Article

A high-performance neuroprosthesis for speech decoding and avatar control

https://doi.org/10.1038/s41586-023-06443-4

Received: 3 February 2023

Accepted: 17 July 2023

Published online: 23 August 2023

Check for updates

Sean L. Metzger[1,2,3,7], Kaylo T. Littlejohn[1,2,4,7], Alexander B. Silva[1,2,3,7], David A. Moses[1,2,7], Margaret P. Seaton[1,7], Ran Wang[1,2], Maximilian E. Dougherty[1], Jessie R. Liu[1,2,3], Peter Wu[4], Michael A. Berger[5], Inga Zhuravleva[4], Adelyn Tu-Chan[6], Karunesh Ganguly[2,6], Gopala K. Anumanchipalli[1,2,4] & Edward F. Chang[1,2,3] ✉

Speech neuroprostheses have the potential to restore communication to people living with paralysis, but naturalistic speed and expressivity are elusive[1]. Here we use high-density surface recordings of the speech cortex in a clinical-trial participant with severe limb and vocal paralysis to achieve high-performance real-time decoding across three complementary speech-related output modalities: text, speech audio and facial-avatar animation. We trained and evaluated deep-learning models using neural data collected as the participant attempted to silently speak sentences. For text, we demonstrate accurate and rapid large-vocabulary decoding with a median rate of 78 words per minute and median word error rate of 25%. For speech audio, we demonstrate intelligible and rapid speech synthesis and personalization to the participant's pre-injury voice. For facial-avatar animation, we demonstrate the control of virtual orofacial movements for speech and non-speech communicative gestures. The decoders reached high performance with less than two weeks of training. Our findings introduce a multimodal speech-neuroprosthetic approach that has substantial promise to restore full, embodied communication to people living with severe paralysis.

Speech is the ability to express thoughts and ideas through spoken words. Speech loss after neurological injury is devastating because it substantially impairs communication and causes social isolation[2]. Previous demonstrations have shown that it is possible to decode speech from the brain activity of a person with paralysis, but only in the form of text and with limited speed and vocabulary[1,3]. A compelling goal is to both enable faster large-vocabulary text-based communication and restore the produced speech sounds and facial movements related to speaking. Although text outputs are good for basic messages, speaking has rich prosody, expressiveness and identity that can enhance embodied communication beyond what can be conveyed in text alone. To address this, we designed a multimodal speech neuroprosthesis that uses broad-coverage, high-density electrocorticography (ECoG) to decode text and audio-visual speech outputs from articulatory vocal-tract representations distributed throughout the sensorimotor cortex (SMC). Owing to severe paralysis caused by a basilar-artery brainstem stroke that occurred more than 18 years ago, our 47-year-old participant cannot speak or vocalize speech sounds given the severe weakness of her orofacial and vocal muscles (anarthria; see Supplementary Note 1) and cannot type given the weakness in her arms and hands (quadriplegia). Instead, she has used commercial head-tracking assistive technology to communicate slowly to select letters at up to 14 words per minute (WPM; Supplementary Note 2). Here we demonstrate flexible, real-time decoding of brain activity into text, speech sounds, and both verbal and non-verbal orofacial movements. Additionally, we show that decoder performance is driven by broad coverage of articulatory representations distributed throughout the SMC that have persisted after years of paralysis.

Overview of multimodal speech-decoding system

We designed a speech-decoding system that enabled a clinical-trial participant (ClinicalTrials.gov; NCT03698149) with severe paralysis and anarthria to communicate by decoding intended sentences from signals acquired by a 253-channel high-density ECoG array implanted over speech cortical areas of the SMC and superior temporal gyrus (Fig. 1a–c). The array was positioned over cortical areas relevant for orofacial movements, and simple movement tasks demonstrated differentiable activations associated with attempted movements of the lips, tongue and jaw (Fig. 1d).

For speech decoding, the participant was presented with a sentence as a text prompt on a screen and was instructed to silently attempt to say the sentence after a visual go cue. Specifically, she attempted to silently speak the sentence without vocalizing any sounds. This differs from imagined or inner speech because she was trying to engage her articulators to the best of her ability, although substantial orofacial weakness prevents her from naturally mouthing words. Meanwhile, we processed neural signals recorded from all 253 ECoG electrodes

[1]Department of Neurological Surgery, University of California, San Francisco, San Francisco, CA, USA. [2]Weill Institute for Neuroscience, University of California, San Francisco, San Francisco, CA, USA. [3]University of California, Berkeley–University of California, San Francisco Graduate Program in Bioengineering, Berkeley, CA, USA. [4]Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA, USA. [5]Speech Graphics Ltd, Edinburgh, UK. [6]Department of Neurology, University of California, San Francisco, San Francisco, CA, USA. [7]These authors contributed equally: Sean L. Metzger, Kaylo T. Littlejohn, Alexander B. Silva, David A. Moses, Margaret P. Seaton. ✉e-mail: edward.chang@ucsf.edu

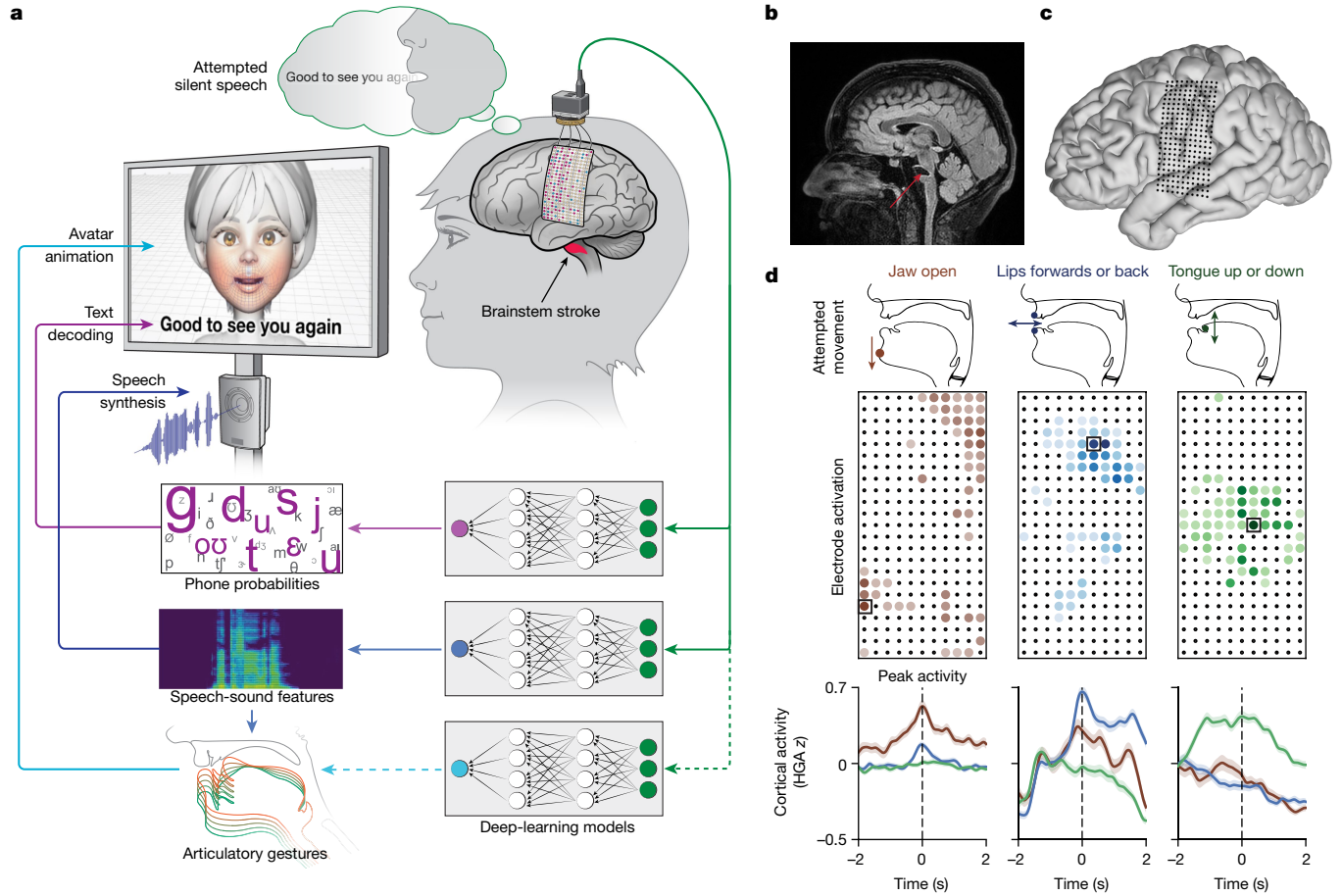


Fig. 1 | Multimodal speech decoding in a participant with vocal-tract paralysis. a, Overview of the speech-decoding pipeline. A brainstem-stroke survivor with anarthria was implanted with a 253-channel high-density ECoG array 18 years after injury. Neural activity was processed and used to train deep-learning models to predict phone probabilities, speech-sound features and articulatory gestures. These outputs were used to decode text, synthesize audible speech and animate a virtual avatar, respectively. **b**, A sagittal magnetic resonance imaging scan showing brainstem atrophy (in the bilateral pons; red arrow) resulting from stroke. **c**, Magnetic resonance imaging reconstruction of the participant's brain overlaid with the locations of implanted electrodes. The ECoG array was implanted over the participant's lateral cortex, centred on the central sulcus. **d**, Top: simple articulatory movements attempted by the participant. Middle: Electrode-activation maps demonstrating robust electrode tunings across articulators during attempted movements. Only the electrodes with the strongest responses (top 20%) are shown for each movement type. Colour indicates the magnitude of the average evoked HGA response with each type of movement. Bottom: *z*-scored trial-averaged evoked HGA responses with each movement type for each of the outlined electrodes in the electrode-activation maps. In each plot, each response trace shows mean ± standard error across trials and is aligned to the peak-activation time ($n = 130$ trials for jaw open, $n = 260$ trials each for lips forwards or back and tongue up or down).

to extract high-gamma activity (HGA; between 70 and 150 Hz) and low-frequency signals (between 0.3 and 17 Hz)[3]. We trained deep-learning models to learn mappings between these ECoG features and phones, speech-sound features and articulatory gestures, which we then used to output text, synthesize speech audio and animate a virtual avatar, respectively (Fig. 1a and Supplementary Video 1).

We evaluated our system using three custom sentence sets containing varying amounts of unique words and sentences named 50-phrase-AAC, 529-phrase-AAC and 1024-word-General. The first two sets closely mirror corpora preloaded on commercially available augmentative and alternative communication (AAC) devices, designed to let patients express basic concepts and caregiving needs[4]. We chose these two sets to assess our ability to decode high-utility sentences at a limited and expanded vocabulary level. The 529-phrase-AAC set contained 529 sentences composed of 372 unique words, and from this set we subselected 50 high-utility sentences composed of 119 unique words to create the 50-phrase-AAC set. To evaluate how well our system performed with a larger vocabulary containing common English words, we created the 1024-word-General set, containing 9,655 sentences composed of 1,024 unique words sampled from Twitter and film transcriptions. We primarily used this set to assess how well our decoders could generalize to sentences that the participant did not attempt to say during training with a vocabulary size large enough to facilitate general-purpose communication (Method 1 in Supplementary Methods).

