estimating the information transfer rate in online brain–computer interfaces. *J. Neural Eng.* **10**, 026014 (2013).
82. Paszke, A. et al. PyTorch: an imperative style, high-performance deep learning library. in *Advances in Neural Information Processing Systems* vol. 32 (Curran Associates, 2019).
83. Ryoo, S. et al. Optimization principles and application performance evaluation of a multithreaded GPU using CUDA. In *Proc. 13th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP '08)* 73–82 (ACM Press, 2008); https://doi.org/10.1145/1345206.1345220
84. Chambers, J. M. & Hastie, T. J. *Statistical Models in S* (Routledge, 1992).
85. Cleveland, W. S., Grosse, E., & Shyu, W. M. *Local Regression Models* (Routledge, 1992).

Acknowledgements

We thank K. Pejsa for her help in running recording sessions and JJ, EGS and NS for accepting to participate in this research. We also thank D. Tang for his contributions to online implementation and stability testing of FENet. Funding for this Institutional Review Board- and FDA-approved work has been provided by the National Institute of Health grant R01EY015545 (R.A.A., T.A., N.P.), Tianqiao and Chrissy Chen Brain–machine Interface Center at Caltech (R.A.A., T.A.) and Boswell Foundation (R.A.A.). B.H. and A.E. were supported by the Center for Sensing to Intelligence (S2I), Braun Foundation, Chen Institute for Neuroscience and Heritage Medical Research Institute.

Author contributions

B.H. and T.A. developed FENet, analysed the results and wrote the paper. T.A. conceptualized the study and provided mentorship. B.H. implemented FENet. B.H., T.A., S.K. and C.G. contributed to experimental design and analysis. J.A.G.d.L. contributed to data recording. A.Y.H. contributed to hyperparameter optimization of FENet. N.P. conducted surgery on human participants and was responsible for participants' care. R.A.A. and A.E. provided mentorship and supervised the research. All authors reviewed and modified the paper.

Competing interests

The authors declare no competing interests.

Additional information

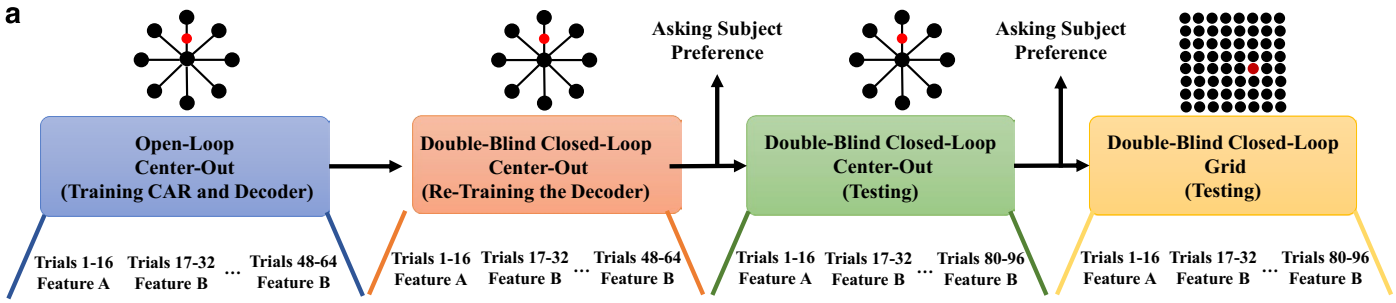
Extended data is available for this paper at https://doi.org/10.1038/s41551-024-01297-1.

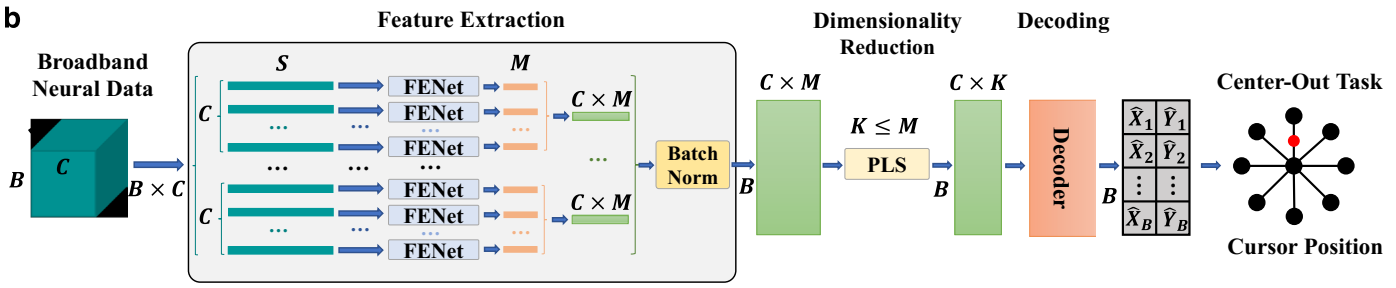
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41551-024-01297-1.

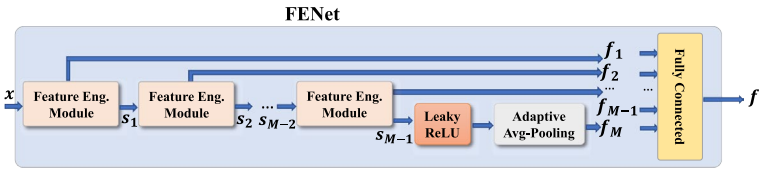
Correspondence and requests for materials should be addressed to Benyamin Haghi, Tyson Aflalo or Azita Emami.

Peer review information *Nature Biomedical Engineering* thanks Reinhold Scherer and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Peer reviewer reports are available.

