Supplementary Appendix

Contents

List of investigators and contributions

List of investigators (authors):

1. Nicholas S. Card
2. Maitreyee Wairagkar
3. Carrina Iaccobacci
4. Xianda Hou
5. Tyler Singer-Clark
6. Francis R. Willett
7. Erin M. Kunz
8. Chaofei Fan
9. Maryam Vahdati Nia
10. Darrel R. Deo
11. Eun Young Choi
12. Matthew F. Glasser
13. Leigh R. Hochberg
14. Jaimie M. Henderson
15. Kiarash Shahlaie
16. David M. Brandman*
17. Sergey D. Stavisky*

* These authors contributed equally

Author contributions:

N.S.C. led the experiments, implemented the real-time neural decoder and language models, designed and implemented the Copy Task and the self-initiated conversational task, analyzed the data, and created the figures. N.S.C. and M.W. developed and implemented the real-time neural signal processing, noise removal, and feature extraction pipelines. N.S.C., M.W., X.H., and T.S-C. coded the real-time data collection system and built the neuroprosthetic cart system. N.S.C., M.W., and C.I. collected the primary data for this study. C.I. interfaced with the participant and scheduled research sessions. E.M.K. ran hyperparameter sweeps to identify optimal hyperparameters for the neural decoder. C.F. designed and trained the language models. M.V.N. created the own-voice text-to-speech model. E.Y.C. acquired the MRI data for the HCP cortical parcellation for surgical targeting. F.R.W., D.R.D., E.Y.C, and M.F.G. processed and interpreted the HCP cortical parcellation data and provided presurgical target locations for array implantation. D.M.B and K.S. led planning and performing the surgical implant placement procedure. L.R.H. is the sponsor-investigator of the multisite clinical trial. D.M.B. was responsible for all clinical trial-related activities at U.C. Davis. N.S.C., M.W., S.D.S., and D.M.B. were involved in conceptualization of the study and experimental design. S.D.S. and D.M.B. supervised all aspects of the project. J.M.H. supervised the team members at Stanford. N.S.C. and S.D.S. wrote the manuscript. All authors reviewed and helped edit the manuscript.

Supplemental Text

S1 - Experimental procedures

S1.01 - Study participant

This study includes data from one participant (referred to as 'SP2' in this preprint rather than the actual trial participant designation, which the participant is familiar with, as per medRxiv policy) who gave informed consent and was enrolled in the BrainGate2 clinical trial (identifier: NCT00912041). This pilot clinical trial was approved under an investigational device exemption by the US Food and Drug Administration (Investigational Device Exemption #G090003). Permission was also granted by the Institutional Review Board at the University of California, Davis (protocol #1843264). SP2 gave consent to publish photographs and videos containing his likeness. All research was performed in accordance with the relevant guidelines and regulations.

SP2 is a left-handed man in his 40's with Amyotrophic Lateral Sclerosis (ALS). During the summer of 2023, his left precentral gyrus was implanted with four 64-channel, 1.5 mm-length silicon microelectrode arrays coated with sputtered iridium oxide (Blackrock Microsystems, Salt lake City, UT). For more information on array targeting, see Sections S1.02 and S1.03, below. Data are reported from post-implant day 25 onward.

SP2 is effectively paralyzed below the neck and severely dysarthric due to his ALS. He retains intact eye movement and limited orofacial movement with the capacity for vocalization, but is unable to produce intelligible speech (Audio 1). He typically communicates through trained interpreters or a gyroscopic head mouse (Quha Zono 2) that enables him to move and click a mouse on a computer screen.

S1.02 - Multi-modal MRI-based speech localization

Prior to array placement, SP2 underwent a multi-modal MRI session for array targeting based on the Human Connectome Project (HCP's) prior protocols[1,2] and as done for prior Braingate2 clinical trial participant 'T12'[3]. SP2 was scanned in a 3T Ultra High Performance scanner (GEHealthcare) with a Nova 32-channel coil. Scan parameters were based on HCP Lifespan protocols and modified for the GE system (Table S1). Briefly, 0.8mm isotropic T1w and T2w images were acquired together with 2mm isotropic resting state fMRI with TR=800ms in 4 runs each lasting 5 minutes and 45 seconds. In addition, phase reversed single band reference fMRI and spin echo MRI images geometrically and distortion matched to the MRI were acquired for distortion correction, unaliasing, and motion correction. The HCP's minimal preprocessing pipelines[1] were used to align the data within and across modalities, correct for image distortions, reconstruct white and pial cortical surfaces, and compute T1w/T2w myelin maps and cortical thickness maps. Subsequently, multi-run spatial Independent Components Analysis (sICA) was applied to remove spatially specific fMRI artifacts related to head motion, physiology, and the MRI scanner[4].

These independent components were hand checked after initial automated classification using the "FIX" tool[5] before non-aggressive regression of the artifactual components out of the fMRI data. Hand component classification was used because the application involved surgical planning. At this point the T1w/T2w myelin maps and fMRI data were used to align SP2's brain to the HCP's atlas space using MSMAll areal-feature-based surface registration[6]. This multi-modal cortical surface registration compensates for individual variability in areal size, shape, and position and enabled the HCP's multi-modal cortical parcellation[1] to be overlaid directly on SP2's pial surface. The multi-modal surface registration was hand checked by comparing the multi-modal features computed from SP2's brain to the same features in the atlas, with special attention paid to the features that defined the borders of areas 4, 6v, and 55b, including the T1w/T2w myelin maps (Fig. S1e) and multiple spatial ICA-based functional networks, including the language network (Fig. S1c-d), head sensori-motor network, and upper extremity sensori-motor network. Again these maps were hand checked given the neurosurgical targeting application. Once precise alignment of the HCP's atlas of multi-modal cortical areas was confirmed, targets for the arrays were proposed, including targets in ventral area 6v, area 4, dorsal area 6v, and area 55b. These visual analyses were carried out within the HCP's Connectome Workbench software (Fig. S1b-h).

S1.03 - Array placement targeting

The surgical targets for array placement within the precentral gyrus were chosen based on gross anatomical structure, vasculature, previous speech decoding studies[3,7–9], and from estimates of cortical boundaries obtained using a cortical parcellation method derived from multi-modal Human Connectome Project (HCP) data (see Section S1.02, above). For two arrays, we targeted the dorsal and ventral aspects of area 6v due to their contributions to speech decoding in[3]. A third array was targeted to speech primary motor cortex (area 4). We targeted area 55b for the fourth array due to emerging evidence it is a speech hub[10].

S1.04 - Neural signal processing and feature extraction

Neural signals were recorded using Neuroplex-E headstages (Blackrock Microsystems) attached to the two percutaneous connectors of the four implanted microelectrode arrays. The headstages analog filtered raw signals between 0.3 to 7.5 kHz (4^{th} order Butterworth filter) and performed analog-to-digital conversion with sampling rate of 30 kHz (250 nV resolution). 1 ms windows of the digitized 30 kHz signal from 256 channels were sent to our custom BRAND node (see Section S1.05) written in Python for real-time digital filtering and feature extraction.

Each incoming 1 ms neural signal window was first band pass filtered between 250 to 5000 Hz using a 4^{th} order zero-phase non-causal Butterworth filter. 1 ms neural signal windows were padded on both sides (using the previous 1 ms window on the left side and 1.2 ms of mean padding on the right side) to minimize discontinuities at the edges. Linear Regression Referencing (LRR) was used to reduce noise artifacts from all channels of filtered signal[3,11].

We extracted threshold crossings and spike-band power features from every 1 ms window of filtered and denoised neural signals. Threshold crossings were identified if the voltage of the signal in this window crossed the threshold of -4.5 times the root mean squared

(RMS) value of the neural signal for each channel. Spike-band power was obtained by squaring the samples in the window and temporally averaging it for each channel. Spike-band power was clipped to avoid outliers. This real-time signal processing, de-noising and feature extraction was performed in less than 1 ms, minimizing the delay. These neural features were then binned into 20 ms non-overlapping bins. Binned threshold crossing counts were obtained by summing threshold crossings in 20 consecutive neural feature windows. Binned spike-band power was computed by averaging spike-band power in 20 consecutive neural feature windows. Threshold crossings and spike-band power are commonly used measurements of local spiking activity that have been shown to be comparable to sorted single unit activity in terms of decoding performance and neural population structure[12–14]. For brain-to-text decoding, binned threshold crossings and spike-band power from all 256 electrodes were assembled into a single 1 x 256 feature vector at every time step. Sequences of the feature vectors were smoothed and normalized before passing them into the RNN decoder (see Section S2.01).