To train our neural-decoding models before real-time testing, we recorded ECoG data as the participant silently attempted to speak individual sentences. A major difficulty in learning statistical mappings between the ECoG features and the sequences of phones and speech-sound features in the sentences was caused by the absence of clear timing information of words and phonemes in the silently attempted speech. To overcome this, we used a connectionist temporal classification (CTC) loss function during training of our neural decoders, which is commonly used in automatic speech recognition to infer sequences of sub-word units (such as phones or letters) from speech waveforms when precise time alignment between the units and the waveforms is unknown[5]. We used CTC loss during training of the text, speech and articulatory decoding models to enable prediction of phone probabilities, discrete speech-sound units and discrete articulator movements, respectively, from the ECoG signals.

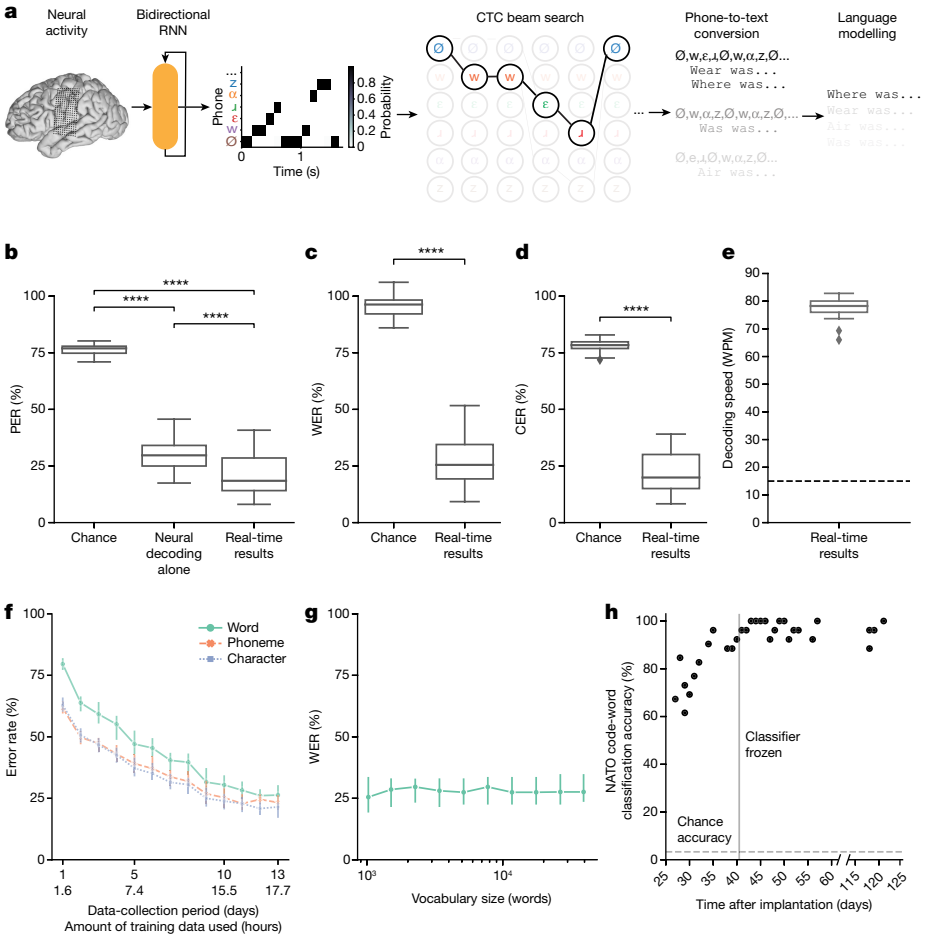


Fig. 2 | High-performance text decoding from neural activity. a, During attempts by the participant to silently speak, a bidirectional RNN decodes neural features into a time series of phone and silence (denoted as Ø) probabilities. From these probabilities, a CTC beam search computes the most likely sequence of phones that can be translated into words in the vocabulary. An *n*-gram language model rescores sentences created from these sequences to yield the most likely sentence. **b**, Median PERs, calculated using shuffled neural data (Chance), neural decoding without applying vocabulary constraints or language modelling (Neural decoding alone) and the full real-time system (Real-time results) across $n = 25$ pseudo-blocks. **c**,**d**, Word (**c**) and character (**d**) error rates for chance and real-time results. In **b**–**d**, ****$P < 0.0001$, two-sided Wilcoxon signed-rank test with five-way Holm–Bonferroni correction for multiple comparisons; *P* values and statistics in Extended Data Table 1. **e**, Decoded WPM. Dashed line denotes previous state-of-the-art speech BCI decoding rate in a person with paralysis[1]. **f**, Offline evaluation of error rates as a function of training-data quantity. **g**, Offline evaluation of WER as a function of the number of words used to apply vocabulary constraints and train the language model. Error bars in **f**,**g** represent 99% CIs of the median, calculated using 1,000 bootstraps across $n = 125$ pseudo-blocks (**f**) and $n = 25$ pseudo-blocks (**g**) at each point. **h**, Decoder stability as assessed using real-time classification accuracy during attempts to silently say 26 NATO code words across days and weeks. The vertical line represents when the classifier was no longer retrained before each session. In **b**–**g**, results were computed using the real-time evaluation trials with the 1024-word-General sentence set. Box plots in all figures depict median (horizontal line inside box), 25th and 75th percentiles (box) ± 1.5 times the interquartile range (whiskers) and outliers (diamonds).

Text decoding

Text-based communication is an important modality for facilitating messaging and interaction with technology. Initial efforts to decode text from the brain activity of a person with anarthria during attempted speech had various limitations, including slow decoding rates and small vocabulary sizes[1,3]. Here we address these limitations by implementing a flexible approach using phone decoding, enabling decoding of arbitrary phrases from large vocabularies while approaching naturalistic speaking rates.

To evaluate real-time performance, we decoded text as the participant attempted to silently say 249 randomly selected sentences from the 1024-word-General set that were not used during model training (Fig. 2a and Supplementary Video 2). To decode text, we streamed features extracted from ECoG signals starting 500 ms before the go cue into a bidirectional recurrent neural network (RNN). Before testing, we trained the RNN to predict the probabilities of 39 phones and silence at each time step. A CTC beam search then determined the most likely sentence given these probabilities. First, it created a set of candidate phone sequences that were constrained to form valid words within the 1,024-word vocabulary. Then, it evaluated candidate sentences by combining each candidate's underlying phone probabilities with its linguistic probability using a natural-language model.

To quantify text-decoding performance, we used standard metrics in automatic speech recognition: word error rate (WER), phone error rate (PER), character error rate (CER) and WPM. WER, PER and CER measure the percentage of decoded words, phones and characters, respectively, that were incorrect.

We computed error rates across sequential pseudo-blocks of ten-sentence segments (and one pseudo-block of nine sentences) using text decoded during real-time evaluation (Method 1 in Supplementary Methods). We achieved a median PER of 18.5% (99% confidence interval (CI) [14.1, 28.5]; Fig. 2b), a median WER of 25.5% (99% CI [19.3, 34.5]; Fig. 2c) and a median CER of 19.9% (99% CI [15.0, 30.1]; Fig. 2d; see Table 1

Table 1 | Illustrative text-decoding examples for the 1024-word-General set

Target sentence	Decoded sentence	WER (%)	Percentile (%)
You should have let me do the talking	You should have let me do the talking	0	44.6
I think I need a little air	I think I need a little air	0	44.6
Do you want to get some coffee	Do you want to get some coffee	0	44.6
What do you get if you finish	Why do you get if you finish	14	47.0
Did you know him very well	Did you know him well	17	49.4
You got your wish	You get your wish	25	61.8
No tell me why	So tell me why	25	61.8
You have no right to keep us here	You have no right to be out here	25	61.8
Why would they come to me	Why would they have to be	33	65.1
Come here I want to show you something	Have here I want to do something	38	65.5
All I told them was the truth	Can I do that was the truth	43	70.3
You got it all in your head	You got here all your right	43	70.3
Is she a friend of yours	I see afraid of yours	67	85.1
How is your cold	Your old	75	89.2

Examples are shown for various levels of WER during real-time decoding with the 1024-word-General set. Each percentile value indicates the percentage of decoded sentences that had a WER less than or equal to the WER of the provided example sentence.

for example decodes; see Extended Data Fig. 1 for the relationship between decoded PER and WER). For all metrics, performance was better than chance, which we computed by re-evaluating performance after using temporally shuffled neural data as the input to our decoding pipeline ($P < 0.0001$ for all three comparisons, two-sided Wilcoxon rank-sum tests with five-way Holm–Bonferroni correction). The average WER passes the 30% threshold below which speech-recognition applications generally become useful[6] while providing access to a large vocabulary of over 1,000 words, indicating that our approach may be viable in clinical applications.

To probe whether decoding performance was dependent on the size of the vocabulary used to constrain model outputs and train the language model, we measured decoding performance in offline simulations using log-spaced vocabulary sizes ranging from 1,506 to 39,378 words. We created each vocabulary by augmenting the 1024-word-General vocabulary with the $n - 1{,}024$ most frequently occurring words outside this set in large-scale corpora, in which n is the size of the vocabulary. Then, for each vocabulary, we retrained the natural-language model to incorporate the new words and enabled the model to output any word from the larger vocabulary, and then carried out decoding with the real-time evaluation trials. We observed robust decoding performance as vocabulary size grew (Fig. 2g; see Extended Data Fig. 2 for CER and PER). With a vocabulary of 39,378 words, we achieved a median offline WER of 27.6% (99% CI [20.0 34.7]).

We verified that our system remained functional in a freeform setting in which the participant volitionally and spontaneously attempted to silently say unprompted sentences, with the neural data aligned to speech onsets detected directly from the neural features instead of to go cues (Method 2 in Supplementary Methods and Supplementary Video 3).

We observed a median real-time decoding rate of 78.3 WPM (99% CI [75.5, 79.4]; Fig. 2f). This decoding rate exceeds our participant's typical communication rate using her assistive device (14.2 WPM; Supplementary Note 2) and is closer to naturalistic speaking rates than has been previously reported with communication neuroprostheses[1,3,7–9].

To assess how well our system could decode phones in the absence of a language model and constrained vocabulary, we evaluated performance using just the RNN neural-decoding model (using the most likely phone prediction at each time step) in an offline analysis. This yielded a median PER of 29.4% (99% CI [26.2, 32.8]; Fig. 2b), which is only 10.9 percentage points higher than that of the full model, demonstrating that the primary contributor to phone-decoding performance was the neural-decoding RNN model and not the CTC beam search or language model ($P < 0.0001$ for all comparisons to chance and to the full model, two-sided Wilcoxon signed-rank tests with five-way Holm–Bonferroni correction; Extended Data Table 1).

We also characterized the relationship between quantity of training data and text-decoding performance in offline analyses. For each day of data collection, we trained five models with different random initializations on all of the data collected on or before that date, and then simulated performance on the real-time blocks. We observed steadily declining error rates over the course of 13 days of training-data collection (Fig. 2f), during which we collected 9,506 sentence trials corresponding to about 1.6 h of training data per day. These results show that functional speech-decoding performance can be achieved after a relatively short period of data collection compared to that of our previous work[1,3] and is likely to continue to improve with more data.