https://doi.org/10.1038/s41551-024-01297-1

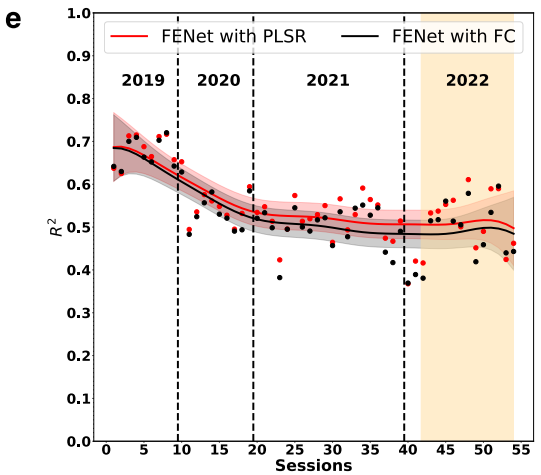


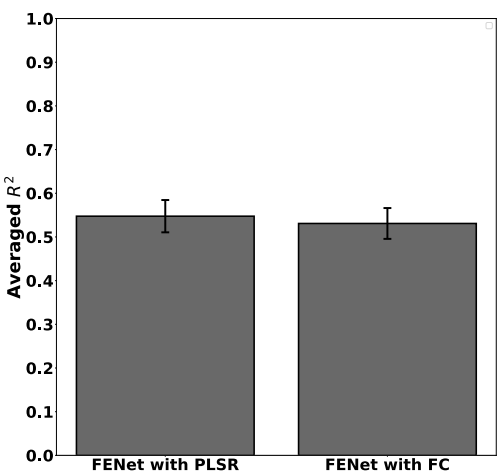




d

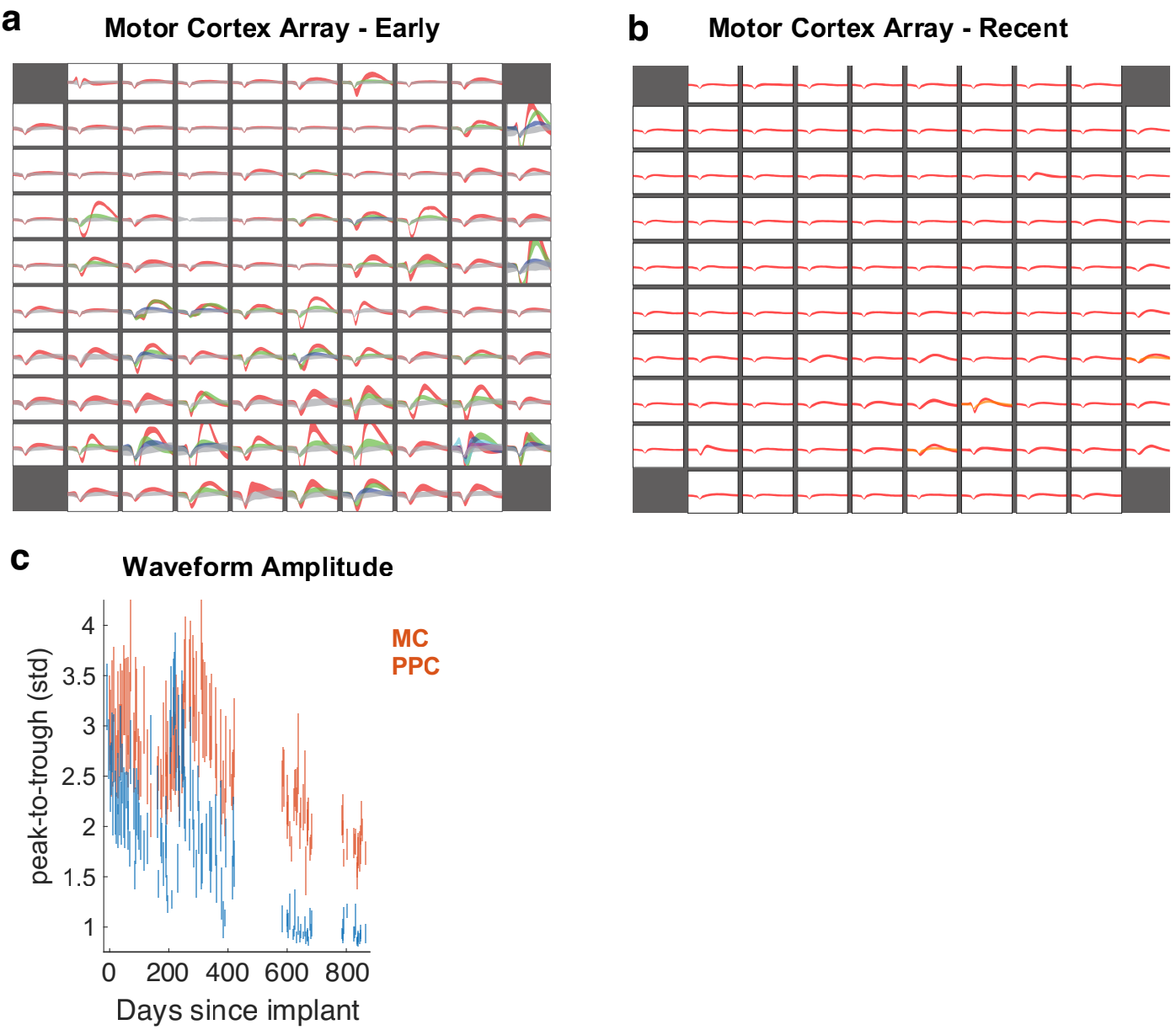
Table 1. FENet parameters used for the BMI experiments	
Number of convolutional layers	7
Convolutional filters size per layer	40
Zero-padding length for each module	39
Convolution filters stride	2
Convolution filters number of input channels	1
Convolution filters number of output channels	1
Learning rate (α)	0.1
Number of parameters	560





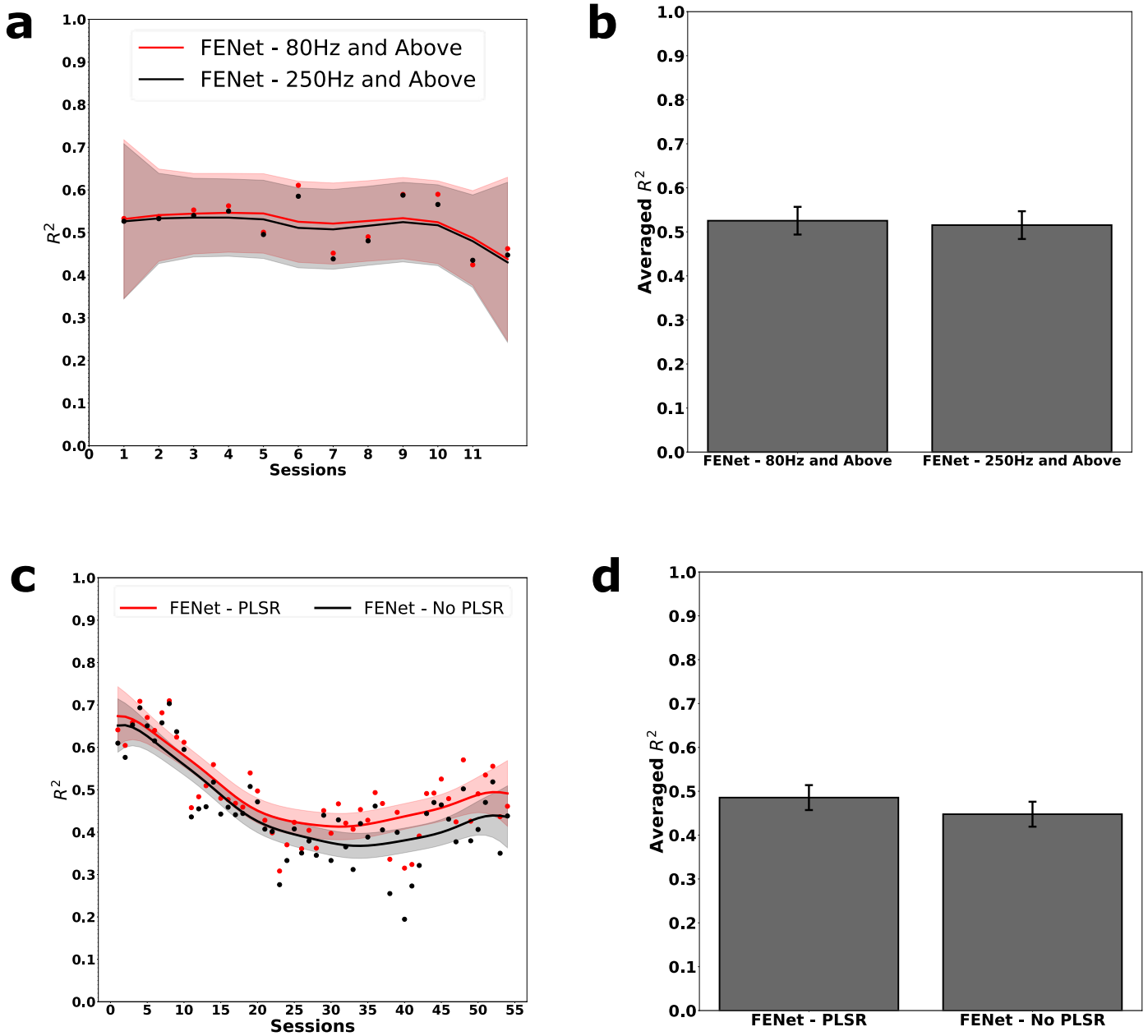
Extended Data Fig. 1 | Closed-loop training and pipeline, and architecture and performance of the BCI system. (**a**) Closed-loop training and pipeline. First, we record a block of open-loop centre-out data that includes 64 trials to train PCA for common average referencing and the decoders. Then, we re-train the decoders by recording a block of closed-loop centre-out data for the total of 64 trials. Next, we record a block of double-blind closed-loop centre-out data (Supplemental Video 1) and a block of double-blind closed-loop grid data (Supplemental Video 2) for closed-loop control performance evaluation, which each includes 96 trials. We switch between the feature extraction techniques A and B every 16 trials for all steps. In the middle, we ask the participant's preference regarding the feature extraction techniques A and B. (**b**) The architecture of the BCI system includes the input broadband neural data, feature extractor, decoder, and the output. (**c**) Adding an internal fully connected layer as the last layer of the FENet instead of applying PLSR for dimensionality reduction. (**d**) Parameter of the FENet. (**e**) The performance and (**f**) the average performance of the FENet for JJ when partial least square (PLSR) or a fully connected layer (FC) are used for the feature dimensionality reduction. Shaded region shows the closed-loop sessions. The band in each time series shows the range of 95% confidence interval of a LOESS[84,85] fit.