At the start of each session, a short “diagnostic” block with attempted speech of repeated single words was recorded (see Section S1.07), which was used get initial estimates of electrode-specific RMS thresholds for obtaining threshold crossing features and LRR filter coefficients for de-noising signals as described above. Subsequently during the session, we recomputed these RMS thresholds and LLR coefficients after every block of neural data recording. Recomputing these parameters after every block helped with minimizing nonstationarities in the neural activity throughout the day.

S1.05 - Data collection rig

All real-time data collection, processing, analysis, and decoding was done between a group of five computers communicating with one another over a local area network. A Windows 10 computer interfaced with the Neuroplex-E system to start and stop neural data recording. A second computer (Ubuntu 22.04 LTS) was used to process and extract neural features from raw 30k neural data. A third computer (Ubuntu 22.04 LTS) was responsible for real time neural decoding, fine-tuning the RNN model online, displaying the task to SP2, and displaying the task control GUI on the research monitor. Finally, a fourth computer (Ubuntu 22.04 LTS) was used to run the language model that converted phoneme sequences to words. We used the Backend for Realtime Asynchronous Neural Decoding (BRAND[15]) to run our data collection computer setup. All code was written in Python, C, or MATLAB.

S1.06 - Overview of data collection sessions

Neural data were recorded in 5-7 hour long research sessions, which took place at the participant’s home twice per week. Sessions typically included 1-2 breaks for food or beverages. During the sessions, SP2 sat in his power lift chair in an upright position. A computer monitor placed in front of SP2 displayed the task. An eye tracker mounted to the bottom of the computer monitor allowed SP2 to select on-screen “buttons” by looking at them. Data was collected in 15-25 minute “blocks” consisting of an uninterrupted series of trials. Trials could be paused as necessary and continued or terminated as appropriate. Between blocks, SP2 was encouraged

to rest as needed. Table S2 lists all data collection sessions reported in this study. In each session, we collected an average of 136 minutes of neural data.

In keeping with historical precedent in our clinical trial, we began data collection 25 days after implantation[16]. While a delay between surgery and device initialization is standard practice in clinically approved neuromodulation procedures such as deep brain stimulation and vagal nerve stimulation, in principle data collection could have begun within hours or days after implantation.

S1.07 - Instructed delay Copy Tasks

In an instructed-delay Copy Task (Videos 1-2), a prompted sentence was displayed as text on a screen facing SP2. A colored square changed from red to green to indicate when he should begin speaking. SP2 triggered the end of each sentence using an on-screen eye-tracker "button", at which time the final decoded sentence was read aloud with a text-to-speech algorithm that was customized to sound like the participant's pre-ALS voice[17] (Section S5). To support users incapable of eye gaze control, we also demonstrated sentence finalization triggered by neural decoding of SP2's attempted hand squeezes (Section S6). The majority of Copy Task blocks were 50 trials long, which took 15-25 minutes depending on how long the prompted sentences were.

At the start of each session, we did a "diagnostic block", which was an instructed delay task with 8 single-word cues each repeated 6-8 times. The word set consisted of the words 'bah', 'choice', 'day', 'kite', 'though', 'veto', 'were', and a 'DO NOTHING' condition where SP2 was instructed not to say or do anything, consistent with [3] . Data from this block was used to calculate initial thresholds and weights for linear regression referencing, which were then updated after each subsequent block.

S1.08 - Self-initiated conversational task

In the self-paced conversational task (Video 3), no prompted sentences were shown on screen. Instead, SP2 could say whatever he wanted. SP2 would initiate a new sentence by simply attempting to speak, which the BCI would reliably detect using only neural data. To accomplish this detection, the RNN decoder was always running in the background to predict phoneme probabilities every 80 ms. These phoneme probabilities were analyzed in real time to detect when speech had started (probability of any phoneme higher than the probability of silence) or ended (probability of silence higher than probability of any phoneme for 6 consecutive seconds; this duration was determined over the first few sessions of this task to balance accidentally timing out a sentence early with not making SP2 wait too long for it to end when he wants it to). SP2 could end a sentence using the eye tracker or by waiting six seconds for the trial to time out, after which time the final sentence was read aloud by the TTS algorithm. SP2 used the eye tracker to confirm whether the final decoded sentence was correct, or if not, he could specify whether it was "mostly correct" or "incorrect". Sentences that were confirmed to be correct were used to fine-tune the RNN in the background, which ensured that decoding performance remained stable and accurate throughout usage of the speech neuroprosthesis. The duration of personal use blocks ranged from approximately five minutes to 4 hours. The

design of the self-paced conversational task was continuously tweaked in response to feedback from SP2 (e.g., see Table S3).

S1.09 - Decoder evaluation

To evaluate speech decoding performance, we computed PER and WER using Levenshtein distance, which counts the number insertions, deletions, or substitutions necessary to match the decoded phonemes or words to the ground truth labels. For assessing the RNN output (without language models), we calculate the "raw PER" by comparing the most probable phoneme decoded in each time step (duplicates removed) with the ground truth phoneme sequence. Consistent with [3], reported error rates were aggregated across all evaluation sentences from each session by summing the number of errors (insertions, deletions, or substitutions) for all sentences and then dividing it by the total number of words in those sentences. This helps prevent very short sentences from overly influencing the result. Confidence intervals for error rates were computed via bootstrap resampling over individual trials and then re-calculating the aggregate error rates over the resampled distribution (10,000 resamples).

Blocks where the participant was excessively tired, per his own report, were excluded from evaluation (2 of 36 total blocks); the WERs on these blocks were 8.3% (session 14) and 5.3% (session 15). The first-ever closed-loop block (session 1) was excluded from evaluation because the participant cried with joy as the words he was trying to say correctly appeared on-screen (1 of 36 total blocks). In each session, we collected 1-4 evaluation blocks (50-200 sentences).

Before every session, an RNN was trained on all previous data. In early sessions (1-11), an additional new model was also trained halfway through data collection to calibrate it to the current day. From session 12 onward, after online fine-tuning was introduced, we stopped training a new model halfway through the day and instead relied on the online fine-tuning.

S1.10 - Sentence selection

For 50-word decoding (sessions 1 and 2), custom-written prompted sentences contained words from a 50-word vocabulary[7]. For 125,000-word decoding (sessions 2 onward), sentences were sourced from the Switchboard corpus[18], as in [3]. Additional training sentences were sourced from the OpenWebText2 corpus[19] and the Harvard Sentences[20] in an effort to expand the sampled vocabulary and thus the decoder's ability to generalize (Fig. 3c). Sentences were screened for grammatical errors or offensive language. We collected data for 4,444 prompted sentences over 17 sessions, totaling to 18.8 hours of neural recording.

S1.11 - Eye tracking

SP2's gaze data were tracked using a Tobii Pro Spark eye tracker. Eye tracker calibration was performed at the beginning of each session, and repeated as necessary between data collection blocks. During data collection blocks, SP2's on-screen gaze location

was recorded at 60 Hz and used to allow him to select on-screen “buttons” by simply looking at them. Gaze data was recorded independently from each eye before being averaged and smoothed over time. All eye tracker calibration, gaze data recording, and logic for on-screen button selection was done with custom written Python code that was integrated into our BRAND-based[15] data collection rig.

S2 - RNN decoder

S2.01 - RNN architecture and feature preprocessing

The RNN used in this study to predict sequences of phoneme probabilities from neural data has an RNN inspired by [3]. In brief, the RNN consisted of (1) linear day-specific input layers to correct for nonstationarity in neural data between days, (2) 5 layers of gated recurrent unit (GRU) architecture with 512 units per layer, and (3) a dense output layer that outputted the probabilities of 41 classes (39 phonemes, silence, and a CTC blank token). The RNN ran every 4 bins (20 ms per bin) to predict phoneme probabilities from the most recent 14 bins of neural data (280 ms). Before neural features were input into the RNN, they were z-scored using their means and standard deviations from the previous 20 trials, and then smoothed with a Gaussian kernel (sd = 40ms) that was delayed by 160ms. Connectionist temporal classification (CTC) loss was used to output a sequence of predicted labels (phonemes) from an unlabeled time sequence of neural data. For additional details about the RNN architecture, refer to the supplemental methods section of [3].