To assess signal stability, we measured real-time classification performance during a separate word and motor task that we collected data for during each research session with our participant. In each trial of this task, we prompted the participant to either attempt to silently say one of the 26 code words from the NATO (North Atlantic Treaty Organization) phonetic alphabet (alpha, bravo, charlie and so forth) or attempt one of four hand movements (described and analysed in a later section). We trained a neural-network classifier to predict the most likely NATO code word from a 4-s window of ECoG features (aligned to the task go cue) and evaluated real-time performance with the classifier during the NATO-motor task (Fig. 2g and Supplementary Video 4). We continued to retrain the model using data available prior to real-time testing until day 40, at which point we froze the classifier after training it on data from the 1,196 available trials. Across 19 sessions after freezing the classifier, we observed a mean classification accuracy of 96.8% (99% CI [94.5, 98.6]), with accuracies of 100% obtained on eight of these sessions. Accuracy remained high after a 61-day hiatus in recording for the participant to travel. These results illustrate the stability of the cortical-surface neural interface without requiring recalibration and demonstrate that high performance can be achieved with relatively few training trials.

To evaluate model performance on predefined sentence sets without any pausing between words, we trained text-decoding models on neural data recorded as the participant attempted to silently say sentences from the 50-phrase-AAC and 529-phrase-AAC sets, and then simulated offline text decoding with these sets (Extended Data Figs. 3 and 4 and Method 1 in Supplementary Methods). With the 529-phrase-AAC set, we observed a median WER of 17.1% across sentences (99% CI [8.89%, 28.9%]), with a median decoding rate of 89.9 WPM (99% CI [83.6, 93.3]). With the 50-phrase-AAC set, we observed a median WER of 4.92% (99% CI [3.18, 14.04]) with median decoding speeds of 101 WPM (99% CI [95.6, 103]). PERs and CERs for each set are given in Extended Data Figs. 3 and 4. These results illustrate extremely rapid and accurate decoding for finite, predefined sentences that could be used frequently by users.

Speech synthesis

An alternative approach to text decoding is to synthesize speech sounds directly from recorded neural activity, which could offer a pathway towards more naturalistic and expressive communication for someone

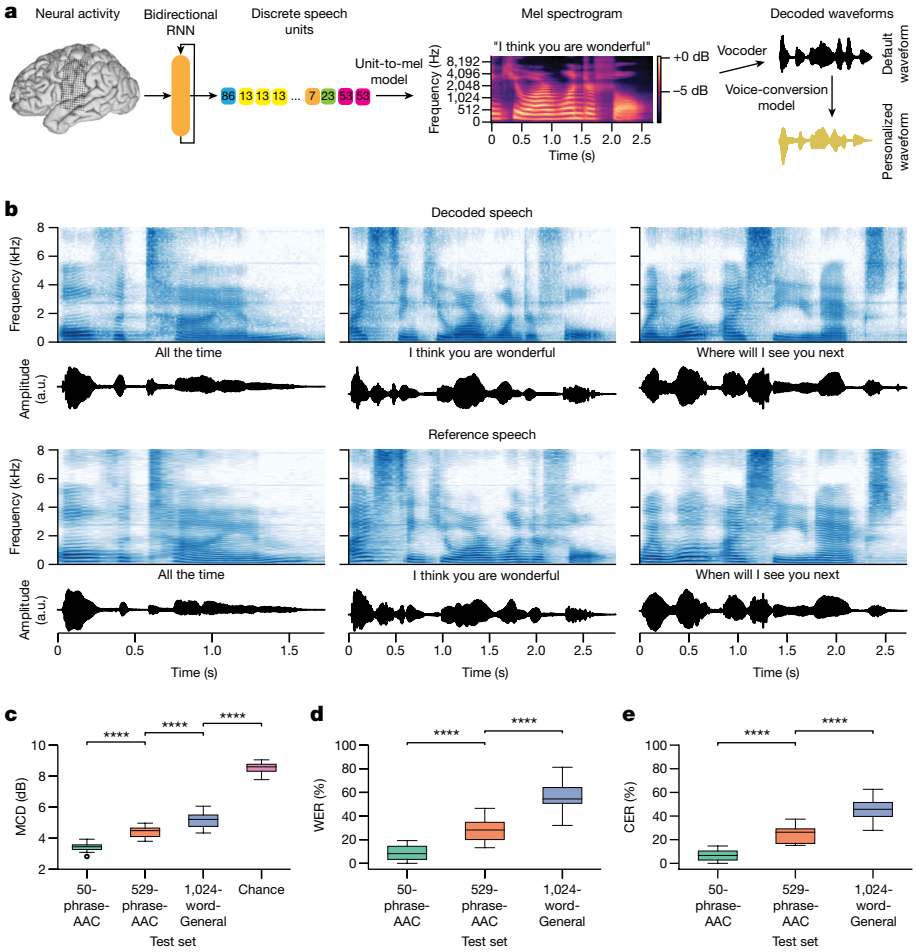


Fig. 3 | Intelligible speech synthesis from neural activity. a, Schematic diagram of the speech-synthesis decoding algorithm. During attempts by the participant to silently speak, a bidirectional RNN decodes neural features into a time series of discrete speech units. The RNN was trained using reference speech units computed by applying a large pretrained acoustic model (HuBERT) on basis waveforms. Predicted speech units are then transformed into the mel spectrogram and vocoded into audible speech. The decoded waveform is played back to the participant in real time after a brief delay. Offline, the decoded speech was transformed to be in the participant's personalized synthetic voice using a voice-conversion model. **b**, Top two rows: three example decoded spectrograms and accompanying perceptual transcriptions (top) and waveforms (bottom) from the 529-phrase-AAC sentence set. Bottom two rows: the corresponding reference spectrograms, transcriptions and waveforms representing the decoding targets. **c**, MCDs for the decoded waveforms during real-time evaluation with the three sentence sets and from chance waveforms computed offline. Lower MCD indicates better performance. Chance waveforms were computed by shuffling electrode indices in the test data for the 50-phrase-AAC set with the same synthesis pipeline. **d**, Perceptual WERs from untrained human evaluators during a transcription task. **e**, Perceptual CERs from the same human-evaluation results as **d**. In **c**–**e**, ****$P < 0.0001$, Mann–Whitney U-test with 19-way Holm–Bonferroni correction for multiple comparisons; all non-adjacent comparisons were also significant ($P < 0.0001$; not depicted); $n = 15$ pseudo-blocks for the AAC sets, $n = 20$ pseudo-blocks for the 1024-word-General set. P values and statistics in Extended Data Table 2. In **b**–**e**, all decoded waveforms, spectrograms and quantitative results use the non-personalized voice (see Extended Data Fig. 5 and Supplementary Table 1 for results with the personalized voice). A.u., arbitrary units.

who is unable to speak. Previous work in speakers with intact speech has demonstrated that intelligible speech can be synthesized from neural activity during vocalized or mimed speech[10,11], but this has not been shown with someone who is paralysed.

We carried out real-time speech synthesis by transforming the participant's neural activity directly into audible speech as she attempted to silently speak during the audio-visual task condition (Fig. 3a and Supplementary Videos 5 and 6). To synthesize speech, we passed time windows of neural activity around the go cue into a bidirectional RNN. Before testing, we trained the RNN to predict the probabilities of 100 discrete speech units at each time step. To create the reference speech-unit sequences for training, we used HuBERT, a self-supervised speech-representation learning model[12] that encodes a continuous speech waveform into a temporal sequence of discrete speech units that captures latent phonetic and articulatory representations[13]. Because our participant cannot speak, we acquired reference speech waveforms from a recruited speaker for the AAC sentence sets or using a text-to-speech algorithm for the 1024-word-General set. We used a CTC loss function during training to enable the RNN to learn mappings between the ECoG features and speech units derived from these reference waveforms without alignment between our participant's silent-speech attempts and the reference waveforms. After predicting the unit probabilities, we passed the most likely unit at each time step into a pretrained unit-to-speech model that first generated a mel spectrogram and then vocoded this mel spectrogram into an audible speech waveform in real time[14,15]. Offline, we used a voice-conversion model trained on a brief segment of the participant's speech (recorded before her injury) to process the decoded speech into the participant's own personalized synthetic voice (Supplementary Video 7).

We qualitatively observed that spectrograms decoded in real time shared both fine-grained and broad timescale information with corresponding reference spectrograms (Fig. 3b). To quantitatively

assess the quality of the decoded speech, we used the mel-cepstral distortion (MCD) metric, which measures the similarity between two sets of mel-cepstral coefficients (which are speech-relevant acoustic features) and is commonly used to evaluate speech-synthesis performance[16]. Lower MCD indicates stronger similarity. We achieved mean MCDs of 3.45 (99% CI [3.25, 3.82]), 4.49 (99% CI [4.07, 4.67]) and 5.21 (99% CI [4.74, 5.51]) dB for the 50-phrase-AAC, 529-phrase-AAC and 1024-word-General sets, respectively (Fig. 3c). We observed similar MCD performance on the participant's personalized voice (Extended Data Fig. 5 and Supplementary Table 1). Performance increased as the number of unique words and sentences in the sentence set decreased but was always better than chance (all $P < 0.0001$, two-sided Wilcoxon rank-sum tests with 19-way Holm–Bonferroni correction; chance MCDs were measured using waveforms generated by passing temporally shuffled ECoG features through the synthesis pipeline). Furthermore, these MCDs are comparable to those observed with text-to-speech synthesizers[16] and better than those in previous neural-decoding work with participants that were able to speak naturally[11].

Human-transcription assessments are a standard method to quantify the perceptual accuracy of synthesized speech[17]. To directly assess the intelligibility of our synthesized speech waveforms, crowd-sourced evaluators listened to the synthesized speech waveforms and then transcribed what they heard into text. We then computed perceptual WERs and CERs by comparing these transcriptions to the ground-truth sentence texts. We achieved median WERs of 8.20% (99% CI [3.28, 14.5]), 28.2% (99% CI [18.6, 38.5]) and 54.4% (99% CI [50.5 65.2]) and median CERs of 6.64% (99% CI [2.71, 10.6]), 26.3% (99% CI [15.9, 29.7]) and 45.7% (99% CI [39.2, 51.6]) across test trials for the 50-phrase-AAC, 529-phrase-AAC and 1024-word-General sets, respectively (Fig. 3d,e; see Supplementary Table 2 for examples of perceptual transcriptions alongside MCD and Extended Data Fig. 6 for correlations between WER and MCD). As for the MCD results, WERs and CERs improved as the number of unique words and sentences in the sentence set decreased (all $P < 0.0001$, two-sided Wilcoxon rank-sum tests with 19-way Holm–Bonferroni correction; chance measured by shuffling the mapping between the transcriptions and the ground-truth sentence texts). Together, these results demonstrate that it is possible to synthesize intelligible speech from the brain activity of a person with paralysis.

Facial-avatar decoding

Face-to-face audio-visual communication offers multiple advantages over solely audio-based communication. Previous studies show that non-verbal facial gestures often account for a substantial portion of the perceived feeling and attitude of a speaker[18,19] and that face-to-face communication enhances social connectivity[20] and intelligibility[21]. Therefore, animation of a facial avatar to accompany synthesized speech and further embody the user is a promising means towards naturalistic communication, and it may be possible via decoding of articulatory and orofacial representations in the speech-motor cortex[22–25]. To this end, we developed a facial-avatar brain–computer interface (BCI) to decode neural activity into articulatory speech gestures and render a dynamically moving virtual face during the audio-visual task condition (Fig. 4a).