https://doi.org/10.1038/s41551-024-01297-1



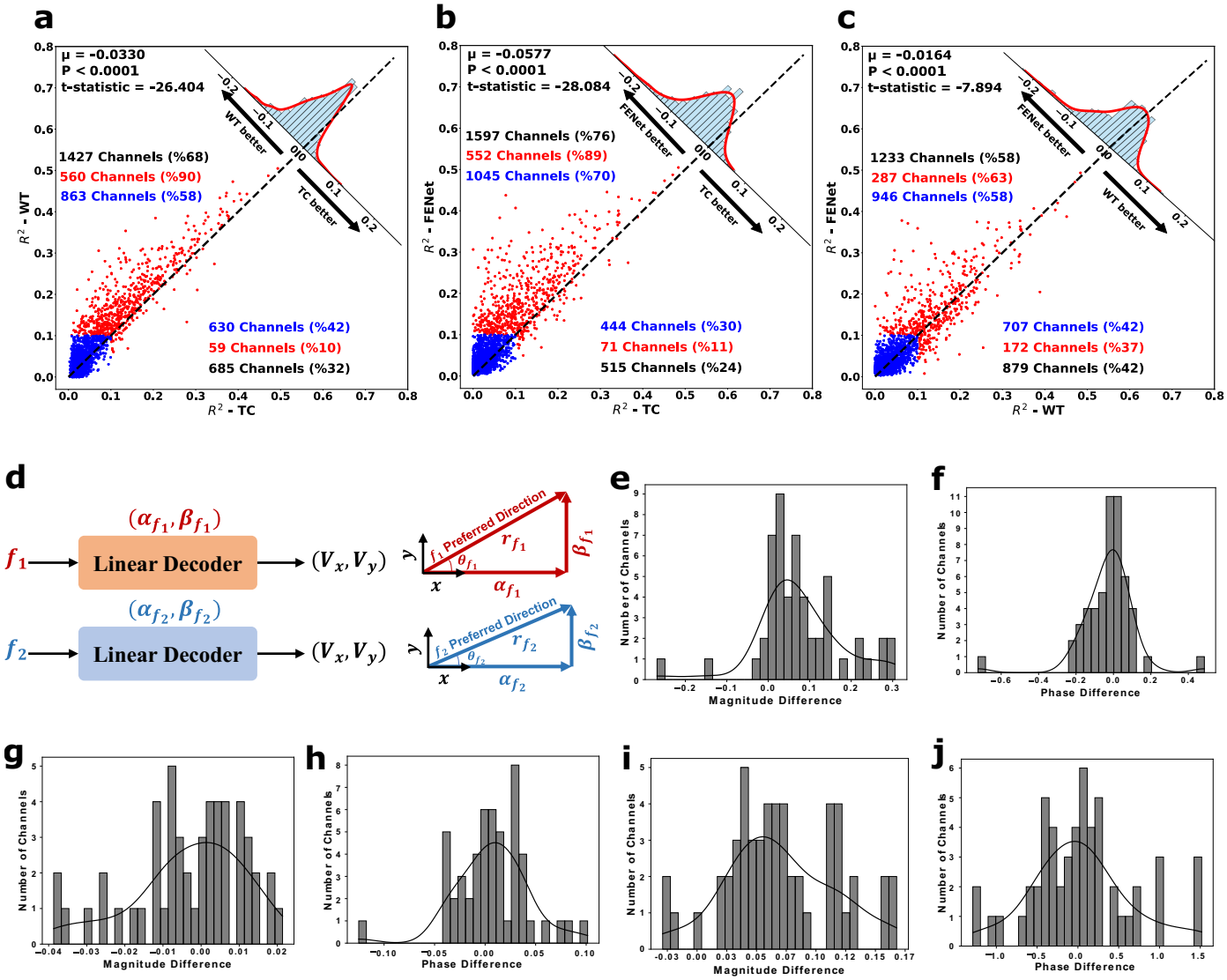
Extended Data Fig. 2 | Drop in performance correlated with degradation of the recording quality for JJ. (**a**) Sorted neural waveforms for each electrode on the recording array shortly after implantation in 2019. (**b**) Similar to a, but taken after several years of implantation in 2022. (**c**) Illustration of how the peak-to-trough amplitude of extracted waveforms decreases over lifetime of the array.

https://doi.org/10.1038/s41551-024-01297-1



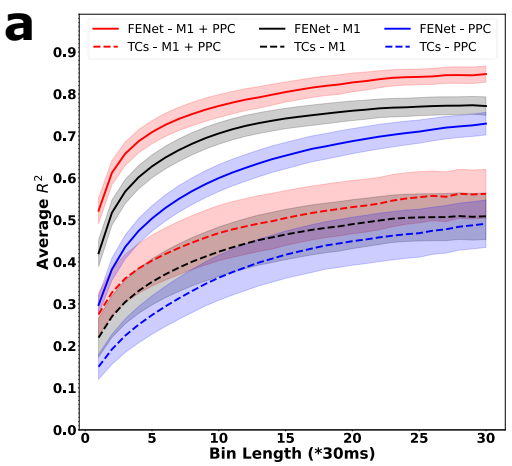
Extended Data Fig. 3 | FENet does not rely on the low-frequency (< 250 Hz) local-field potentials to achieve its enhanced decode performance. To examine if FENet is using local field potential (LFP) for its long-term stability, we filtered the broadband data recorded from the closed-loop sessions before extracting the FENet features by using the high pass filters with the cut-off frequency of 80 Hz and 250 Hz, respectively. We have used an 80 Hz filter since window size of 30 ms used for JJ is small enough to assume that the lower frequency activities are excluded from the broadband neural activity in the 30 ms window. Moreover, to mitigate potential residual 60 Hz noise, we established a lower cutoff frequency of 80 Hz. (**a**) the performance of linear decoder operating on FENet slightly drops per session and (**b**) on average when we filter data using a high-pass filter with the cut-off frequency of 250 Hz compared to the case that the cut-off frequency of the high-pass filter is 80 Hz, which shows FENet is not directly affected by the information that is extracted from the LFP band. (**c**, **d**) we conducted a comprehensive evaluation of the effect of PLSR on the performance of linear decoder operating on FENet's using all 54 sessions of participant JJ. We compared FENet's performance with and without Partial Least Squares Regression (PLSR) applied to the top 40 electrodes in these sessions. We selected the top 40 electrodes to mitigate overfitting in the linear decoder, particularly in cases where PLSR is not applied. The results presented in this figure demonstrate that FENet, regardless of PLSR, effectively captures informative features from the broad neural data. The application of PLSR serves to reduce feature dimensionality and prevent overfitting of the decoders when working with limited neural data from human participants per session. The band in each time series shows the range of its 95% confidence interval of a LOESS[84,85] fit.