S2.02 - Offline RNN training

The offline RNN training protocol used here is the same as in [3]. A new RNN was trained before each session, and also mid-session for the first 11 sessions (before online fine-tuning was introduced). Whenever an offline RNN was trained, 90% of all previous data were used for training, and 10% of data were randomly (uniformly from each session) held-out for validation. Cue sentences from each trial were converted to a sequence of phonemes using the Python g2p-en package [21]. The RNN was trained with neural feature sequences and target phoneme sequences for 5,000-15,000 batches (the number of batches was increased as the training data pool grew throughout data collection), and the learning rate was linearly decayed from 0.02 to 0.0 across all batches. In each batch, up to 64 trials of data from a randomly selected session were input into the corresponding day-specific input layer followed by the GRU and dense layers. Data was dynamically augmented on a batch-by-batch basis to improve decoder generalizability and stability by adding (1) white noise and (2) artificial constant offsets to the neural features. At the end of each batch, weights for the relevant day-specific input layer and the GRU layers were updated using stochastic gradient descent (ADAM; β_1 = 0.9, β_2 = 0.999, ϵ = 0.1). We applied dropout and L2 weight regularization during training to improve generalization. For additional details about the offline RNN training, refer to the supplemental methods section of [3].

S2.03 - Online RNN finetuning

Online RNN fine-tuning was introduced in session 12 in an effort to constantly adjust the RNN to shifts in neural signals to ensure that speech decoding performance remains consistently high throughout a session. The online finetuning method employed here is similar to the one introduced in [22] for a handwriting decoder, but was adapted here to work with speech decoding tasks. At the beginning of a new session, an RNN trained on all previous data was loaded, and the day-specific input layer corresponding to the most recent session was duplicated. After each trial in the new session (starting after ten trials of data had accumulated), the weights of this day-specific input layer and the weights of the base GRU model were updated using the neural data and ground-truth sentences from each trial. Ground-truth sentences are defined as either the cued sentence (in the instructed delay Copy Task) or as decoded sentences that SP2 confirmed to be correct (using the eye tracker) in the self-initiated conversational speech task after each sentence. During each fine-tuning epoch, data from previous sessions was randomly sampled (in a proportion of 60% new data to 40% old data) to train the model in an effort to ensure that the model did not overfit to the current day's data. A static learning rate of 0.04 was used to fine-tune the RNN throughout the session. For additional details about online RNN fine-tuning, refer to [22].

S2.04 - Hyperparameter optimization

Optimal hyperparameters for RNN architecture and training were determined with hyperparameter sweeps twice throughout data collection (Fig. S5), and optimal parameters were used in subsequent online decoding sessions. RNNs were trained offline (using all previous data) with one hyperparameter varied at a time. Each RNN was validated on randomly-selected held-out validation trials to evaluate performance (raw phoneme error rate). For each parameter condition, 10 RNNs were trained and their validation performances were averaged.

S3 - Language model

S3.01 - Architecture

The *n*-gram language models in this study, which take sequences of phoneme probabilities as an input and output the most likely sequence of words, are the same general architecture as described in [3]. The 50-word language model used here was a 5-gram model trained on 2,413 custom-written sentences that contained only words from the 50 word vocabulary[7]. This model outputs only the singular most likely sequence of words. The 125,000-word language model, trained on the OpenWebText2 corpus[19], was the same 5-gram model described for post-hoc offline analyses in [3]. The 125,000-word vocabulary included in this language model stems from the CMU dictionary (http://www.speech.cs.cmu.edu/cgi-bin/cmudict) and encompasses the majority of the English language; native English speakers typically know ~20,000-40,000 words[23,24]. This language model initially predicts up to 100 of the most likely sequences of words, before rescoring them in multiple stages to identify the singular most likely

sequence of words. To use this language model online and in real time, we implemented custom-written Python code to integrate it into our real-time decoding system. For additional details about the language models utilized in this study, refer to the supplemental methods section of [3].

S3.02 - Hyperparameter optimization

As we collected data, we ran offline language model hyperparameter sweeps to identify optimal speech decoding parameters (Fig. S6). In particular, the *acoustic scale* and *alpha* parameters had the biggest impact on decoding performance. The *acoustic scale* parameter is the weighting ratio between the RNN-derived phoneme probabilities and the n-gram derived sentence probabilities, and the *alpha* parameter is the weight ratio between the *n*-gram model rescoring and the OPT LLM rescoring (see supplemental methods in [3] for more details). After hyperparameter tuning, we used these identified optimal hyperparameters to enhance speech decoding accuracy online in subsequent speech decoding sessions.

S4 - Offline analyses

S4.01 - Offline RNN analyses

Offline RNN decoding analyses were utilized in Figures 2, 3, S3, and S5 of this study. All analyses averaged the results from 5-10 RNN seeds per condition. Chance decoding values (reported in Fig. 3c) were calculated by training a decoder where the phonemes of the ground-truth sentence for each trial were shuffled. This allowed us to maintain the statistical distribution of uttered phonemes while obtaining a chance value. For each channel count condition of the channel dropping analysis (Fig. 2d), random channels were sampled uniformly from all arrays, and data from unused channels was zeroed out. Unless specified, RNN architecture and hyperparameters remained consistent between all conditions.

S4.02 - Offline language model analyses

Offline language model analyses were performed for Figures S4 and S6. In these offline analyses, RNN-predicted phoneme sequences, either from online speech decoding sessions or from offline RNN analyses (see Section S4.01), were sequentially fed into language models (consistent with online language model inference) for each trial of data, before finalization. Offline language models were initialized with a range of parameters or vocabularies as relevant for the analysis. Language model performance was assessed as the aggregate WER of all predicted sentences, calculated as described in Section S1.09.

S5 - Own-voice text-to-speech

A text-to-speech algorithm[17] was locally trained to sound like SP2's pre-ALS voice. SP2 and his family provided us with home videos and other recordings of SP2 speaking. Recordings with clear samples of SP2's voice were segmented into individual sentences, noise-reduced (RNNoise 1.4[25]), and amplitude-normalized in preparation for training the TTS model. Signal-to-noise ratio (SNR; calculated with Waveform Amplitude Distribution Analysis (WADA) with the Coqui TTS Check-DatasetSNR notebook[17]) was used to quantify how noisy each audio clip was. Each audio clip was then transcribed, and the phoneme distribution across all audio clips was calculated (using the Python g2p-en package[21]) to ensure that each phoneme was adequately sampled. Comprehensive coverage of each phoneme is required to train robust text-to-speech models and accurately reproduce speech patterns.

For training an own-voice TTS, we chose the VITS model[26], which we subjectively found to reproduce SP2's pre-ALS voice most accurately and without a long delay at the end of each sentence. A pre-trained VITS model with LJSpeech corpus was fine-tuned using SP2's processed audio samples to create a TTS that sounded like SP2's pre-ALS voice. This TTS was used to read the final decoded sentence at the end of each trial in both the Copy Task and the self-initiated conversational task. SP2 and his family found our recreation of his pre-ALS voice to be more representative than his previously purchased commercial version.

S6 - Gesture decoding for task control

People with ALS may lose eye gaze control as their disease progresses. Thus, eliminating reliance on eye gaze (e.g., by making the system controllable through exclusively neural signals) is necessary to ensure that the BCI system will remain usable in the long-term for users with degenerative diseases such as ALS. We provided SP2 with a "neural click" functionality as an alternative to eye tracker control for indicating that he is done speaking, which triggers sentence finalization and text-to-speech output. This neural click functionality was used in sessions 17 and 18.

S6.01 - Motor imagery

We chose "right-hand squeeze" as the motor imagery to perform the neural click, because it had a robust neural SNR in previously-collected SP2 movement sweep data, and the hand squeeze gesture has been used for neural click in prior iBCI studies[27,28]. Other discrete gestures may have worked just as well for this purpose[29].

S6.02 - Decoder architecture

We implemented a linear gesture decoder (independent from the RNN speech decoder) to solve the binary classification problem: *"For each 10 ms bin, is the user attempting right-hand squeeze or not?"*. We used a linear discriminant analysis (LDA) model, because linear models

are simple and fast to train, and were able to reliably distinguish the neural correlates of hand squeezes from those of speech or silence in offline tests.