To synthesize the avatar's motion, we used an avatar-animation system designed to transform speech signals into accompanying facial-movement animations for applications in games and film (Speech Graphics). This technology uses speech-to-gesture methods that predict articulatory gestures (Method 5 in the Supplementary Methods) from sound waveforms and then synthesizes the avatar animation from these gestures[26]. We designed a three-dimensional (3D) virtual environment to display the avatar to our participant during testing. Before testing, the participant selected an avatar from multiple potential candidates.

We implemented two approaches for animating the avatar: a direct approach and an acoustic approach. We used the direct approach for offline analyses to evaluate whether articulatory movements could be directly inferred from neural activity without the use of a speech-based intermediate, which has implications for potential future uses of an avatar that are not based on speech representations, including non-verbal facial expressions. We used the acoustic approach for real-time audio-visual synthesis because it provided low-latency synchronization between decoded speech audio and avatar movements.

For the direct approach, we trained a bidirectional RNN with CTC loss to learn a mapping between ECoG features and reference discretized articulatory gestures. These articulatory gestures were obtained by passing the reference acoustic waveforms through the animation system's speech-to-gesture model. We then discretized the articulatory gestures using a vector-quantized variational autoencoder (VQ-VAE)[27]. During testing, we used the RNN to decode the discretized articulatory gestures from neural activity and then dequantized them into continuous articulatory gestures using the VQ-VAE's decoder. Finally, we used the gesture-to-animation subsystem to animate the avatar face from the continuous gestures.

We found that the direct approach produced articulatory gestures that were strongly correlated with reference articulatory gestures across all datasets (Supplementary Figs. 1 and 2 and Supplementary Table 4), highlighting the system's ability to decode articulatory information from brain activity.

We then evaluated direct-decoding results by measuring the perceptual accuracy of the avatar. Here we used a forced-choice perceptual assessment to test whether the avatar animations contained visually salient information about the target utterance. Crowd-sourced evaluators watched silent videos of the decoded avatar animations and were asked to identify to which of two sentences each video corresponded. One sentence was the ground-truth sentence and the other was randomly selected from the set of test sentences. We used the median bootstrapped accuracy across six evaluators to represent the final accuracy for each sentence. We obtained median accuracies of 85.7% (99% CI [79.0, 92.0]), 87.7% (99% CI [79.7, 93.7]) and 74.3% (99% CI [66.7, 80.8]) across the 50-phrase-AAC, 529-phrase-AAC and 1024-word-General sets, demonstrating that the avatar conveyed perceptually meaningful speech-related facial movements (Fig. 4b).

Next, we compared the facial-avatar movements generated during direct decoding with real movements made by healthy speakers. We recorded videos of eight healthy volunteers as they read aloud sentences from the 1024-word-General set. We then applied a facial-keypoint recognition model (dlib)[28] to avatar and healthy-speaker videos to extract trajectories important for speech: jaw opening, lip aperture and mouth width. For each pseudo-block of ten test sentences, we computed the mean correlations across sentences between the trajectory values for each possible pair of corresponding videos (36 total combinations with 1 avatar and 8 healthy-speaker videos). Before calculating correlations between two trajectories for the same sentence, we applied dynamic time warping to account for variability in timing. We found that the jaw opening, lip aperture and mouth width of the avatar and healthy speakers were well correlated with median values of 0.733 (99% CI [0.711, 0.748]), 0.690 (99% CI [0.663, 0.714]) and 0.446 (99% CI [0.417, 0.470]), respectively (Fig. 4c). Although correlations among pairs of healthy speakers were higher than between the avatar and healthy speakers (all $P < 0.0001$, two-sided Mann–Whitney U-test with nine-way Holm–Bonferroni correction; Supplementary Table 3), there was a large degree of overlap between the two distributions, illustrating that the avatar reasonably approximated the expected articulatory trajectories relative to natural variances between healthy speakers. Correlations for both distributions were significantly above chance, which was calculated by temporally shuffling the human trajectories and then recomputing correlations with dynamic time warping (all $P < 0.0001$, two-sided Mann–Whitney U-test with nine-way Holm–Bonferroni correction; Supplementary Table 3).

Avatar animations rendered in real time using the acoustic approach also exhibited strong correlations between decoded and reference

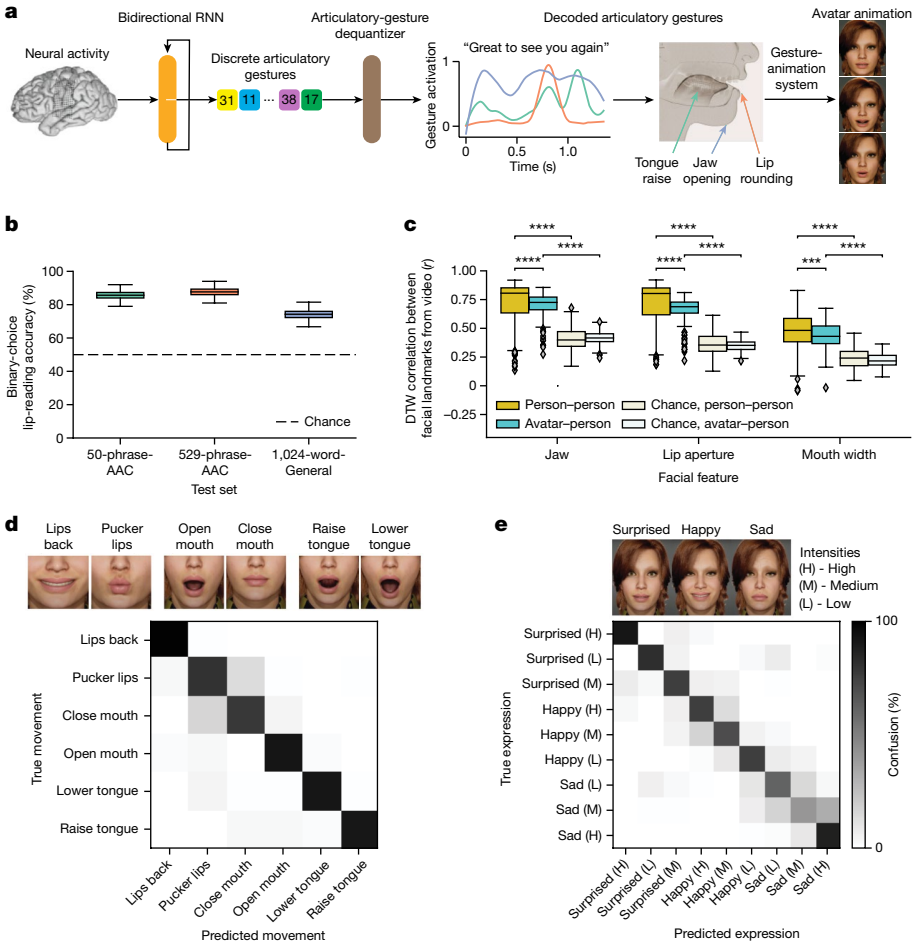


Fig. 4 | Direct decoding of orofacial articulatory gestures from neural activity to drive an avatar. a, Schematic diagram of the avatar-decoding algorithm. Offline, a bidirectional RNN decodes neural activity recorded during attempts to silently speak into discretized articulatory gestures (quantized by a VQ-VAE). A convolutional neural network dequantizer (VQ-VAE decoder) is then applied to generate the final predicted gestures, which are then passed through a pretrained gesture-animation model to animate the avatar in a virtual environment. **b**, Binary perceptual accuracies from human evaluators on avatar animations generated from neural activity, $n = 2{,}000$ bootstrapped points. **c**, Correlations after applying dynamics time warping (DTW) for jaw, lip and mouth-width movements between decoded avatar renderings and videos of real human speakers on the 1024-word-General sentence set across all pseudo-blocks for each comparison ($n = 152$ for avatar–person comparison, $n = 532$ for person–person comparisons; $^{****}P < 0.0001$, Mann–Whitney U-test with nine-way Holm–Bonferroni correction; P values and U-statistics in Supplementary Table 3). A facial-landmark detector (dlib) was used to measure orofacial movements from the videos. **d**, Top: snapshots of avatar animations of six non-speech articulatory movements in the articulatory-movement task. Bottom: confusion matrix depicting classification accuracy across the movements. The classifier was trained to predict which movement the participant was attempting from her neural activity, and the prediction was used to animate the avatar. **e**, Top: snapshots of avatar animations of three non-speech emotional expressions in the emotional-expression task. Bottom: confusion matrix depicting classification accuracy across three intensity levels (high, medium and low) of the three expressions, ordered using a hierarchical agglomerative clustering on the confusion values. The classifier was trained to predict which expression the participant was attempting from her neural activity, and the prediction was used to animate the avatar.

articulatory gestures (Supplementary Fig. 3 and Supplementary Table 5), high perceptual accuracy (Supplementary Fig. 4) and visual facial-landmark trajectories that were closely correlated with healthy-speaker trajectories (Supplementary Fig. 5 and Supplementary Table 6). These findings emphasize the strong performance of the speech-synthesis neural decoder when used with the speech-to-gesture rendering system, although this approach cannot be used to generate meaningful facial gestures in the absence of a decoded speech waveform.

In addition to articulatory gestures to visually accompany synthesized speech, a fully embodying avatar BCI would also enable the user to portray non-speech orofacial gestures, including movements of particular orofacial muscles and expressions that convey emotion[29]. To this end, we collected neural data from our participant as she carried out two additional tasks: an articulatory-movement task and an emotional-expression task. In the articulatory-movement task, the participant attempted to produce six orofacial movements (Fig. 4d). In the emotional-expression task, the participant attempted to produce three types of expression—happy, sad and surprised—with either low, medium or high intensity, resulting in nine unique expressions. Offline, for the articulatory-movement task, we trained a small feedforward neural-network model to learn the mapping between the ECoG features and each of the targets. We observed a median classification accuracy of 87.8% (99% CI [85.1, 90.5]; across $n = 10$ cross-validation folds; Fig. 4d) when classifying between the six articulatory movements. For the emotional-expression task, we trained a small RNN to learn the mapping between ECoG features and each of the expression targets. We observed a median classification accuracy of 74.0% (99% CI [70.8, 77.1]; across $n = 15$ cross-validation folds; Fig. 4e) when classifying between the nine possible expressions and a median classification accuracy of 96.9% (99% CI [93.8,100]) when considering the

classifier's outputs for only the strong-intensity versions of the three expression types (Supplementary Fig. 6). In separate, qualitative task blocks, we showed that the participant could control the avatar BCI to portray the articulatory movements (Supplementary Video 8) and strong-intensity emotional expressions (Supplementary Video 9), illustrating the potential of multimodal communication BCIs to restore the ability to express meaningful orofacial gestures.