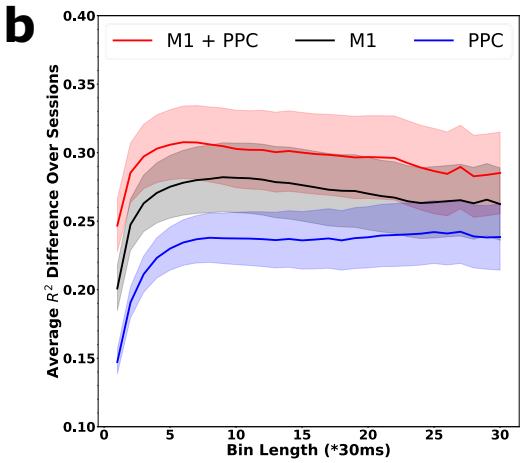
https://doi.org/10.1038/s41551-024-01297-1

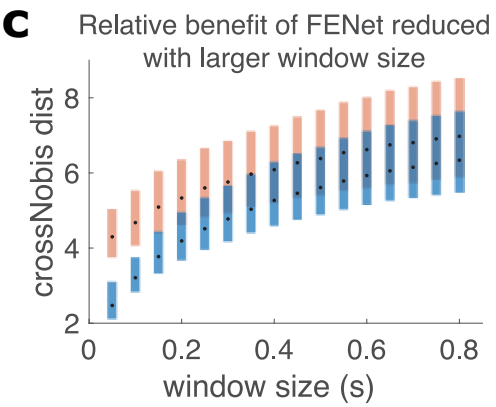


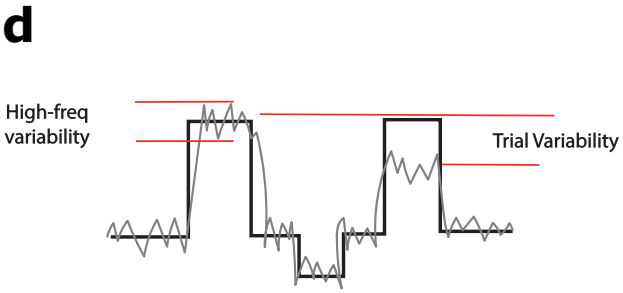
Extended Data Fig. 4 | Open-loop single-electrode performance of linear decoder operating on FENet, WTs, and TCs. (**a**-**c**) Comparison of the cross-validated R^2 of linear decoder for FENet, WTs, and TCs as different feature extraction techniques on all 192 neural channels (electrodes) of participant JJ's 2019 sessions. The dashed line shows line y = x. The red dots show the electrodes with R^2 greater than 0.1 in at least one of the feature extraction techniques and the blue dots are the electrodes with R^2 smaller than 0.1 for both techniques. Analysis is performed on red electrodes that carry more information about movement kinematics. The reported black, red, and blue numbers demonstrate the percentage of electrodes in each side of y = x for all the dots, red dots, and blue dots, respectively. The percentage of dots on each side of the line y = x shows the number of electrodes in favor of the corresponding feature extraction technique. The t-test statistics have also been reported to show the confidence level of the reported statistics. According to this analysis, FENet-based features improve the decoding performance of each single electrode in term of R^2 compared to TCs and WTs. (**d**) To compare the preferred direction and tuning properties of the same electrode in two feature extraction technique, we trained a linear decoder on feature that is extracted from that similar electrode for each feature, and we have reported the magnitude and angle difference between the vectors that are generated by the coefficients of the trained linear decoders. (**e**-**j**) Comparison of the single electrodes tuning properties. We compared the parameters of linear tuning models for same electrode between two features. Although the feature extraction techniques are inherently different, activity of a similar electrode maintains its preferred direction independent from a specific feature extraction technique. The phase difference is shown in radian. (**e**, **f**) FENet vs. TCs, (**g**, **h**) FENet vs. WTs, (**i**, **j**) WTs vs. TCs.