S6.03 - Decoder training

We interspersed 16 trials of a "*RIGHT HAND - CLOSE"* condition into the instructed delay task (our "diagnostic block") performed at the start of each session. After the diagnostic block, we trained the LDA classifier on all the trials ("*RIGHT HAND - CLOSE"* = click, single-word speech trials = non-click, "*DO NOTHING"* = non-click). The training data from each trial came from the epoch 0.5 - 1.5 seconds after the go cue. Each trial yielded 100 training samples, as our LDA classifier operated on individual 10 ms time bins. Each training sample was a single time bin's feature vector (256 threshold crossings + 256 spike band power = 512 features). These feature vectors were processed identically to those used for speech decoding (filtered, z-scored, etc.). To fit the LDA model on these training data, we used the *LinearDiscriminantAnalysis* class from the Python package *sklearn*[30].

S6.04 - Decoder inference

After training the LDA classifier, we used it during online speech blocks to decode neural clicks in real-time. Because this LDA neural click decoder was independent from the RNN speech decoder, it was run in parallel using the BRAND software architecture. Though the LDA model outputted a prediction for every 10 ms time bin, a click was not immediately performed every time the LDA model predicted click. Instead, a click was only performed when all time bins in a 100 ms sliding window were predicted as click, to reduce spurious clicks. Additionally, after each click we maintained a refractory period of 1 second during which no additional clicks were performed, to avoid rapidly clicking.

Supplemental Figures

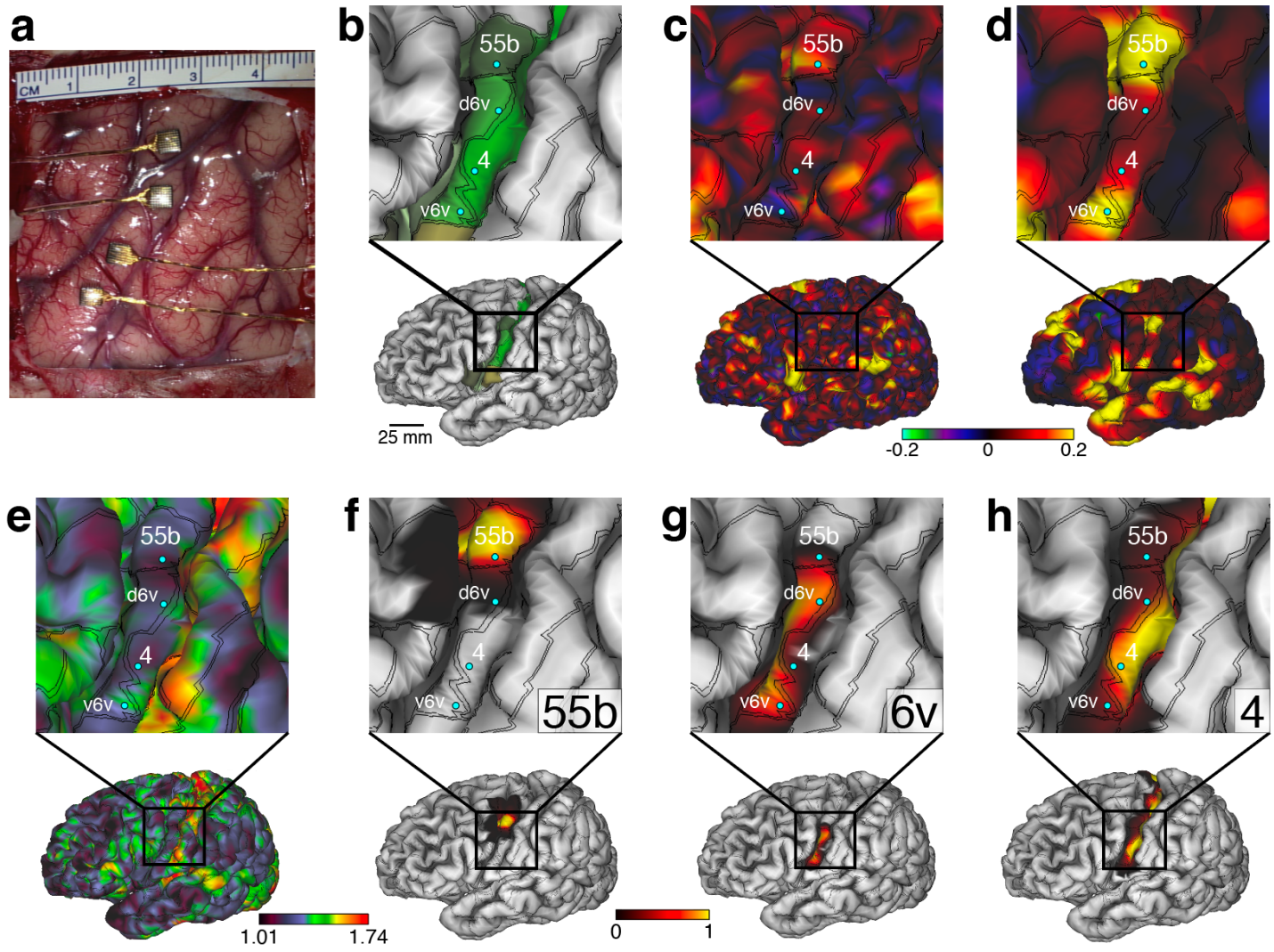

Figure S1: Multi-modal MRI-based speech localization and array targeting.

a, Array implants shown on the surface of SP2's brain during surgery. **b,** Approximate array locations on SP2's inflated brain using Connectome Workbench software, overlaid on the cortical areal boundaries (double black lines) estimated by the Human Connectome Project (HCP) cortical parcellation. **c**, Approximate array locations overlaid on a language-related resting state network shown for SP2's individual scan. **d**, The same resting state network identified in the Human Connectome Project data (i.e., averaged across many subjects) and aligned to SP2's brain. **e**, Approximate array locations overlaid on a myelin density map. **f-h**, Approximate array locations overlaid on the confidence maps of the areal region labeled in the bottom right of the magnified panel.

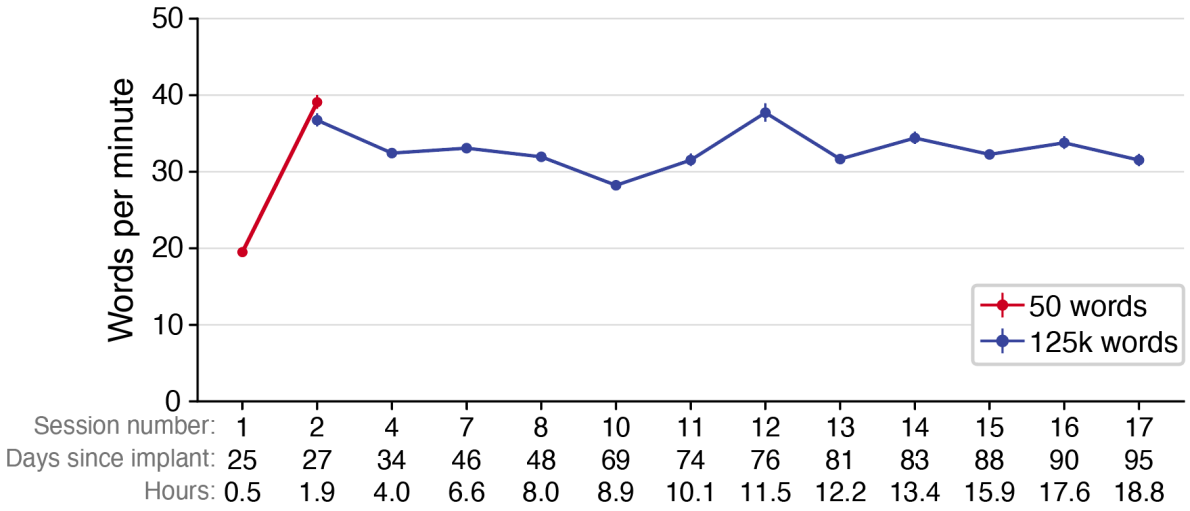

Figure S2: Words per minute during evaluation blocks.

SP2's average rate of attempted speech during evaluation blocks for each session. Error bars denote the 95% confidence interval. For each sentence, words per minute (WPM) was calculated as the number of words in the target sentence divided by the duration from the beginning of the first word until SP2 signaled the end of the sentence (using the eye tracker or gesture decoder). The relatively low WPM in session 1 may be due to SP2 getting used to using the speech decoder.