Articulatory representations drive decoding

In healthy speakers, neural representations in the SMC (comprising the precentral and postcentral gyri) encode articulatory movements of the orofacial musculature[22,24,30]. With the implanted electrode array centred over the SMC of our participant, we reasoned that articulatory representations persisting after paralysis drove speech-decoding performance. To assess this, we fitted a linear temporal receptive-field encoding model to predict HGA for each electrode from the phone probabilities computed by the text decoder during the 1024-word-General text task condition. For each speech-activated electrode, we calculated the maximum encoding weight for each phone, yielding a phonetic-tuning space in which each electrode had an associated vector of phone-encoding weights. Within this space, we determined whether phone clustering was organized by the primary orofacial articulator of each phone (place of articulation (POA); Fig. 5a), which has been shown in previous studies with healthy speakers[22,23]. We parcelled phones into four POA categories: labial, vocalic, back tongue and front tongue. Hierarchical clustering of phones revealed grouping by POA ($P < 0.0001$ compared to chance, one-tailed permutation test; Fig. 5b). We observed a variety of tunings across the electrodes, with some electrodes exhibiting tuning to single POA categories and others to multiple categories (such as both front-tongue and back-tongue phones or both labial and vocalic phones; Fig. 5c and Supplementary Fig. 7). We visualized the phonetic tunings in a 2D space, revealing separability between labial and non-labial consonants (Fig. 5d) and between lip-rounded and non-lip-rounded vowels (Fig. 5e).

Next we investigated whether these articulatory representations were arranged somatotopically (with ordered regions of cortex preferring single articulators), which is observed in healthy speakers[23]. As the dorsal-posterior corner of our ECoG array provided coverage of the hand cortex, we also assessed how neural activation patterns related to attempted hand movements fit into the somatotopic map, using data collected during the NATO-motor task containing four finger-flexion targets (either thumb or simultaneous index- and middle-finger flexion for each hand). We visualized the grid locations of the electrodes that most strongly encoded the vocalic, front-tongue and labial phones as well attempted hand movement (the top 30% of electrodes having maximal tuning for each condition; Fig. 5f; see Supplementary Fig. 8 for full electrode encoding maps). Kernel density estimates revealed a somatotopic map with encoding of attempted hand movements, labial phones and front-tongue phones organized along a dorsal–ventral axis. The relatively anterior localization of the vocalic cluster in the precentral gyrus is probably associated with the laryngeal motor cortex, consistent with previous investigations in healthy speakers[23,24,31].

Next we assessed whether the same electrodes that encoded POA categories during silent-speech attempts also encoded non-speech articulatory-movement attempts. Using the previously computed phonetic encodings and HGA recorded during the articulatory-movement task, we found a positive correlation between front-tongue phonetic encoding and HGA magnitude during attempts to raise the tongue ($P < 0.0001$, $r = 0.84$, ordinary least-squares regression; Fig. 5g). We also observed a positive correlation between labial phonetic tuning and HGA magnitude during attempts to pucker the lips ($P < 0.0001$, $r = 0.89$, ordinary least-squares regression; Fig. 5h). Although most electrodes were selective to either lip or tongue movements, others were activated by both (Fig. 5i). Together, these findings suggest that, after 18 years of paralysis, our participant's SMC maintains general-purpose articulatory encoding that is not speech specific and contains representations of non-verbal emotional expressions and articulatory movements (see Fig. 4). During the NATO-motor task, electrodes encoding attempted finger flexions were largely orthogonal to those encoding NATO code words, which helped to enable accurate neural discrimination between the four finger-flexion targets and the silent-speech targets (the model correctly classified 569 out of 570 test trials as either finger flexion or silent speech; Supplementary Fig. 9).

To characterize the relationship between encoding strength and importance during decoding, we computed a contribution score for each electrode and decoding modality by measuring the effect of small perturbations to the electrode's activity on decoder predictions, as in previous work[1,3,32] (Extended Data Fig. 7a–c). We noted that many important electrodes were adjacent, suggesting sampling of useful, non-redundant information from the cortex despite the electrodes' close proximity. We also observed degraded performance during an offline simulation of low-density sampling (Supplementary Figs. 10 and 11 and Supplementary Table 8), further highlighting the benefit of high-density cortical recording. As we reasoned, many of the highest-contributing electrodes also exhibited substantial articulatory-feature encoding defined in Fig. 5 (Supplementary Figs. 12 and 13) and were similarly important for all three modalities (Extended Data Fig. 7e–g). Indeed, the brain areas that most strongly encoded POA, notably the SMC, were the most critical to decoding performance in leave-one-area-out offline analyses (Extended Data Fig. 8, Supplementary Fig. 14 and Supplementary Table 8).

These results are in line with growing evidence for motor-movement encoding in the postcentral gyrus[33–35], which is further supported by an analysis of peak-activation times that revealed no significant difference between electrodes in the precentral versus postcentral gyrus during silent attempts to speak (Supplementary Fig. 15; $P > 0.01$ two-sided Mann–Whitney U-test)[33–35]. We found that some temporal-lobe electrodes were not only active during passive listening but also contributed to silently attempted speech decoding ($r \geq 0.55$, $P < 0.0001$, Pearson correlation permutation test; Supplementary Fig. 16), suggesting that they may record cortical activity from the subcentral gyrus[36] or sites with production activity within the temporal lobe[37].

Discussion

Faster, more accurate, and more natural communication are among the most desired needs of people who have lost the ability to speak after severe paralysis[2,38–40]. Here we have demonstrated that all of these needs can be addressed with a speech-neuroprosthetic system that decodes articulatory cortical activity into multiple output modalities in real time, including text, speech audio synchronized with a facial avatar, and facial expressions.

During 14 days of data collection shortly after device implantation, we achieved high-performance text decoding, exceeding communication speeds of previous BCIs by a factor of 4 or more[1,3,9] and expanding the vocabulary size of our previous direct-speech BCI by a factor of 20 (ref. 1). We also showed that intelligible speech can be synthesized from the brain activity of a person with paralysis. Finally, we introduced a modality of BCI control in the form of a digital 'talking face'—a personalized avatar capable of dynamic, realistic and interpretable speech and non-verbal facial gestures. We believe that, together, these results have surpassed an important threshold of performance, generalizability and expressivity that could soon have practical benefits to people with speech loss.

The progress here was enabled by several key innovations and findings: advances in the neural interface, providing denser and broader sampling of the distributed orofacial and vocal-tract representations across the lateral SMC; highly stable recordings from non-penetrating cortical-surface electrodes, enabling training and testing across days and weeks without requiring retraining on the day of testing; custom

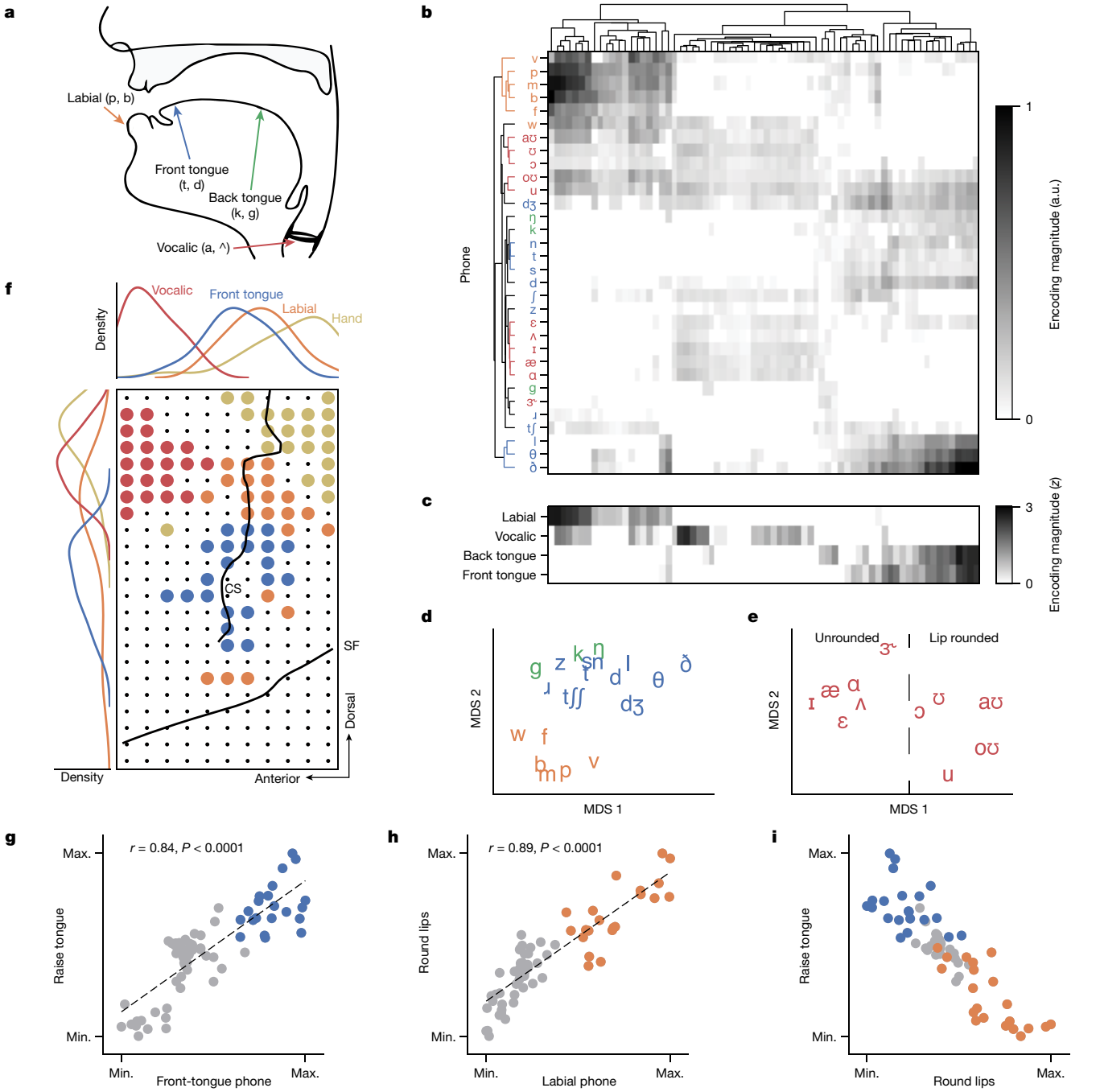


Fig. 5 | Articulatory encodings driving speech decoding. a, Mid-sagittal schematic of the vocal tract with phone POA features labelled. **b**, Phone-encoding vectors for each electrode computed by a temporal receptive-field model on neural activity recorded during attempts to silently say sentences from the 1024-word-General set, organized by unsupervised hierarchical clustering. a.u., arbitrary units. **c**, *z*-scored POA encodings for each electrode, computed by averaging across positive phone encodings within each POA category. *z* values are clipped at 0. **d**,**e**, Projection of consonant (**d**) and vowel (**e**) phone encodings into a 2D space using multidimensional scaling (MDS). **f**, Bottom right: visualization of the locations of electrodes with the greatest encoding weights for labial, front-tongue and vocalic phones on the ECoG array. The electrodes that most strongly encoded finger flexion during the NATO-motor task are also included. Only the top 30% of electrodes within each condition are shown, and the strongest tuning was used for categorization if an electrode was in the top 30% for multiple conditions. Black lines denote the central sulcus (CS) and Sylvian fissure (SF). Top and left: the spatial electrode distributions for each condition along the anterior–posterior and ventral–dorsal axes, respectively. **g**–**i**, Electrode-tuning comparisons between front-tongue phone encoding and tongue-raising attempts (**g**; $r = 0.84$, $P < 0.0001$, ordinary least-squares regression), labial phone encoding and lip-puckering attempts (**h**; $r = 0.89$, $P < 0.0001$, ordinary least-squares regression) and tongue-raising and lip-rounding attempts (**i**). Non-phonetic tunings were computed from neural activations during the articulatory-movement task. Each plot depicts the same electrodes encoding front-tongue and labial phones (from **f**) as blue and orange dots, respectively; all other electrodes are shown as grey dots. Max., maximum; min., minimum.

sequence-learning neural-decoding models, facilitating training without alignment of neural activity and output features; self-supervised learning-derived discrete speech units, serving as effective intermediate representations for intelligible speech synthesis; control of a virtual face from brain activity to accompany synthesized speech and convey facial expressions; and persistent articulatory encoding in the SMC of

our participant that is consistent with previous intact-speech characterizations despite more than 18 years of anarthria, including hand and orofacial-motor somatotopy organized along a dorsal–ventral axis and phonetic tunings clustered by POA.