https://doi.org/10.1038/s41551-024-01297-1



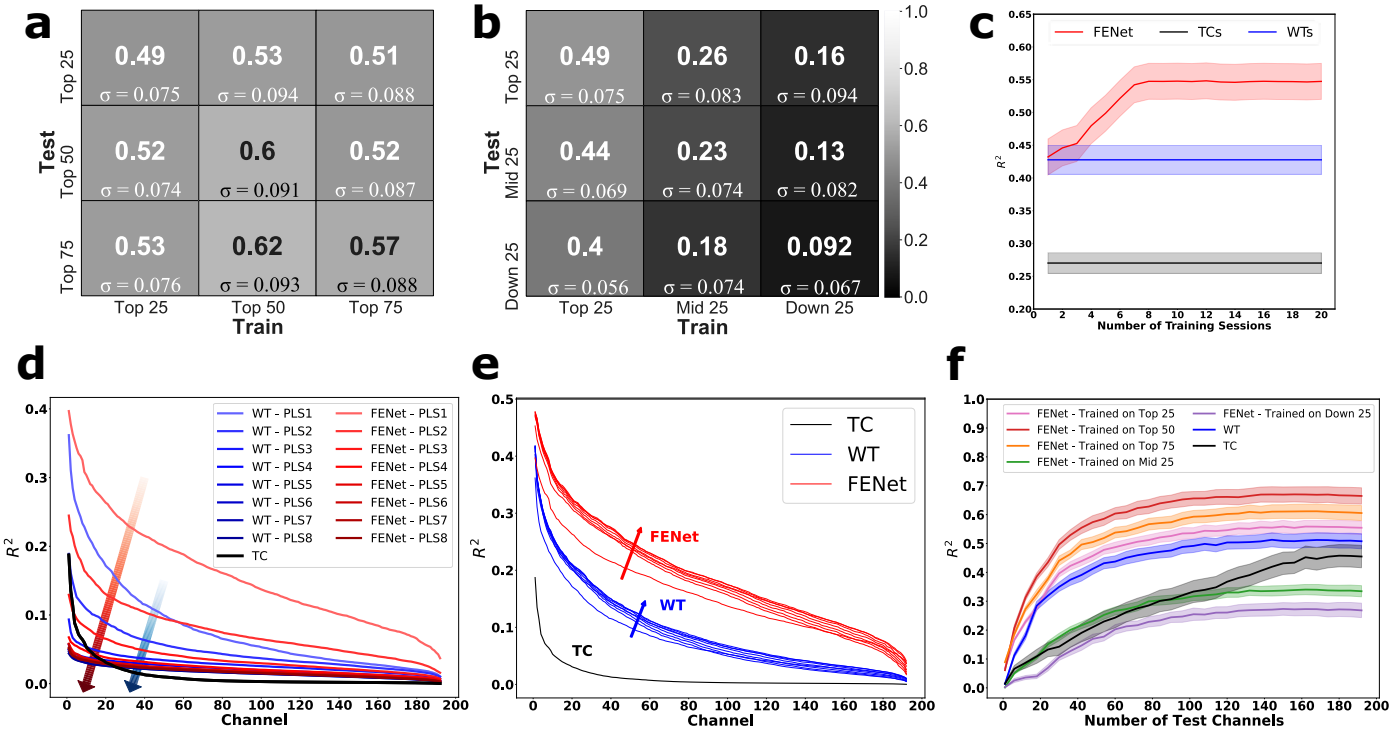






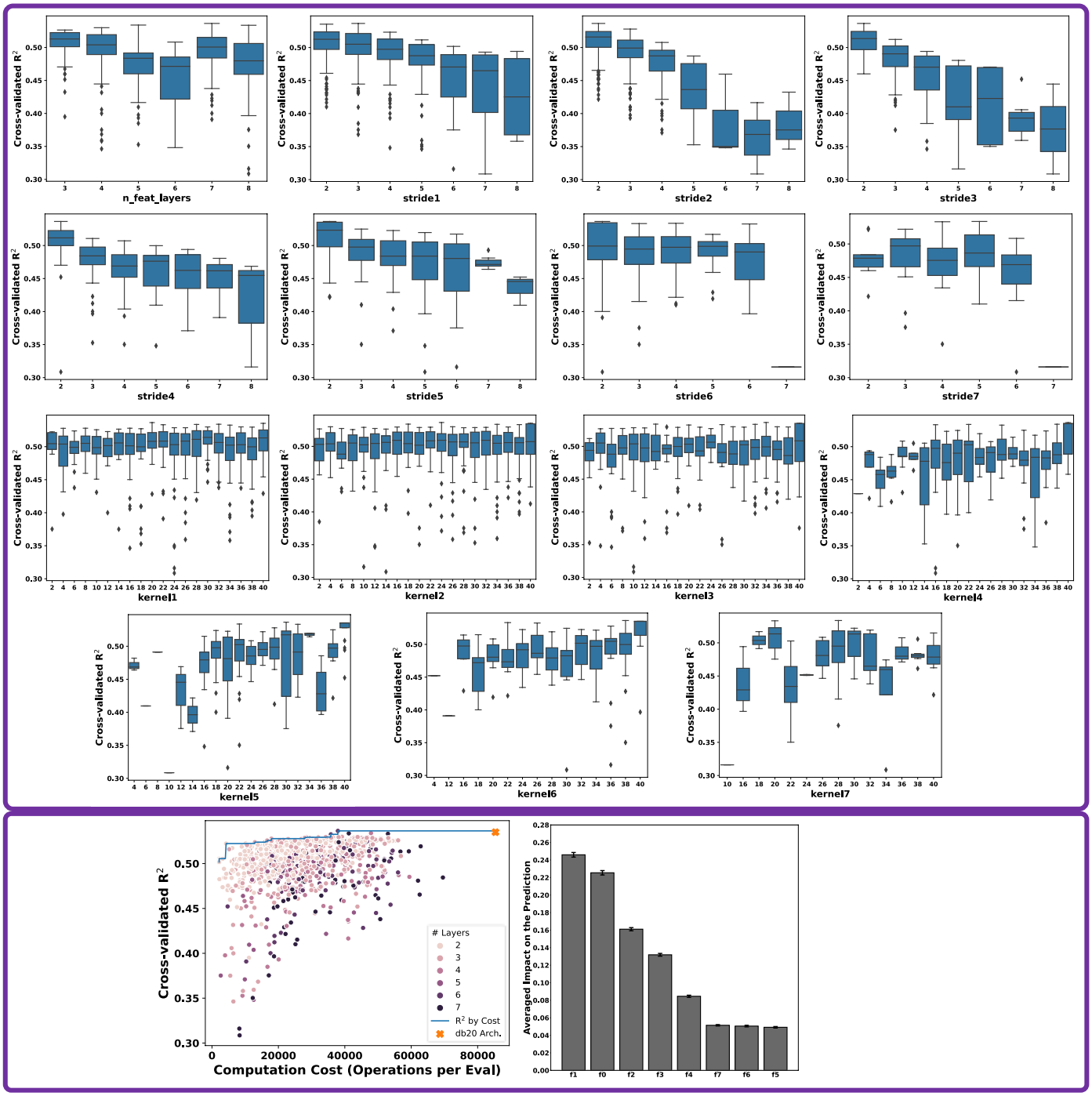
Extended Data Fig. 5 | The impact of modifying window sizes on decoder performance subsequent to averaging the extracted features across larger temporal windows. FENet performance is robust against the change of the recording window size length in the centre-out trajectory task. (**a**) The averaged R^2 of a linear decoder operating on FENet and TCs when we increase the feature extraction window size, which has a smoothing effect on the extracted features. The solid curves show the performances for FENet, and the dashed curves show the performance for the TCs. The blue curves, the red curves, and the black curves show the performance of the feature extraction techniques on the neural features extracted from the neural data recorded from all the electrodes, electrodes of M1, and electrodes of PPC. FENet maintains its superior performance over TCs when we increase the feature extraction window size in our trajectory task. (**b**) The average R^2 difference between the curves in fig. a. The band in each time series shows the range of its 95% confidence interval of a LOESS[84,85] fit. (**c**) here we measure how our crossNobis distance metric compared between sorted neurons and FENet as a function of window size. At small window sizes (for example, 50 ms) we see comparable benefits of FENet over sorted units. However, as the size the window increases, the relative benefit of the FENet is reduced. (**d**) The explanation of the high frequency and the between trial variability of the kinematic prediction. The black curve shows the ground-truth movement kinematics, and the gray curve shows the decoder prediction.