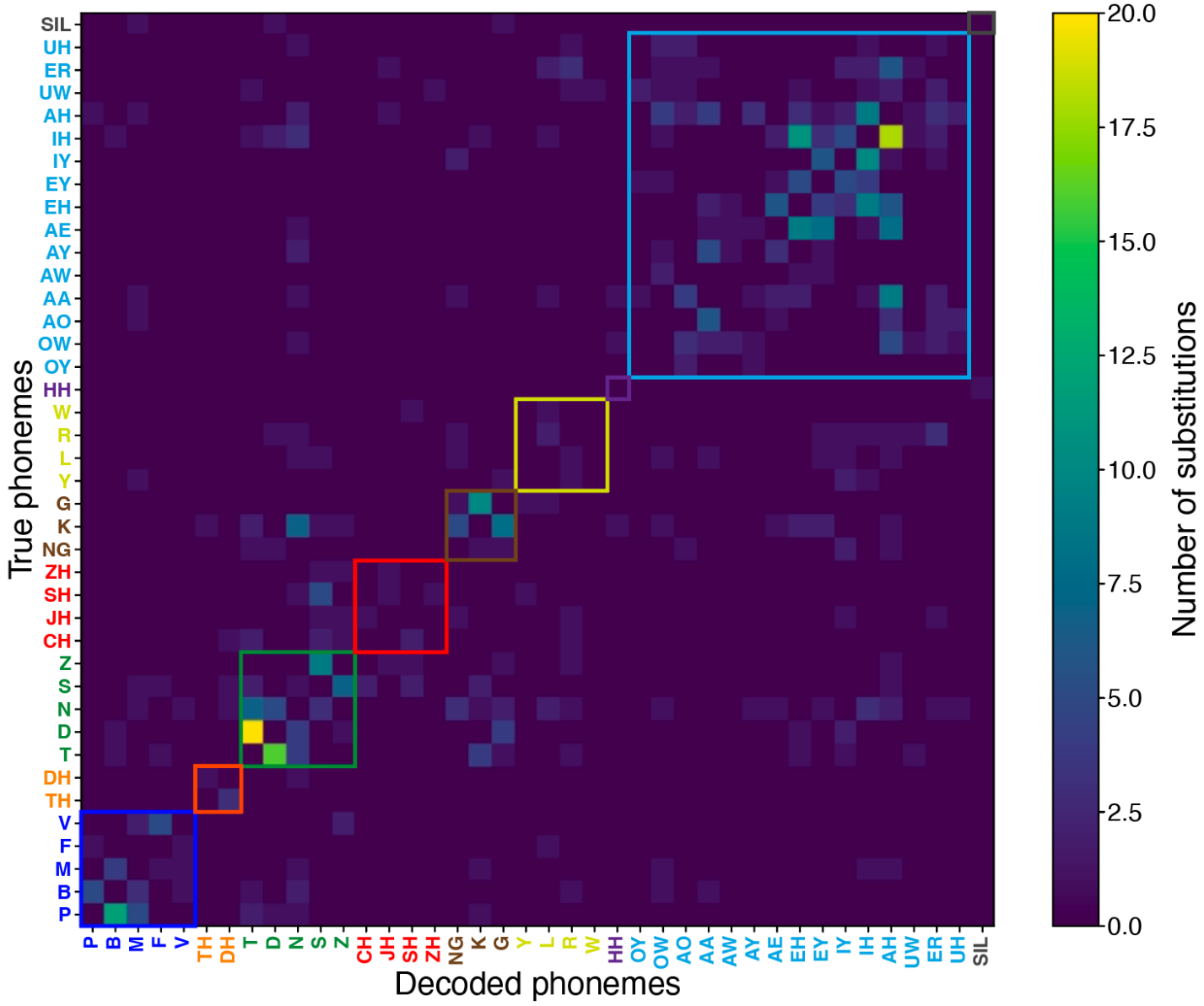

Figure S3: Phoneme substitution errors observed across all real-time evaluation sentences.

Entry *(i,j)* in the matrix indicates the number of substitutions between true phoneme *i* and decoded phoneme *j*. Substitutions were identified using an edit distance algorithm that determines the minimum number of insertions, deletions, and substitutions required to make the raw (pre-language model) decoded phoneme sequence match the true phoneme sequence. The majority of substitutions appear to occur between phonemes that are articulated similarly (within place of articulation groupings indicated by the boxes colored the same as in Fig. 2e), including between voiced and unvoiced consonant pairs (e.g., /p/ vs /b/, and /t/ vs /d/).

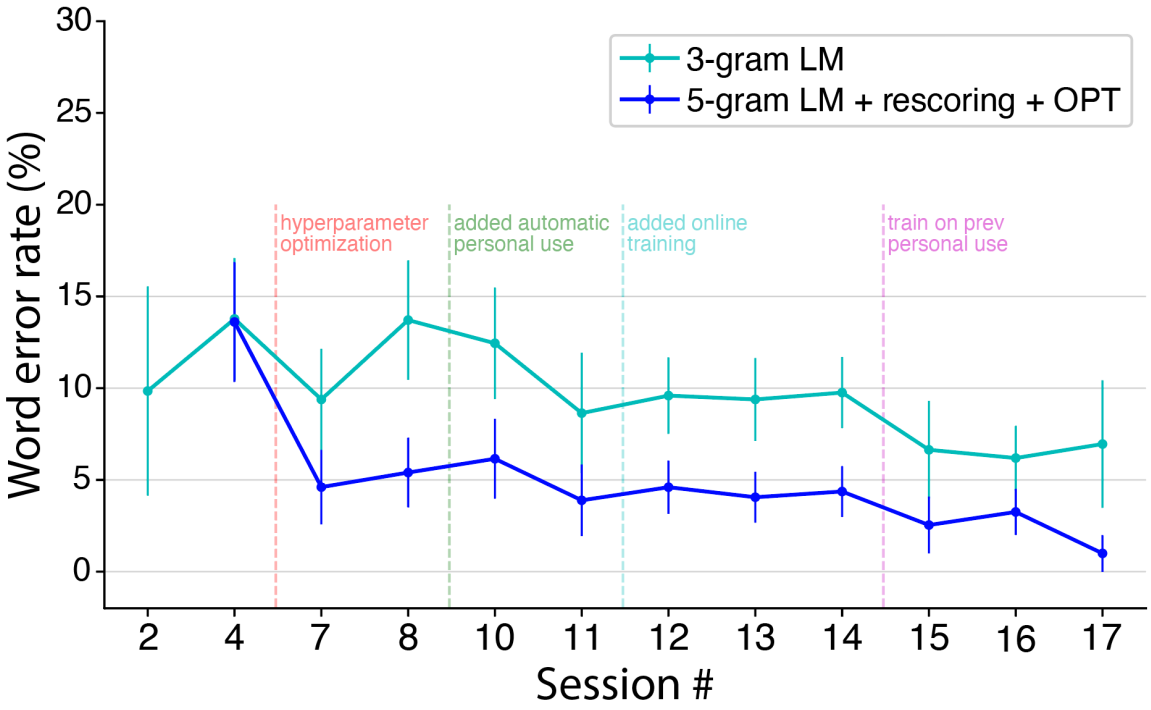

Figure S4: Decoding performance comparison between 3-gram and 5-gram language models.

Comparison of offline evaluation performance using a 3-gram language model without rescoring (cyan line; as demonstrated online in Willett et al. 2023) or a 5-gram language model with multi-stage rescoring of candidate sentences (blue line; as demonstrated offline in [3] and online in the main figures of this study). Both models used the same 125k-word English vocabulary. RNN-decoded phoneme probabilities from SP2's online evaluation blocks were fed into both language models in offline analyses to compare their performance. Results were averaged over 5 RNN seeds. We used the 3-gram language model for online evaluation in session 2, and the upgraded 5-gram language model in subsequent sessions. After hyperparameter optimization of both the RNN decoder and the language model (red dashed line; between sessions 4 and 7), the 5-gram language model consistently outperformed the 3-gram model, resulting in lower word error rates.