A limitation of the present proof-of-concept study is that the results shown are from only one participant. An important next step is to validate these decoding approaches in other individuals with varying degrees and etiologies of paralysis (for example, patients who are fully locked-in with ALS)[8,41]. Additionally, providing instantaneous closed-loop feedback during decoding has the potential to improve user engagement, model performance and neural entrainment[42,43]. Also, further advances in electrode interfaces[44] to enable denser and broader cortical coverage should continue to improve accuracy and generalizability towards eventual clinical applications.

The ability to interface with evolving technology to communicate with family and friends, facilitate community involvement and occupational participation, and engage in virtual, Internet-based social contexts (such as social media and metaverses) can vastly expand a person's access to meaningful interpersonal interactions and ultimately improve their quality of life[2,39]. We show here that BCIs can give this ability back to patients through highly personalizable audio-visual synthesis capable of restoring aspects of their personhood and identity. This is further supported by our participant's feedback on the technology, in which she describes how a multimodal BCI would improve her daily life by increasing expressivity, independence and productivity (Supplementary Table 9). A major goal now is to move beyond these initial demonstrations and build seamless integration with real-world applications.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41586-023-06443-4.

1. Moses, D. A. et al. Neuroprosthesis for decoding speech in a paralyzed person with anarthria. *N. Engl. J. Med.* **385**, 217–227 (2021).
2. Peters, B. et al. Brain-computer interface users speak up: The Virtual Users' Forum at the 2013 International Brain-Computer Interface Meeting. *Arch. Phys. Med. Rehabil.* **96**, S33–S37 (2015).
3. Metzger, S. L. et al. Generalizable spelling using a speech neuroprosthesis in an individual with severe limb and vocal paralysis. *Nat. Commun.* **13**, 6510 (2022).
4. Beukelman, D. R. et al. *Augmentative and Alternative Communication* (Paul H. Brookes, 1998).
5. Graves, A., Fernández, S., Gomez, F. & Schmidhuber, J. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proc. 23rd International Conference on Machine learning - ICML '06* (eds Cohen, W. & Moore, A.) 369–376 (ACM Press, 2006); https://doi.org/10.1145/1143844.1143891.
6. Watanabe, S., Delcroix, M., Metze, F. & Hershey, J. R. *New Era for Robust Speech Recognition: Exploiting Deep Learning*. (Springer, 2017).
7. Vansteensel, M. J. et al. Fully implanted brain–computer interface in a locked-in patient with ALS. *N. Engl. J. Med.* **375**, 2060–2066 (2016).
8. Pandarinath, C. et al. High performance communication by people with paralysis using an intracortical brain-computer interface. *eLife* **6**, e18554 (2017).
9. 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).
10. Angrick, M. et al. Speech synthesis from ECoG using densely connected 3D convolutional neural networks. *J. Neural Eng.* **16**, 036019 (2019).
11. Anumanchipalli, G. K., Chartier, J. & Chang, E. F. Speech synthesis from neural decoding of spoken sentences. *Nature* **568**, 493–498 (2019).
12. Hsu, W.-N. et al. HuBERT: self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Trans. Audio Speech Lang. Process.* **29**, 3451–3460 (2021).
13. Cho, C. J., Wu, P., Mohamed, A. & Anumanchipalli, G. K. Evidence of vocal tract articulation in self-supervised learning of speech. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, 2023).
14. Lakhotia, K. et al. On generative spoken language modeling from raw audio. In *Trans. Assoc. Comput. Linguist.* **9**, 1336–1354 (2021).
15. Prenger, R., Valle, R. & Catanzaro, B. Waveglow: a flow-based generative network for speech synthesis. In *Proc. ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (eds Sanei. S. & Hanzo, L.) 3617–3621 (IEEE, 2019); https://doi.org/10.1109/ICASSP.2019.8683143.
16. Yamagishi, J. et al. Thousands of voices for HMM-based speech synthesis–analysis and application of TTS systems built on various ASR corpora. *IEEE Trans. Audio Speech Lang. Process.* **18**, 984–1004 (2010).
17. Wolters, M. K., Isaac, K. B. & Renals, S. Evaluating speech synthesis intelligibility using Amazon Mechanical Turk. In *Proc. 7th ISCA Workshop Speech Synth. SSW-7* (eds Sagisaka, Y. & Tokuda, K.) 136–141 (2010).
18. Mehrabian, A. *Silent Messages: Implicit Communication of Emotions and Attitudes* (Wadsworth, 1981).
19. Jia, J., Wang, X., Wu, Z., Cai, L. & Meng, H. Modeling the correlation between modality semantics and facial expressions. In *Proc. 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference* (eds Lin, W. et al.) 1–10 (2012).
20. Sadikaj, G. & Moskowitz, D. S. I hear but I don't see you: interacting over phone reduces the accuracy of perceiving affiliation in the other. *Comput. Hum. Behav.* **89**, 140–147 (2018).
21. Sumby, W. H. & Pollack, I. Visual contribution to speech intelligibility in noise. *J. Acoust. Soc. Am.* **26**, 212–215 (1954).
22. Chartier, J., Anumanchipalli, G. K., Johnson, K. & Chang, E. F. Encoding of articulatory kinematic trajectories in human speech sensorimotor cortex. *Neuron* **98**, 1042–1054 (2018).
23. Bouchard, K. E., Mesgarani, N., Johnson, K. & Chang, E. F. Functional organization of human sensorimotor cortex for speech articulation. *Nature* **495**, 327–332 (2013).
24. Carey, D., Krishnan, S., Callaghan, M. F., Sereno, M. I. & Dick, F. Functional and quantitative MRI mapping of somatomotor representations of human supralaryngeal vocal tract. *Cereb. Cortex* **27**, 265–278 (2017).
25. Mugler, E. M. et al. Differential representation of articulatory gestures and phonemes in precentral and inferior frontal gyri. *J. Neurosci.* **4653**, 1206–1218 (2018).
26. Berger, M. A., Hofer, G. & Shimodaira, H. Carnival—combining speech technology and computer animation. *IEEE Comput. Graph. Appl.* **31**, 80–89 (2011).
27. van den Oord, A., Vinyals, O. & Kavukcuoglu, K. Neural discrete representation learning. *In Proc. 31st International Conference on Neural Information Processing Systems* 6309–6318 (Curran Associates, 2017).
28. King, D. E. Dlib-ml: a machine learning toolkit. *J. Mach. Learn. Res.* **10**, 1755–1758 (2009).
29. Salari, E., Freudenburg, Z. V., Vansteensel, M. J. & Ramsey, N. F. Classification of facial expressions for intended display of emotions using brain–computer interfaces. *Ann. Neurol.* **88**, 631–636 (2020).
30. Eichert, N., Papp, D., Mars, R. B. & Watkins, K. E. Mapping human laryngeal motor cortex during vocalization. *Cereb. Cortex* **30**, 6254–6269 (2020).
31. Breshears, J. D., Molinaro, A. M. & Chang, E. F. A probabilistic map of the human ventral sensorimotor cortex using electrical stimulation. *J. Neurosurg.* **123**, 340–349 (2015).
32. Simonyan, K., Vedaldi, A. & Zisserman, A. Deep inside convolutional networks: visualising image classification models and saliency maps. In *Proc. Workshop at International Conference on Learning Representations* (eds. Bengio, Y. & LeCun, Y.) (2014).
33. Umeda, T., Isa, T. & Nishimura, Y. The somatosensory cortex receives information about motor output. *Sci. Adv.* **5**, eaaw5388 (2019).
34. Murray, E. A. & Coulter, J. D. Organization of corticospinal neurons in the monkey. *J. Comp. Neurol.* **195**, 339–365 (1981).
35. Arce, F. I., Lee, J.-C., Ross, C. F., Sessle, B. J. & Hatsopoulos, N. G. Directional information from neuronal ensembles in the primate orofacial sensorimotor cortex. *J. Neurophysiol.* **110**, 1357–1369 (2013).
36. Eichert, N., Watkins, K. E., Mars, R. B. & Petrides, M. Morphological and functional variability in central and subcentral motor cortex of the human brain. *Brain Struct. Funct.* **226**, 263–279 (2021).
37. Binder, J. R. Current controversies on Wernicke's area and its role in language. *Curr. Neurol. Neurosci. Rep.* **17**, 58 (2017).
38. Rousseau, M.-C. et al. Quality of life in patients with locked-in syndrome: evolution over a 6-year period. *Orphanet J. Rare Dis.* **10**, 88 (2015).
39. Felgoise, S. H., Zaccheo, V., Duff, J. & Simmons, Z. Verbal communication impacts quality of life in patients with amyotrophic lateral sclerosis. *Amyotroph. Lateral Scler. Front. Degener.* **17**, 179–183 (2016).
40. Huggins, J. E., Wren, P. A. & Gruis, K. L. What would brain-computer interface users want? Opinions and priorities of potential users with amyotrophic lateral sclerosis. *Amyotroph. Lateral Scler.* **12**, 318–324 (2011).
41. Bruurmijn, M. L. C. M., Pereboom, I. P. L., Vansteensel, M. J., Raemaekers, M. A. H. & Ramsey, N. F. Preservation of hand movement representation in the sensorimotor areas of amputees. *Brain* **140**, 3166–3178 (2017).
42. Brumberg, J. S., Pitt, K. M. & Burnison, J. D. A noninvasive brain-computer interface for real-time speech synthesis: the importance of multimodal feedback. *IEEE Trans. Neural Syst. Rehabil. Eng.* **26**, 874–881 (2018).
43. Sadtler, P. T. et al. Neural constraints on learning. *Nature* **512**, 423–426 (2014).
44. Chiang, C.-H. et al. Development of a neural interface for high-definition, long-term recording in rodents and nonhuman primates. *Sci. Transl. Med.* **12**, eaay4682 (2020).

Methods

Clinical-trial overview

This study was completed within the BCI Restoration of Arm and Voice clinical trial (ClinicalTrials.gov; NCT03698149). The primary endpoint of this trial is to assess the long-term safety and tolerability of an ECoG-based interface. All data presented here are part of the ongoing exploratory clinical trial and do not contribute towards any conclusions regarding the primary safety endpoints of the trial. The clinical trial began in November 2018, with all data in this present work collected in 2022 and 2023. Following the Food and Drug Administration's investigational device exemption approval for the neural-implant device used in this study, the study protocol was approved by the University of California, San Francisco Institutional Review Board. The participant gave her informed consent to participate in this trial following multiple conversations with study investigators in which the details of study enrolment, including risks related to the study device, were thoroughly explained to her. The original and current clinical protocols are provided in the Supplementary Information.