https://doi.org/10.1038/s41551-024-01297-1



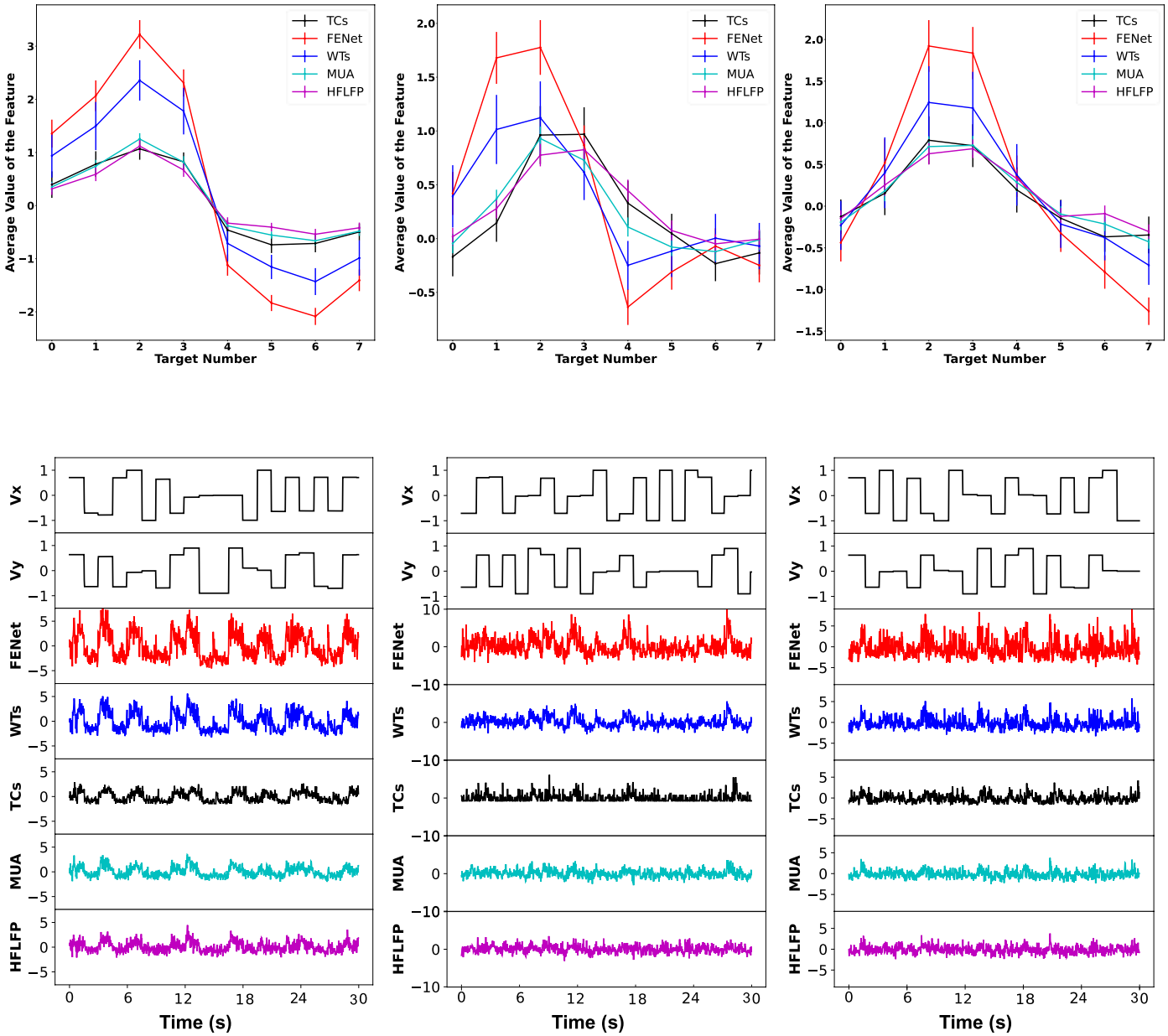
Extended Data Fig. 6 | Training and testing FENet. Training and testing FENet on (**a**) top 25, top 50, or top 75 electrodes, (**b**) on top 25, mid 50, or down 75 electrodes. These figures show that features generated by a FENet optimized on the 50 top electrodes have higher averaged performance and is more generalizable to the electrodes that were excluded from training compared to the other feature extraction techniques and parameters. (**c**) Number of sessions needed to train the FENet. We have changed the number of FENet training sessions from 1 to 10 for each left-out test session. We pick these training sessions from all the available training sessions randomly and repeat this process 10 times for each left-out test session to report the cross-validated performance. This figure shows that the performance of the linear decoder saturates by using about 7 sessions for training. Shaded regions show the 95% confidence intervals. (**d**) The single and (**e**) the cumulative PLSR generated features performance. The direction of arrows in (**d**) show the decrease in the performance of the linear decoder moving from best to worst PLS extracted features for FENet (red) and WTs (blue) compared to TCs (black). The direction of arrows in (**e**) show the increase in performance as more PLS features are included per electrode for decoding. To pick the optimum number of features per electrode for FENet and WTs, we compare the 10-fold cross-validated R^2 of single-electrode TCs, FENet, and WTs using different number of output features. Results are shown separately for each PLS-based latent dimension after sorting the electrodes by maximum per-session R^2, and then averaging across the sessions. Electrodes were sorted based on the R^2 value between the ground-truth and the linearly regressed movement kinematics using each single electrode. We have also shown the performance of a linear decoder operating on single electrodes' cumulative PLS features, starting from the best PLS feature (for example, PLS feature 1, 1&2, 1&2&3, and so on). This figure shows that top two WTs and FENet PLS features are enough for the linear decoder to reach approximately maximum performance. We take advantage of this finding by limiting our features to the top two PLS features for our population-based reconstructions of movement kinematics. Limiting the number of features prevents an explosion of predictive features that can result in overfitting and poor generalization. (**f**) Neuron dropping curves. To generate neuron dropping curves, we randomly picked a group of electrodes from all the available 192 electrodes and tested the performance of the decoder on the selected electrodes. We have repeated this process 100 times for each group size. The group size can vary from 1 (that is, a single electrode) to 192 (all electrodes). Neuron dropping curves were generated on the neural data of participant JJ on the sessions recorded in 2019. This figure shows the performance of FENets that are trained on top 25, top 50, top 75, mid 25, and down 25 electrodes as well as the performance of the WTs and TCs. FENet trained on the top 50 electrodes shows superior performance and generalizability compared to the other techniques.

https://doi.org/10.1038/s41551-024-01297-1



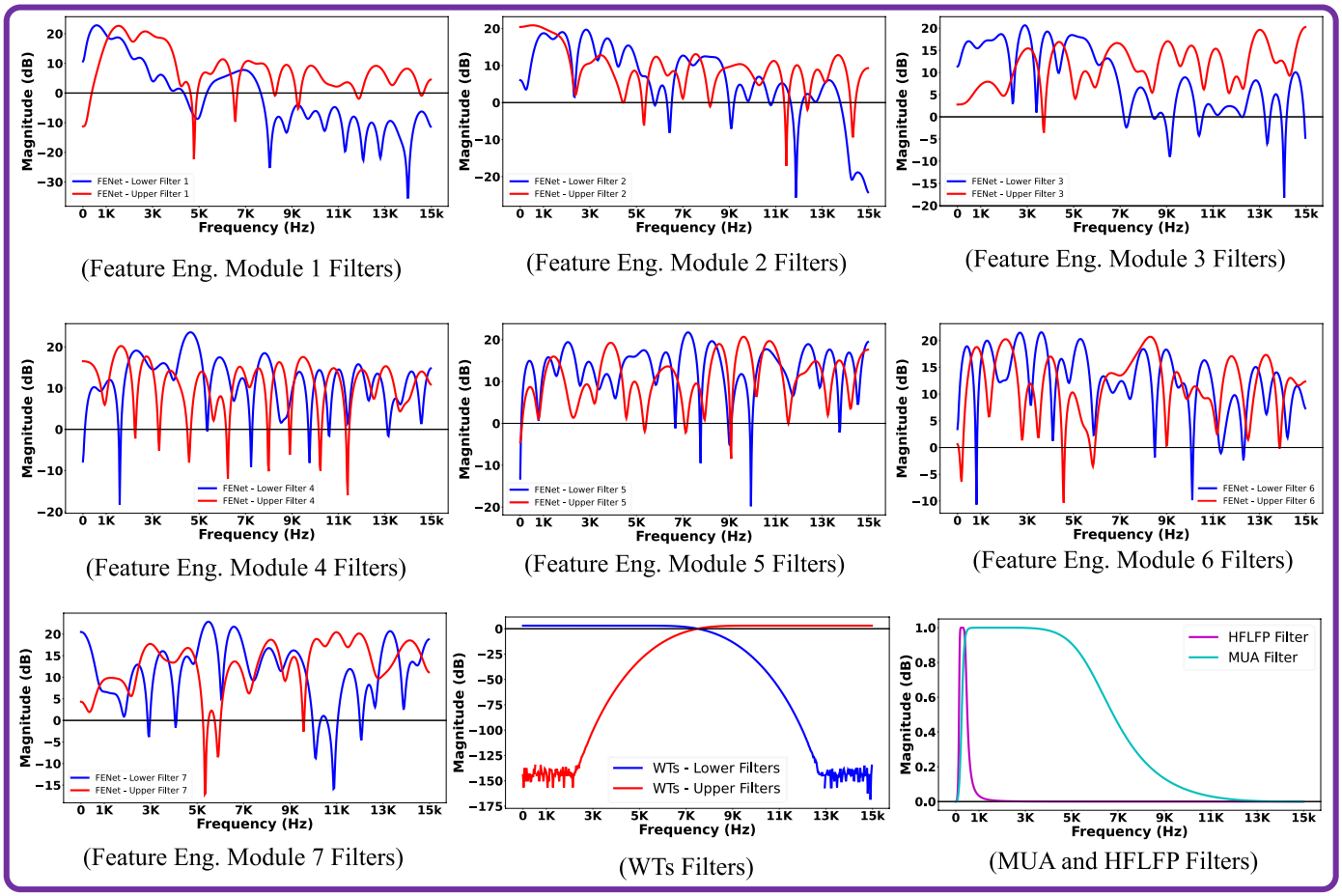
Extended Data Fig. 7 | Relative Shapley values of the eight FENet-extracted features. (Upper panel) We conducted parameter sweeps using Bayesian optimization on the FENet model to assess the importance and impact of each hyperparameter in the model's architecture. The results indicate a correlation between the R^2 values of the linear decoder and the parameter values. Notably, the strides of the initial layers emerge as the most influential parameters, with smaller strides yielding higher performance. This is because smaller strides allow the convolutional kernels to cover a greater variety of local patterns in the input. Conversely, larger strides limit the coverage between consecutive kernel movements, resulting in the filters learning fewer patterns. Additionally, we observed that the kernel size becomes more crucial in later layers compared to the initial layers. This suggests that the inputs to later layers summarize information from multiple samples in the preceding layers. Consequently, the network becomes more sensitive to kernel size when combining richer features with different kernel sizes, as these layers combine samples providing less abstract information than deeper layers. (Lower panel) (Left) The cross-validated R^2 of linear decoder operating on features extracted by using different FENet architectures vs the computational cost of these different architectures. (Right) The importance of each extracted FENet feature per electrode. We averaged the proportional Shapley values of all the electrodes over all the sessions for participant JJ.