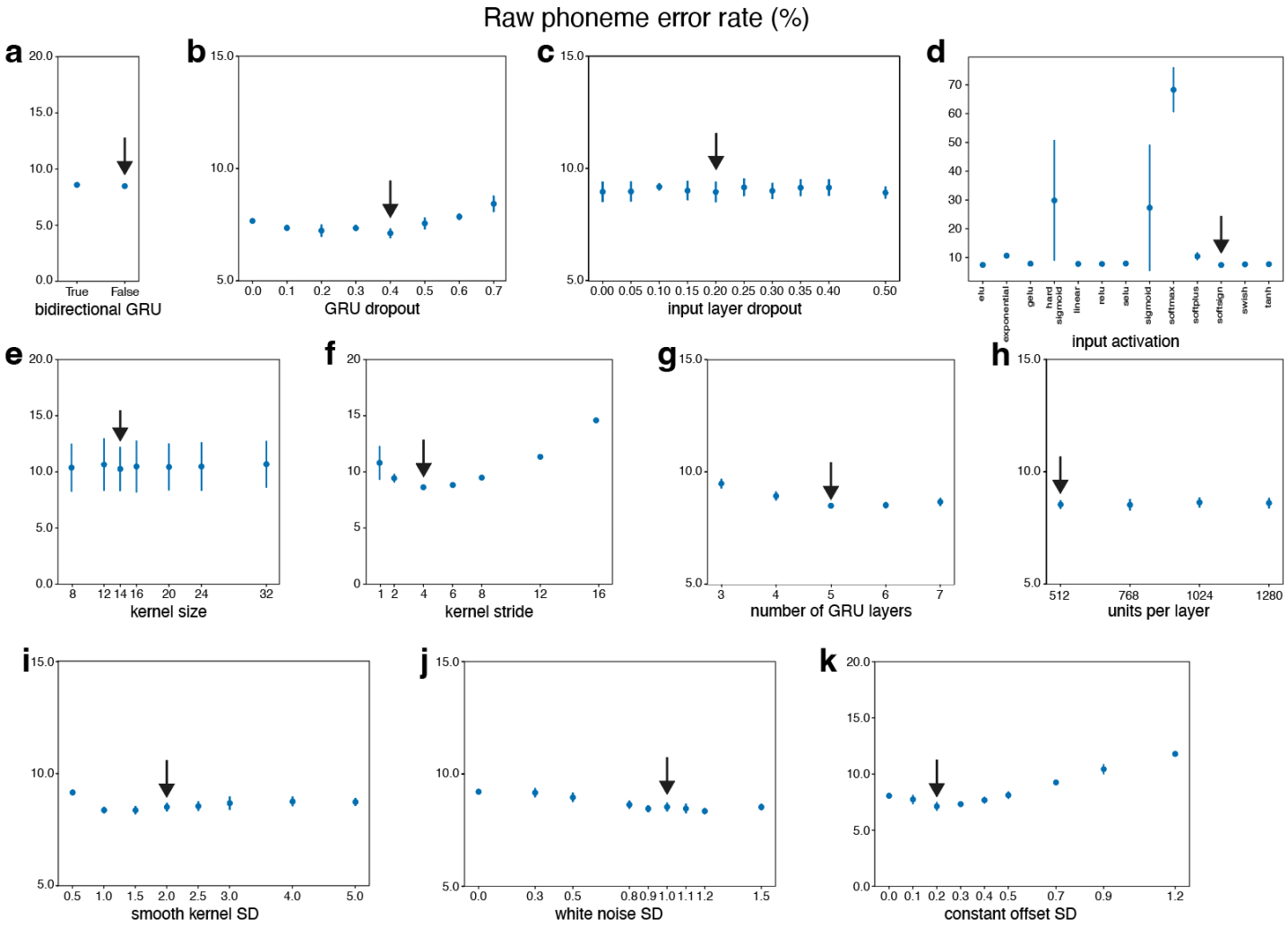

Figure S5: Offline parameter sweeps indicate near-optimal RNN parameter choices were used online.

We tested the effect on raw (pre-language model) phoneme error rate (PER) as a function of several RNN parameters. Each point in each plot represents the average raw PER (± standard deviation) of 10 RNN seeds trained with the corresponding parameter, on data from the first *n* sessions. This process was repeated twice throughout data collection to ensure that we were using optimal RNN decoding parameters in subsequent online decoding sessions. Here, results from the first 12 sessions of data are shown. Black arrows represent parameters used in online evaluation. Tested parameters include: **a**, Bidirectional vs. unidirectional GRU layers. **b,** Dropout percentage for GRU layers. **c,** Dropout percentage for input layers. **d,** Activation type for input layers. **e,** "Kernel size" (i.e., the number of 20 ms bins stacked together as input and fed into the RNN at each time step). **f,** "Kernel stride" (a stride of N means the RNN steps forward only every N time bins). **g,** Number of GRU layers. **h,** Number of units per GRU layer. **i,** Standard deviation of the Gaussian smoothing kernel (larger number means more smoothing). This parameter was not quite optimized for online decoding. **j**, Standard deviation of white noise dynamically added to training data during RNN training for data augmentation. This parameter was not quite optimized for online decoding. **k,** Standard deviation of constant offset noise added to training data during RNN training for data augmentation.

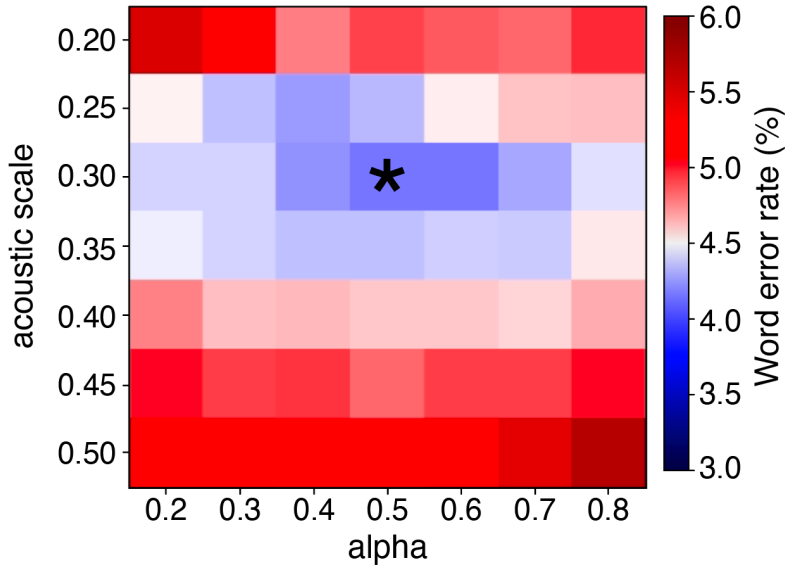

Figure S6: Offline language model parameter sweeps informed subsequent online parameter choices.

We ran offline analyses to identify the optimal language model parameters (i.e., the parameters yielding the lowest word error rate [WER]). An RNN was trained on all data from the first *n* sessions (in this case, *n*=12), and the RNN-decoded phoneme probabilities from held-out validation trials were fed into 5-gram language models initialized using a range of parameters. Varied parameters included the blank penalty, acoustic scale, and alpha values (Section S3; also see supplemental methods section of [3]). While varying the blank penalty (set to log(9) here) did not result in a large change in WER, the acoustic scale and alpha parameters made appreciable differences in performance. This language model parameter sweep was repeated thrice throughout data collection, and consistently showed that an acoustic scale of 0.3 and an alpha of 0.5 (denoted with * in the plot) resulted in the lowest WER. These optimal language model parameters were subsequently used for online speech decoding.