Participant

The participant, who was 47 years old at time of enrolment into the study, was diagnosed with quadriplegia and anarthria by neurologists and a speech–language pathologist. She experienced a pontine infarct in 2005, when she was 30 years old and in good health; she experienced sudden-onset dizziness, slurred speech, quadriplegia and bulbar weakness. She was found to have a large pontine infarct with left vertebral artery dissection and basilar artery occlusion. During enrolment evaluation, she scored 29/30 on the Mini Mental State Exam and was unable to achieve the final point only because she could not physically draw a figure due to her paralysis. She can vocalize a small set of monosyllabic sounds, such as 'ah' or 'ooh', but she is unable to articulate intelligible words (Supplementary Note 1). During clinical assessments, a speech–language pathologist prompted her to say 58 words and 10 phrases and also asked her to respond to 2 open-ended questions within a structured conversation. From the resulting audio and video transcriptions of her speech attempts, the speech–language pathologist measured her intelligibility to be 5% for the prompted words, 0% for the prompted sentences and 0% for the open-ended responses. To investigate how similar her movements during silent-speech attempts were relative to neurotypical speakers, we applied a state-of-the-art visual-speech-recognition model[45] to videos of the participant's face during imagined, silently attempted and vocal attempted speech. We found a median WER of 95.8% (99% CI [90.0, 125.0]) for silently attempted speech, which was far higher than the median WER from videos of volunteer healthy speakers, which was 50.0% (99% CI [37.5, 62.5]; Supplementary Fig. 17). Functionally, she cannot use speech to communicate. Instead, she relies on a transparent letter board and a Tobii Dynavox for communication (Supplementary Note 2). She used her transparent letter board to provide informed consent to participate in this study and to allow her image to appear in demonstration videos. To sign the physical consent documents, she used her communication board to spell out "I consent" and directed her spouse to sign the documents on her behalf.

Neural implant

The neural-implant device used in this study featured a high-density ECoG array (PMT) and a percutaneous pedestal connector (Blackrock Microsystems). The ECoG array consists of 253 disc-shaped electrodes arranged in a lattice formation with 3-mm centre-to-centre spacing. Each electrode has a 1-mm recording-contact diameter and a 2-mm overall diameter. The array was surgically implanted subdurally on the pial surface of the left hemisphere of the brain, covering regions associated with speech production and language perception, including the middle aspect of the superior and middle temporal gyri, the precentral gyrus and the postcentral gyrus. The percutaneous pedestal connector, which was secured to the skull during the same operation, conducts electrical signals from the ECoG array to a detachable digital headstage and HDMI cable (CerePlex E256; Blackrock Microsystems). The digital headstage minimally processes and digitizes the acquired cortical signals and then transmits the data to a computer for further signal processing. The device was implanted in September 2022 at UCSF Medical Center with no surgical complications.

Signal processing

We used the same signal-processing pipeline detailed in our previous work[3] to extract HGA[46] and low-frequency signals (LFSs) from the ECoG signals at a 200-Hz sampling rate. Briefly, we first apply common average referencing to the digitized ECoG signals and downsample them to 1 kHz after applying an anti-aliasing filter with a cutoff of 500 Hz. Then we compute HGA as the analytic amplitude of these signals after band-passing them in the high-gamma frequency range (70–150 Hz), and then downsample them to 200 Hz. For LFSs, we apply only a low-pass anti-aliasing filter with a cutoff frequency of 100 Hz, and then downsample signals to 200 Hz. For data normalization, we applied a 30-s sliding-window z score in real time to the HGA and LFS features from each ECoG channel.

We carried out all data collection and real-time decoding tasks in the common area of the participant's residence. We used a custom Python package named rtNSR, which we created in previous work but have continued to augment and maintain over time[1,3,47], to collect and process all data, run the tasks and coordinate the real-time decoding processes. After each session, we uploaded the neural data to our laboratory's server infrastructure, where we analysed the data and trained decoding models.

Task design

Experimental paradigms. To collect training data for our decoding models, we implemented a task paradigm in which the participant attempted to produce prompted targets. In each trial of this paradigm, we presented the participant with text representing a speech target (for example, "Where was he trying to go?") or a non-speech target (for example, "Lips back"). The text was surrounded by three dots on both sides, which sequentially disappeared to act as a countdown. After the final dot disappeared, the text turned green to indicate the go cue, and the participant attempted to silently say that target or carry out the corresponding action. After a brief delay, the screen cleared and the task continued to the next trial.

During real-time testing, we used three different task conditions: text, audio-visual and NATO motor. We used the text task condition to evaluate the text decoder. In this condition, we used the top half of the screen to present prompted targets to the participant, as we did for training. We used the bottom half of the screen to display an indicator (three dots) when the text decoder first predicted a non-silence phone, which we updated to the full decoded text once the sentence was finalized.

We used the audio-visual task condition to evaluate the speech-synthesis and avatar-animation models, including the articulatory-movement and emotional-expression classifiers. In this condition, the participant attended to a screen showing the Unreal Engine environment that contained the avatar. The viewing angle of the environment was focused on the avatar's face. In each trial, speech and non-speech targets appeared on the screen as white text. After a brief delay, the text turned green to indicate the go cue, and the participant attempted to silently say that target or carry out the corresponding action. Once the decoding models processed the neural data associated with the trial, the decoded predictions were used to animate the avatar and, if the current trial presented a speech target, play the synthesized speech audio.

We used the NATO-motor task condition to evaluate the NATO code-word classification model and to collect neural data during attempted hand-motor movements. This task contained 26 speech targets (the code words in the NATO phonetic alphabet) and 4 non-speech hand-motor targets (left-thumb flexion, right-thumb flexion, right-index- and middle-finger flexion, and left-index- and middle-finger flexion). We instructed the participant to attempt to carry out the hand-motor movements to the best of her ability despite her severe paralysis. This task condition resembled the text condition, except that the top three predictions from the classifier (and their corresponding predicted probabilities) were shown in the bottom half of the screen as a simple horizontal bar chart after each trial. We used the prompted-target paradigm to collect the first few blocks of this dataset, and then we switched to the NATO-motor task condition to collect all subsequent data and to carry out real-time evaluation.

Sentence sets. We used three different sentence sets in this work: 50-phrase-AAC, 529-phrase-AAC and 1024-word-General. The first two sets contained sentences that are relevant for general dialogue as well as AAC[4]. The 50-phrase-AAC set contained 50 sentences composed of 119 unique words, and the 529-phrase-AAC set contained 529 sentences composed of 372 unique words and included all of the sentences in the 50-phrase-AAC set. The 1024-word-General set contained sentences sampled from Twitter and film transcriptions for a total of 13,463 sentences and 1,024 unique words (Method 1 in Supplementary Methods).

To create the 1024-word-General sentence set, we first extracted sentences from the nltk Twitter corpus[48] and the Cornell film corpus[49]. We drew 18,284 sentences from these corpora that were composed entirely from the 1,152-word vocabulary from our previous work[3], which contained common English words. We then subjectively pruned out offensive sentences, sentences that grammatically did not make sense, and sentences with overly negative connotation, and kept sentences between 4 and 8 words, which resulted in 13,463 sentences composed of a total of 1,024 unique words. Partway through training, we removed sentences with syntactic pauses or punctuation in the middle (Method 1 in Supplementary Methods). Of these sentences, we were able to collect 9,406 unique sentences (100 sentences were collected twice, for a total of 9,506 trials) with our participant for use during the training of text and avatar models. We used 95% of this data to train the models and 5% as a held-out development set to evaluate performance and choose hyperparameters before real-time testing. As the synthesis model required several days to train to convergence, this model used only 6,449 trials for training data as the remaining trials were collected while the model was training. Of these trials, 100 were used as a held-out development set to evaluate performance and choose hyperparameters before real-time testing.

We randomly selected 249 sentences from the 1024-word-General set to use as the final test sentences for text decoding (Method 1 in Supplementary Methods). We did not collect training data with these sentences as targets. For evaluation of audio-visual synthesis and the avatar, we randomly selected 200 sentences that were not used during training and were not included in the 249 sentences used for text-decoding evaluation (Method 1 in Supplementary Methods). As a result of the previous reordering, the audio-visual synthesis and avatar test sets contained a larger proportion of common words.

For training and testing with the 1024-word-General sentence set, to help the decoding models infer word boundaries from the neural data without forgoing too much speed and naturalness, we instructed the participant to insert small syllable-length pauses (approximately 300–500 ms) between words during her silent-speech attempts. For all other speech targets, we instructed the participant to attempt to silently speak at her natural rate.

Text decoding

Phone decoding. For the text-decoding models, we downsampled the neural signals by a factor of 6 (from 200 Hz to 33.33 Hz) after applying an anti-aliasing low-pass filter at 16.67 Hz using the Scipy python package[50], as in previous work[1,3]. We then normalized the HGA and LFSs separately to have an L2 norm of 1 across all time steps for each channel. We used all available electrodes during decoding.

We trained an RNN to model the probability of each phone at each time step, given these neural features. We trained the RNN using the CTC loss[5] to account for the lack of temporal alignment between neural activity and phone labels. The CTC loss maximizes the probability of any correct sequence of phone outputs that correspond to the phone transcript of a given sentence. To account for differences in the length of individual phones, the CTC loss collapses over consecutive repeats of the same phone. For example, predictions corresponding to /w ɒ z/ (the phonetic transcription of 'was') could be a result of the RNN predicting the following valid time series of phones: /w ɒ z z/, /w w ɒ ɒ z/ z/, /w w ɒ z/ and so forth.

We determined reference sequences using g2p-en (ref. 51), a grapheme-to-phoneme model that enabled us to recover phone pronunciations for each word in the sentence sets. We inserted a silence token in between each word and at the beginning and end of each sentence. For simplicity, we used a single phonetic pronunciation for each word in the vocabulary. We used these sentence-level phone transcriptions for training and to measure performance during evaluation.

The RNN itself contained a convolutional portion followed by a recurrent portion, which is a commonly used architecture in automatic speech recognition[52,53]. The convolutional portion of our RNN was composed of a 1D convolutional layer with 500 kernels, a kernel size of 4 and a stride of 4. The recurrent portion was composed of 4 layers of bidirectional gated recurrent units with 500 hidden units. The hidden states of the final recurrent layer were passed through a linear layer and projected into a 41D space. These values were then passed through a softmax activation function to estimate the probability of each of the 39 phones, the silence token and the CTC blank token (used in the CTC loss to predict two tokens in a row or to account for silence at each time step)[5]. We implemented these models using the PyTorch Python package (version 1.10.0)[54].

We trained the RNN to predict phone sequences using an 8-s window of neural activity. To improve the model's robustness to temporal variability in the participant's speech attempts, we introduced jitter during training by randomly sampling a continuous 8-s window from a 9-s window of neural activity spanning from 1 s before to 8 s after the go cue, as in previous work[1,3]. During inference, the model used a window of neural activity spanning from 500 ms before to 7.5 s after the go cue. To improve communication rates and decoding of variable-length sentences, we terminated trials before a full 8-s window if the decoder determined that the participant had stopped attempted speech by using silence detection. Here we use 'silence' to refer to the absence of an ongoing speech attempt; all of the participant's attempts to speak were technically silent, so the 'silence' described here can be thought of as idling. To implement this early-stopping mechanism, we carried out the following steps: starting 1.9 s after the go cue and then every 800 ms afterwards, we used the RNN to decode the neural features acquired up to that point in the trial; if the RNN predicted the silence token for the most recent 8 time steps (960 ms) with higher than 88.8% average probability (or, in 2 out of the 249 real-time test trials, if the 7.5-s trial duration expired), the current sentence prediction was used as the final model output and the trial ended. We attempted a version of the task in which the current decoded text was presented to the participant every 800 ms; however, the participant generally preferred seeing only the finalized decoded text. See Method 2 in Supplementary Methods for further details about the data-processing, data-augmentation and training procedures used to fit the RNN and Supplementary Table 10 for hyperparameter values.