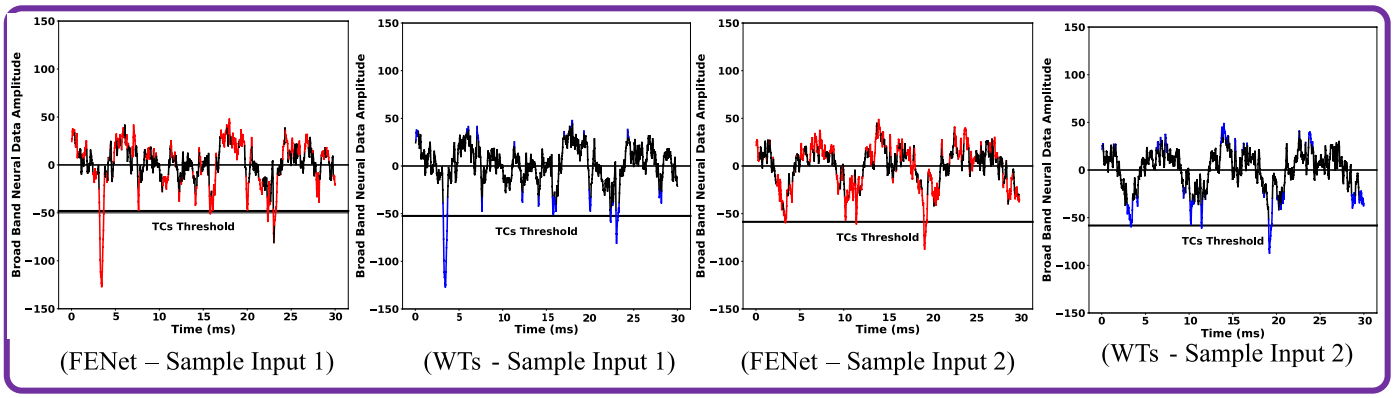
https://doi.org/10.1038/s41551-024-01297-1



Extended Data Fig. 8 | Assessment of extracted features. To assess the extracted features obtained from various feature-extraction techniques, the offline data of the top electrode of three sample sessions 20190314, 20200928, and 20210312 were partitioned into eight centre-out task trials, each trial corresponding to a different target. We named the target with $x > 0$ and $y = 0$ as Target0. The upper figures depict the average values of features obtained from multiple repetitions of a trial in their respective sessions, while the lower figures display the actual values of the extracted features during the initial 30 s of the recorded data. To identify the top electrode within a session, we organized electrodes based on their individual electrode R^2 values, indicating the linear predictability of kinematics for each electrode using each distinct feature extraction technique. Subsequently, we randomly chose above-mentioned three sample sessions spanning 2019, 2020, and 2021 from those where the index of the top electrodes remained consistent across all feature extraction techniques. The results demonstrate the preservation of the fundamental tuning characteristics of the neurons, FENet generates higher values within tuning curves and achieves improved trial separability compared to other feature extraction techniques.

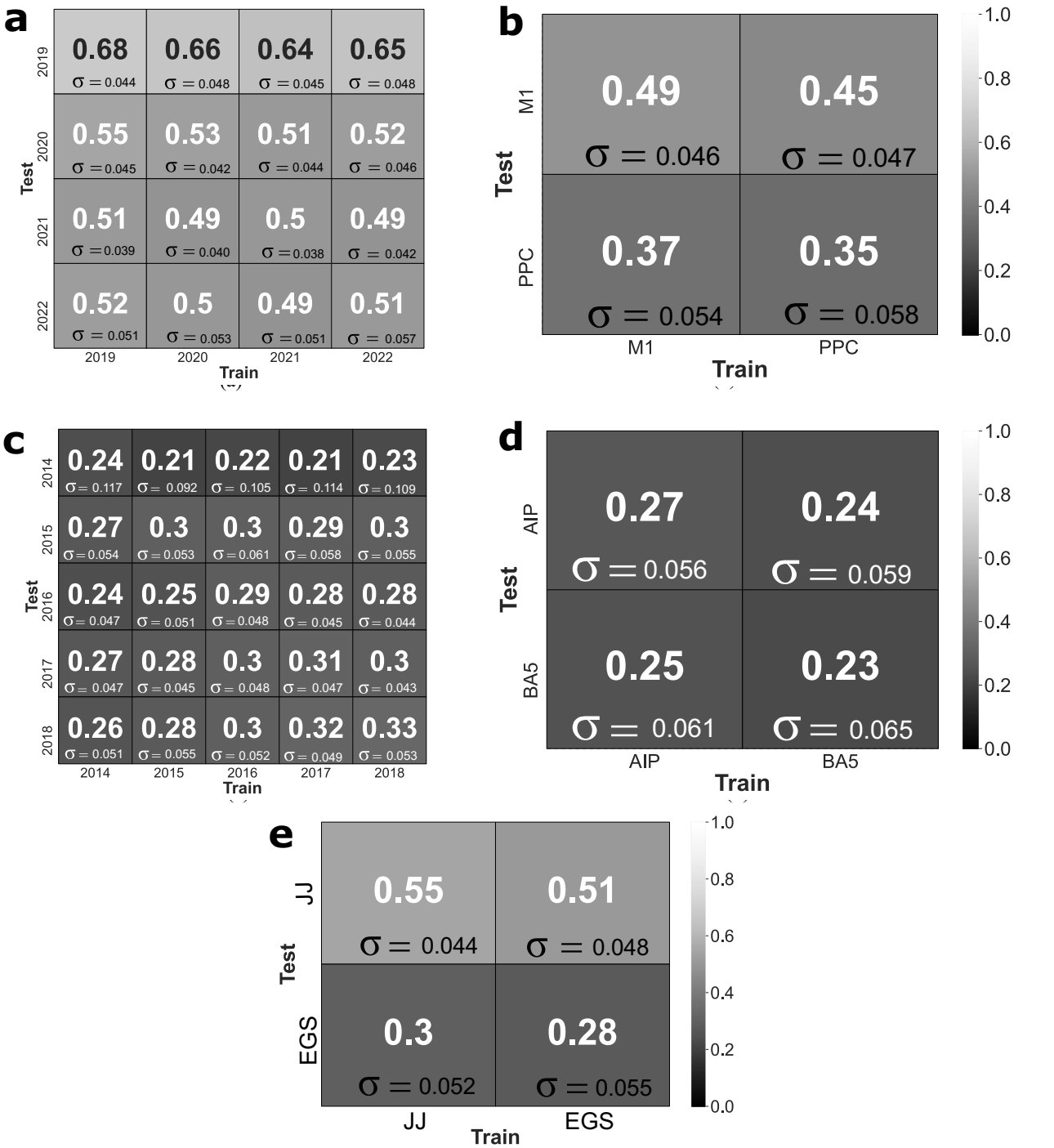
https://doi.org/10.1038/s41551-024-01297-1