Supplemental Tables

Table S1: MRI Scan Parameters

Image	**T1w**	**T2w**	**rsfMRI**	**rsfMRI-single band**	**spin echo fieldmap**
Sequence	3D MPRAGE	3D CUBE	2D Gradient Echo EPI	2D Gradient Echo EPI	2D Spin Echo EPI
TR (ms)	3000	2500	800	4200	8000
TE (ms)	3.5	60-78	37	30	min full
TI (ms)	1060	-	-	-	-
Parallel imaging	2 x 1.25	1.9 x 1.9	-	-	-
Fat suppression	no	no	yes	yes	yes
Resolution (mm)	0.8 x 0.8 x 0.8	0.8 x 0.8 x 0.8	2 x 2 x 2	2 x 2 x 2	2 x 2 x 2
Matrix size	320 x 320 x 230	320 x 320 x 216	104 x 104 x 72	104 x 104 x 72	104 x 104 x 72
FOV (mm)	256 x 256 x 184	256 x 256 x 184	208 x 208 x 144	208 x 208 x 144	208 x 208 x 144
Flip angle	8	-	54	90	-
slice orientation	sagittal, AC-PC	sagittal, AC-PC	axial, AC-PC	axial, AC-PC	axial, AC-PC
phase encoding			AP and PA (separately)	AP and PA (separately)	AP and PA (separately)
multiband factor	-	-	8	1	-

Table S2: Data collection sessions

Session Number	Post-implant day	Description	Data
1	25	50-word training data collection and decoding	290 50-word-vocab training sentences. 50 50-word-vocab evaluation sentences.
2	27	125,000-word training data collection and evaluation	330 Switchboard training sentences. 40 50-word-vocab training sentences. 50 50-word-vocab evaluation sentences. 30 Switchboard evaluation sentences. 10 personal use sentences.
3	32	125,000-word training data collection	300 Switchboard training sentences.
4	34	125,000-word training data collection, evaluation, and personal use	280 Switchboard training sentences. 100 Switchboard evaluation sentences. 74 personal use sentences.
5	39	125,000-word training data collection and other experiments	140 Switchboard training sentences.
6	41	125,000-word training data collection	200 Switchboard training sentences.
7	46	125,000-word training data collection and evaluation	300 Switchboard training sentences. 100 Switchboard evaluation sentences.
8	48	125,000-word training data collection and evaluation	275 Switchboard training sentences. 120 Switchboard evaluation sentences.
9	67	125,000-word training data collection and other experiments	10 Switchboard training sentences.
10	69	125,000-word training data collection, evaluation, and personal use	170 Switchboard training sentences. 115 Switchboard evaluation sentences. 180 personal use sentences.
11	74	125,000-word training data collection, evaluation, and personal use	120 Switchboard training sentences. 50 OpenWebText training sentences. 75 Switchboard evaluation sentences. 140 personal use sentences.
12	76	125,000-word training data collection, evaluation, and personal use	50 Switchboard training sentences. 40 OpenWebText training sentences. 210 Switchboard evaluation sentences. 85 personal use sentences.
13	81	125,000-word training data collection and evaluation	110 Switchboard training sentences. 140 Switchboard evaluation sentences.
14	83	125,000-word training data collection, evaluation, and personal use	115 Switchboard training sentences. 45 OpenWebText training sentences. 170 Switchboard evaluation sentences. 280 personal use sentences.
15	88	125,000-word training data collection, evaluation, and personal use	140 Switchboard training sentences. 90 Switchboard evaluation sentences. 100 personal use sentences.
16	90	125,000-word training data collection, evaluation, and personal use	140 Switchboard training sentences. 40 OpenWebText training sentences. 150 Switchboard evaluation sentences. 140 personal use sentences.

17	95	125,000-word training data collection, evaluation, and personal use	50 Switchboard training sentences. 20 Harvard training sentences. 50 Switchboard evaluation sentences. 140 personal use sentences.
18	97	125,000-word training data collection and personal use	100 Switchboard training sentences. 30 Harvard training sentences. 180 personal use sentences.

Table S3: Additional selected personal use transcripts.[1]

Context	Selected transcripts	WER
Session 11: SP2 thanks a member of the research team for complimenting his plants	SP2: testing testing one two SP2: hello how is everyone SP2: thank you for complimenting my rubber and stick [snake] plan [plants] … SP2: when i first got them they were in a one inch pot SP2: and they resided on my window sill SP2: hell yeah SP2: your degree is not in botany SP2: you know the saying that before you have kids you need to successfully take care of place [plants] in [and] the [then] past [pets] SP2: the progression is place [plants] and then past [pets]	11.3%
Session 15: SP2 gives feedback to Sergey about how he's enjoying using the BCI for conversational speech.	SP2: he should know better than to ask me what i want to say SP2: i'm a smart ass SP2: does he want me to talk in english or russian SP2: yes it should SP2: we can ease into it by talking in spanish first … SP2: have you noticed anything different with the program SP2: it is way more accurate than before SP2: it is about ninety eight percent accurate by my informal estimation … SP2: thank you again for all the improvements and the time it took to make them	0%
Session 17: SP2 gives feedback about using the neural-click decoder (right hand squeeze) to signal the end of sentences. Then he suggests adding a third confirmation button "mostly correct" (in addition to "100% correct" and "incorrect") to the self-initiated conversational task.	SP2: does it make a sound when i press the button SP2: that makes sense SP2: do you want some feedback about my right hand squeezing SP2: so i can actually still squeeze my hand SP2: but it was not a problem because i really had to intend to raise [squeeze] my hand or it would not work SP2: the only problem was if i was running [yawning] SP2: so when i was doing that i would involuntarily raise [squeeze] my hand and that would trigger the complete button SP2: but initially my fears were mainly unfounded SP2: yes i do but not because i didn't like using the eye tracker method but because i like having multiple options SP2: thank you … SP2: have you thought about adding a third party [button] that is almost correct SP2: yes that would be good SP2: we can try the eye tracker SP2: how many rolls [trials] do you need in total SP2: let's do one country [hundred] at a time	4.7%

Session 18: SP2 is telling a friend about the speech decoder.	SP2: testing testing SP2: thank you SP2: what i was trying to say is that i have noticed that the computer has the same problem understanding what i am saying that people have meaning the same exact words that people have a problem with the computer also has a problem understanding SP2: totally SP2: yes i can in the last session the computer had problems with understanding when i said next and it made the mistake of typing this instead of next and i thought that people who can understand me often make the same mistake SP2: totally … SP2: when it is thinking of what to write it is because it is confused about what i said and it is running through different models of possible sentences SP2: know [no] what i am doing is all the same type of degree [decoding] SP2: the last word should have be be [been] guarding [decoding] SP2: so we have found that there is a slight increase in accuracy when i realize [vocalize] what i am saying but we think that is possibly true because the model trained on this approach rather than me doing what i am doing now SP2: the word really [realize] should have been vocally [vocalize] … SP2: we are in the future SP2: on this post [wednesday] i will have my first day of only personal use SP2: on this coming west [wednesday] SP2: why can you not see [say] mr [wednesday] SP2: and it is having a problem with hearing me when i am saying that particular day of the week and now i am talking a little shit to it	4.9%
Session 30: An interviewer is asking SP2 about his experience with using the speech decoder.	Interviewer: Hello [SP2], thank you so much for letting us into your home and taking the time to talk to us, it is nice to meet you. SP2: well it is nice to meet you and also your excellent camera made [man] SP2: your name is [camera man's name] right Cameraman: Yup! Nice to meet you. Thank you for letting us into your home. SP2: that is okay i will be very happy to have both of you here to witness what i can do with my super powers Interviewer: So tell me [SP2] why are you doing this, and what it means for you to be a part of this. SP2: well i have this terrible disease and it is slowly taking away my ability to move and to talk Interviewer: I am so sorry, [SP2]. SP2: it is fine but i am not a fan [ashamed] of you seeing me cry SP2: it should have said that i am not at camp [ashamed] to have you see me cry SP2: so because i have this terrible disease i have had the pleasure of meeting some amazing people like the nurse [ones] here SP2: the one [ones] where [here] SP2: i did my homework when i was thinking about having brain surgery because it was not an easy decision for me but i trusted the team that was behind me and i asked david if he would do this if he was in my position and while i will not tell you what he said it helped me make my decision with confidence Interviewer: Okay, how did it help you?	2.7%