Extended Data Fig. 9 | Gain of a sample set of FENet-trained convolutional filters across its feature-engineering modules. (Upper Panel) the gain of sample trained convolutional filters of FENet for all the feature engineering modules using JJ's data compared to the gain of WTs, MUA, and HFLFP filters. In contrast to these conventional filters, FENet exhibits the ability to dynamically amplify distinct frequency bands during its training process, considering the encoded information within each specific frequency band. (Lower Panel) We present two single electrode input samples from FENet and WTs obtained during a sample session of participant JJ, labelled as 20190625. In these inputs, the coloured sections highlight the segments that hold greater importance in the predictions made by the linear decoder operating on each of FENet and WTs as the feature extraction technique. To represent the more crucial sections of the input signals, we calculated the average Shapley value across all samples of these inputs and coloured the samples with Shapley values exceeding this average. Additionally, the horizontal lines indicate the threshold used to extract Threshold Crossings (TCs) features for each input sample. These figures provide evidence that the trained FENet not only utilizes spike information (TCs) and Wavelet Transforms (WTs) to extract features, but also identifies local patterns more effectively. Moreover, FENet demonstrates superior ability to track rapid and abrupt changes in the input signals compared to WTs. These findings indicate that FENet captures more nuanced and localized information, resulting in enhanced feature extraction capabilities when compared to WTs and TCs.

https://doi.org/10.1038/s41551-024-01297-1



Extended Data Fig. 10 | FENet generalizes within and across splits. FENet generalizability analysis using centre-out task. R^2 of training and testing FENet on different (**a**) time-periods and (**b**) brain regions for participant JJ and (**c**-**d**) for participant EGS. In each square, FENet is trained on the data coming from either the sessions/region corresponding to a specific year/brain region and is tested on the other year/region to evaluate generalizability of the FENet. (**e**) FENet trained on one human participant and tested on the other human participant (either participant JJ or EGS). These results suggest that FENet can generalize across different time periods, brain areas, participants, and electrodes. Linear decoder has been employed for all the analysis.

nature portfolio

Corresponding author(s): Benyamin Haghi, Tyson Aflalo and Azita Emami

Last updated by author(s): Dec 19, 2023

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☐	☒	A description of all covariates tested
☐	☒	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☐	☒	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	Data were collected using Blackrock Microsystems Biopotential Signal Processor and Matlab software.
Data analysis	The codes used for training and inference of FENet are available at https://github.com/BenyaminHaghi/FENet. Codes used for analysing and displaying the results presented in this study are available from the corresponding authors on reasonable request.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

The behavioural and neurophysiological data are archived in the Division of Biology and Biological Engineering at the California Institute of Technology. The broadband neural data are confidential and hence cannot be publicly shared. The raw and analysed data generated during the study are available for research purposes from the corresponding authors on reasonable request.

Research involving human participants, their data, or biological material

Policy information about studies with human participants or human data. See also policy information about sex, gender (identity/presentation), and sexual orientation and race, ethnicity and racism.

Reporting on sex and gender	Neural recordings were made from participants: JJ, a 54-year-old male; EGS, a 32-year-old male; and NS, a 62-year-old woman.
Reporting on race, ethnicity, or other socially relevant groupings	The study design did not consider race, ethnicity and other socially relevant groupings.
Population characteristics	We conducted our FDA-approved and IRB-approved BCI study with three participants: JJ, a 54-year-old tetraplegic (C5-C6) male; EGS, a 32-year-old tetraplegic (C5-C6) male; and NS, a tetraplegic 62-year-old woman with a complete C3/C4 spinal cord injury.
Recruitment	The participants were recruited through Casa Colina and Rancho Los Amigos patient network of occupational therapists, physicians, former patients, and support groups. Recruiting tetraplegic patients is difficult, so self-selection bias was not considered heavily. Key inclusion criteria included: * Non-functional hand and arm strength bilaterally * Age 22–65 * Able and willing to provide informed consent * Surgical clearance * Psychological support system Key exclusion criteria included: * Ongoing health concerns that put the participant at unnecessary risk for implantation surgery and implant maintenance.
Ethics oversight	The study and all procedures were approved by the Institutional Review Boards of the California Institute of Technology, the University of California, Los Angeles, Casa Colinas Hospital and Centers for Healthcare, the University of Southern California, and Rancho Los Amigos Rehabilitation Center.

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	No statistical methods were used to pre-determine sample sizes. Our sample sizes are similar to those reported in previous publications using similar experimental procedures.
Data exclusions	No data were excluded from the analyses.
Replication	All experimental findings were replicated over multiple sessions, with the same results for each session.
Randomization	Participant randomization was not relevant, because a within-participant and multiple-session experimental design was used. The sequence of tasks was randomized for each run / session.
Blinding	Blinding was not applicable, because the experiments did not involve different participant groups.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☐	☒ Clinical data
☒	☐ Dual use research of concern
☒	☐ Plants

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging

Clinical data

Policy information about clinical studies

All manuscripts should comply with the ICMJE guidelines for publication of clinical research and a completed CONSORT checklist must be included with all submissions.

Clinical trial registration	NCT01958086 and NCT01849822
Study protocol	Available at clinicaltrials.gov under the above registration numbers.
Data collection	All data were collected at Casa Colinas Hospital and Centers for Healthcare and at Rancho Los Amigos Rehabilitation Center. Data were collected in 2–3 hour recording sessions, which occurred 1–4 times/week over 4 years for JJ, 5 years for EGS, and 6 weeks for NS.
Outcomes	Primary outcomes: 1. Accurate control of computer cursor. 2. Absence of implant infection and irritation.