	SP2: i think that i understand that there are human beings that are behind all of this science and technology where they are thinking about what would be better for me and my family rather than simply thinking about me as a test subject SP2: does that make sense to you Interviewer: Yes, very much it does, and I can assure you that protecting you and your family is at the top of their list. I would like to thank you again, and I want to ask you how this has helped you communicate with your loved ones? SP2: why are you making me cry again Interviewer: It's okay if you'd rather move on to a different subject. SP2: i can definitely answer that question i was just giving you a hard time to lighten the mood SP2: i really do not have much time and opportunity to use my home [humor] when i am relying on other people to translate what i say so please indulge my attempts at home [humor] because i really miss making jokes Interviewer: So tell me about communicating, how is this helping you? SP2: i have absolutely loved talking to my friends and family again without help from other people who can still understand to me SP2: so when my symptoms started my daughter was only **[redacted]**[2] and now she is **[redacted]**[2] and she doesn't remember what i sounded like before this disease took away my ability to talk normally and she was a little shy at first but now is super proud that her mother [father] is a robot SP2: that has been the lead [highlight] but i would also say that it is it possible [pretty cool] that i have been able to talk to other adults who do remember what i sounded like and they have been brought to tears to hear me again Interviewer: That's profound! SP2: so what people have told me is that they can totally understand what this is as a concept but when they see it in action it is [a] totally different type of experience Interviewer: Definitely. SP2: i have been able to talk to my parents over it and keep up with the conversation because they are from the south and talk really really really slowly SP2: so i have been able to use this device to help me communicate with my colleagues who are working far away from here and i am on mobile [meetings] with them and this will work fine to help me communicate on calls Interviewer: That's great. SP2: it really is because i have an awesome job and i feel like people have really invested in me to help make me who i am and i feel like i have a lot of really important work left to do and this will help me do it SP2: one of the things that people with my disease suffer from is isolation and depression because they do not feel like they matter anymore and something like this technology will help bring people back into life and into society SP2: that really cannot be understated how important that is SP2: because we will probably not find a cure for this disease but we will find medicines that help people live longer but if they are completely miserable than what is the point and something like this could really add value to people's life SP2: so i think about this from my personal perspective and also the perspective of people like me who might not be as lucky but deserve the same treatment SP2: does that make sense Interviewer: Yes it does, absolutely. SP2: what else would you like to know Interviewer: I think you covered pretty much most of the things I had in mind. I loved hearing what you had to say and engaging in this conversation with you which was amazing with	

	the help of this technology. And your sense of humor! If there's anything else that you would like to share, I am here to hear it. SP2: i hope that we are very close to the time when everyone who is in a position like me has the same option to have this device as i do Interviewer: I hope so too. I want to thank you so much, [SP2]. SP2: let's make it happen okay Interviewer: Yes, and thank you so much. SP2: my pleasure really Cameraman: That was wonderful, thanks for letting us into your home. SP2: of course my pleasure	

[1]Transcripts included here are non-exhaustive and exclude sensitive personal conversations where SP2 used the speech BCI to converse with friends, family members, or medical professionals. Selected transcripts show snippets of conversations SP2 had with the research team or others. Gaps in time between transcribed sentences are represented by ellipses (...). Incorrectly decoded words are colored red, followed by the word that SP2 meant to say (confirmed with him) in [green brackets].

[2]Potentially identifying information has been removed from these transcripts as per medrXiv policy.

References

1. Glasser MF, Coalson TS, Robinson EC, et al. A multi-modal parcellation of human cerebral cortex. Nature 2016;536(7615):171–8.
2. Harms MP, Somerville LH, Ances BM, et al. Extending the Human Connectome Project across ages: Imaging protocols for the Lifespan Development and Aging projects. NeuroImage 2018;183:972–84.
3. Willett FR, Kunz EM, Fan C, et al. A high-performance speech neuroprosthesis. Nature 2023;620(7976):1031–6.
4. Glasser MF, Coalson TS, Bijsterbosch JD, et al. Using temporal ICA to selectively remove global noise while preserving global signal in functional MRI data. NeuroImage 2018;181:692–717.
5. Salimi-Khorshidi G, Douaud G, Beckmann CF, Glasser MF, Griffanti L, Smith SM. Automatic denoising of functional MRI data: Combining independent component analysis and hierarchical fusion of classifiers. NeuroImage 2014;90:449–68.
6. Robinson EC, Garcia K, Glasser MF, et al. Multimodal surface matching with higher-order smoothness constraints. NeuroImage 2018;167:453–65.
7. Moses DA, Metzger SL, Liu JR, et al. Neuroprosthesis for Decoding Speech in a Paralyzed Person with Anarthria. N Engl J Med 2021;385(3):217–27.
8. Metzger SL, Liu JR, Moses DA, et al. Generalizable spelling using a speech neuroprosthesis in an individual with severe limb and vocal paralysis. Nat Commun 2022;13(1):6510.
9. Metzger SL, Littlejohn KT, Silva AB, et al. A high-performance neuroprosthesis for speech decoding and avatar control. Nature 2023;620(7976):1037–46.
10. Silva AB, Liu JR, Zhao L, Levy DF, Scott TL, Chang EF. A Neurosurgical Functional Dissection of the Middle Precentral Gyrus during Speech Production. J Neurosci 2022;42(45):8416–26.
11. Young D, Willett F, Memberg WD, et al. Signal processing methods for reducing artifacts in microelectrode brain recordings caused by functional electrical stimulation. J Neural Eng 2018;15(2):026014.
12. Trautmann EM, Stavisky SD, Lahiri S, et al. Accurate Estimation of Neural Population Dynamics without Spike Sorting. Neuron 2019;103(2):292-308.e4.
13. Chestek CA, Gilja V, Nuyujukian P, et al. Long-term stability of neural prosthetic control signals from silicon cortical arrays in rhesus macaque motor cortex. J Neural Eng 2011;8(4):045005.
14. Christie BP, Tat DM, Irwin ZT, et al. Comparison of spike sorting and thresholding of voltage waveforms for intracortical brain–machine interface performance. J Neural Eng 2015;12(1):016009.
15. Ali YH, Bodkin K, Rigotti-Thompson M, et al. BRAND: A platform for closed-loop experiments with deep network models [Internet]. 2023 [cited 2023 Dec 11];2023.08.08.552473. Available from: https://www.biorxiv.org/content/10.1101/2023.08.08.552473v1
16. Rubin DB, Ajiboye AB, Barefoot L, et al. Interim Safety Profile From the Feasibility Study of the BrainGate Neural Interface System. Neurology 2023;100(11):e1177–92.
17. Eren G¨olge. Coqui TTS [Internet]. 2021;Available from: https://github.com/coqui- ai/TTS.
18. Godfrey JJ, Holliman EC, McDaniel J. SWITCHBOARD: telephone speech corpus for research and development [Internet]. In: [Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing. 1992 [cited 2023 Dec 11]. p. 517–20 vol.1.Available from: https://ieeexplore.ieee.org/document/225858

19. Gao L, Biderman S, Black S, et al. The Pile: An 800GB Dataset of Diverse Text for Language Modeling [Internet]. 2020 [cited 2023 Dec 12];Available from: http://arxiv.org/abs/2101.00027
20. IEEE Recommended Practice for Speech Quality Measurements. IEEE No 297-1969 1969;1–24.
21. Park J, Kim K. g2pe [Internet]. 2019;Available from: https://github.com/Kyubyong/g2p
22. Fan C, Hahn N, Kamdar F, et al. Plug-and-Play Stability for Intracortical Brain-Computer Interfaces: A One-Year Demonstration of Seamless Brain-to-Text Communication [Internet]. 2023 [cited 2023 Dec 11];Available from: http://arxiv.org/abs/2311.03611
23. Lexical facts. The Economist [Internet] [cited 2023 Dec 12];Available from: https://www.economist.com/johnson/2013/05/29/lexical-facts?zid=319&ah=17af09b0281b01505c226b1e574f5cc1
24. Brysbaert M, Stevens M, Mandera P, Keuleers E. How Many Words Do We Know? Practical Estimates of Vocabulary Size Dependent on Word Definition, the Degree of Language Input and the Participant's Age. Front Psychol [Internet] 2016 [cited 2023 Dec 12];7. Available from: https://www.frontiersin.org/articles/10.3389/fpsyg.2016.01116
25. Valin J-M. A Hybrid DSP/Deep Learning Approach to Real-Time Full-Band Speech Enhancement [Internet]. In: 2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP). Vancouver, BC: IEEE; 2018 [cited 2023 Dec 12]. p. 1–5.Available from: https://ieeexplore.ieee.org/document/8547084/
26. Kim J, Kong J, Son J. Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech.
27. Simeral JD, Kim S-P, Black MJ, Donoghue JP, Hochberg LR. Neural control of cursor trajectory and click by a human with tetraplegia 1000 days after implant of an intracortical microelectrode array. J Neural Eng 2011;8(2):025027.
28. Pandarinath C, Nuyujukian P, Blabe CH, et al. High performance communication by people with paralysis using an intracortical brain-computer interface. eLife 2017;6:e18554.
29. Willett FR, Deo DR, Avansino DT, et al. Hand Knob Area of Premotor Cortex Represents the Whole Body in a Compositional Way. Cell 2020;181(2):396-409.e26.
30. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine Learning in Python. J Mach Learn Res 2011;12(85):2825–30.