Beam-search algorithm. We used a CTC beam-search algorithm to transform the predicted phone probabilities into text[55]. To implement this CTC beam search, we used the ctc_decode function in the torchaudio Python package[56]. Briefly, the beam search finds the most likely sentence given the phone probabilities emitted by the RNN. For each silent-speech attempt, the likelihood of a sentence is computed as the emission probabilities of the phones in the sentence combined with the probability of the sentence under a language-model prior. We used a custom-trained 5-gram language model[57] with Kneser-ney smoothing[58]. We used the KenLM software package[59] to train the 5-gram language model on the full 18,284 sentences that were eligible to be in the 1024-word-General set before any pruning. The 5-gram language model is trained to predict the probability of each word in the vocabulary given the preceding words (up to 4). We chose this approach because the linguistic structure and content of conversational tweets and film lines are more relevant for everyday usage than formal written language commonly used in many standard speech-recognition databases[60,61]. The beam search also uses a lexicon to restrict phone sequences to form valid words within a limited vocabulary. Here we used a lexicon defined by passing each word in the vocabulary through a grapheme-to-phoneme conversion module (g2p-en) to define a valid pronunciation for each word. We used a language model weight of 4.5 and a word insertion score of −0.26 (Method 2 in Supplementary Methods).

Decoding speed. To measure decoding speed during real-time testing, we used the formula $\frac{N}{T}$, in which n is the number of words in the decoded output and T is the time (in minutes) that our participant was attempting to speak. We calculated T by computing the elapsed time between the appearance of the go cue and the time of the data sample that immediately preceded the samples that triggered early stopping, giving the resulting formula:

$$\text{rate} = \frac{n}{t_{\text{silence detected}} - t_{\text{go cue}}}.$$

Here, n remains the number of words in the decoded output. $t_{\text{silence detected}}$ is the time of the data sample that immediately preceded the samples that triggered early stopping, and $t_{\text{go cue}}$ is the time when the go cue appeared.

Error-rate calculation. WER is defined as the word edit distance, which is the minimum number of word deletions, insertions and substitutions required to convert the decoded sentence into the target (prompted) sentence, divided by the number of words in the target sentence. PER and CER are defined analogously for phones and characters, respectively. When measuring PERs, we ignored the silence token at the start of each sentence, as this token is always present at the start of both the reference phone sequence and the phone decoder's output.

For BCIs, error-rate distributions are typically assessed across sets of 5 or more sentences rather than single trials, as single-trial error rates can be noisy and are highly dependent on sentence length[1,3,9]. Hence, we sequentially parcelled sentences into pseudo-blocks of 10 sentences and then evaluated error rates and other metrics across these pseudo-blocks. As in previous work[3,9], this entailed taking the sum of the phone, word and character edit distances between each of the predicted and target sentences in a given pseudo-block, and dividing it by the total number of phones, words or characters across all target sentences in the block, respectively. In the single case in which a pseudo-block contained an invalid trial, that trial was ignored.

Offline simulation of large-vocabulary, 50-phrase-AAC and 500-phrase-AAC results. To simulate text-decoding results using the larger vocabularies, we used the same neural activity, RNN decoder, and start and end times that were used during real-time evaluation. We changed only the underlying 5-gram language model to be trained on all sentences 4 to 8 words in length in the Twitter and Cornell film corpora that fell within the desired vocabulary. We evaluated performance using log-spaced vocabulary sizes consisting of 1,506, 2,270, 3,419, 5,152, 7,763, 11,696, 17,621, 11,696, 26,549 and 39,378 words, and also included the real-time results (1,024 words). To choose the words at each vocabulary size, with the exception of the already defined vocabulary for the real-time results, we first included all words in the 1024-word-General set. Then we used a readily available pronunciation dictionary from the Librispeech Corpus[60] to select all words that were present in both the Twitter and Cornell films corpora and the pronunciation dictionary. The most frequent words that were not in the 1024-word-General set but fell within the pronunciation dictionary were added to reach the target vocabulary size. We then simulated the results on the task with the larger vocabulary and language model.

To simulate text-decoding results on the 50-phrase-AAC and 500-phrase-AAC sentence sets (because we tested the text decoder in real time only with the 1024-word-General set), we trained RNN decoders on data associated with these two AAC sets (Method 2 in Supplementary Methods; see Table S10 for hyperparameter values). We then simulated decoding using the neural data and go cues from the real-time blocks used for evaluation of the avatar and synthesis methods. We checked for early stopping 2.2 s after the start of the sentence and again every subsequent 350 ms. Once an early stop was detected, or if 5.5 s had elapsed since the go cue, we finalized the sentence prediction. During decoding, we applied the CTC beam search using a 5-gram language model fitted on the phrases from that set.

Decoding NATO code words and hand-motor movements. We used the same neural-network decoder architecture (but with a modified input and output layer dimensionality to account for differences in the number of electrodes and target classes) as in previous work[3] to output the probability of each of the 26 NATO code words and the 4 hand-motor targets. To maximize data efficiency, we used transfer learning between our participants; we initialized the decoder using weights from our previous work, and we replaced the first and last layers to account for differences in the number of electrodes and number of classes being predicted, respectively. See Method 3 in Supplementary Methods for further details about the data-processing, data-augmentation and training procedures used to fit the classifier and Supplementary Table 11 for hyperparameter values. For the results shown in Fig. 2h, we computed NATO code-word classification accuracy using a model that was also capable of predicting the motor targets; here we measured performance only on trials in which the target was a NATO code word, and we deemed incorrect any such trial in which a code-word attempt was misclassified as a hand-motor attempt.

Speech synthesis

Training and inference procedure. We used CTC loss to train an RNN to predict a temporal sequence of discrete speech units extracted using HuBERT[12] from neural data. HuBERT is a speech-representation learning model that is trained to predict acoustic k-means-cluster identities corresponding to masked time points from unlabelled input waveforms. We refer to these cluster identities as discrete speech units, and the temporal sequence of these speech units represents the content of the original waveform.

As our participant cannot speak, we generated reference sequences of speech units by applying HuBERT to a speech waveform that we refer to as the basis waveform. For the 50-phrase-AAC and 529-phrase-AAC sets, we acquired basis waveforms from a single male speaker (recruited before our participant's enrolment in the trial) who was instructed to read each sentence aloud in a consistent manner. Owing to the large number of sentences in the 1024-word-General set, we used the Wavenet text-to-speech model[62] to generate basis waveforms.

We used HuBERT to process our basis waveforms and generate a series of reference discrete speech units sampled at 50 Hz. We used the base 100-unit, 12-transformer-layer HuBERT trained on 960 h of LibriSpeech[60], which is available in the open-source fairseq library[63]. In addition to the reference discrete speech units, we added the blank token needed for CTC decoding as a target during training.

The synthesis RNN, which we trained to predict discrete speech units from the ECoG features (HGA and LFSs), consisted of the following layers (in order): a 1D convolutional layer, with 260 kernels with width and stride of 6; three layers of bidirectional gated recurrent units, each with a hidden dimension size of 260; and a 1D transpose convolutional layer, with a size and stride of 6, that output discrete-unit logits. To improve robustness, we applied data augmentations using the SpecAugment method[64] to the ECoG features during training. See Method 4 in the Supplementary Methods for the complete training procedure and Supplementary Table 12 for hyperparameter values.

From the ECoG features, the RNN predicted the probability of each discrete unit every 5 ms. We retained only the most likely predicted unit at each time step. We ignored time steps in which the CTC blank token was decoded, as this is primarily used to adjust for alignment and repeated decodes of discrete units. Next we synthesized a speech waveform from the sequence of discrete speech units, using a pretrained unit-to-speech vocoder[65].

During each real-time inference trial in the audio-visual task condition, we provided the speech-synthesis model with ECoG features collected in a time window around the go cue. This time window spanned from 0.5 s before to 4.62 s after the go cue for the 50-phrase-AAC and 529-phrase-AAC sentence sets and from 0 s before to 7.5 s after the go cue for the 1024-word-General sentence set. The model then predicted the most likely sequence of HuBERT units from the neural activity and generated the waveform using the aforementioned vocoder. We streamed the waveform in 5-ms chunks of audio directly to the real-time computer's sound card using the PyAudio Python package.

To decode speech waveforms in the participant's personalized voice (that is, a voice designed to resemble the participant's own voice before her injury), we used YourTTS[66], a zero-shot voice-conversion model. After conditioning the model on a short clip of our participant's voice extracted from a pre-injury video of her, we applied the model to the decoded waveforms to generate the personalized waveforms (Extended Data Fig. 5 and Supplementary Table 1). To reduce the latency of the personalized speech synthesizer during real-time inference for a qualitative demonstration (Supplementary Video 1), we trained a HiFi-CAR convolutional neural network[67] to vocode HuBERT units into personalized speech. This model used voice-converted LJSpeech (by means of YourTTS) as training data.

Evaluation. To evaluate the quality of the decoded speech, we computed the MCD between the decoded and reference waveforms ($\hat{y}$ and y, respectively)[68]. This is defined as the squared error between dynamically time-warped sequences of mel cepstra (mc_d, in which d is the index of the mel cepstra) extracted from the target and decoded waveforms and is commonly used to evaluate the quality of synthesized speech:

$$\mathrm{MCD}(\hat{y}, y) = \frac{10}{\log(10)} \sqrt{\left(\sum_{d=1}^{24} (\mathrm{mc}_d^{y} + \mathrm{mc}_d^{\hat{y}})^2 \right)}$$

We excluded silence time points at the start and end of each waveform during MCD calculation. For each pseudo-block, we combined the MCD of 10 individual trials by taking their mean.

We designed a perceptual assessment using a crowd-sourcing platform (Amazon Mechanical Turk), where each test trial was assessed by 12 evaluators (except for 3 of the 500 trials, in which only 11 workers completed their evaluations). In each evaluation, the evaluator listened to the decoded speech waveform and then transcribed what they heard (Method 4 in the Supplementary Methods). For each sentence, we then computed the WER and CER between the evaluator's transcriptions and the ground-truth transcriptions. To control for outlier evaluator performance, for each trial, we used the median WER and CER across evaluators as the final accuracy metric for the decoded waveform. We reported metrics across pseudo-blocks of ten sentences to be consistent with text-decoding evaluations and calculated WER across each pseudo-block in the same manner as for text decoding

Avatar

Articulatory-gesture data. We used a dataset of articulatory gestures for all sentences from the 50-phrase-AAC, 529-phrase-AAC and 1024-word-general datasets provided by Speech Graphics. We generated these articulatory gestures from reference waveforms using Speech Graphics' speech-to-gesture model, which was designed to animate avatar movements given a speech waveform. For each trial, articulatory gestures consisted of 16 individual gesture time series corresponding to jaw, lip and tongue movements (Supplementary Table 13).

Offline training and inference procedure for the direct-avatar-animation approach. To carry out direct decoding of articulatory gestures from neural activity (the direct approach for avatar animation), we first trained a VQ-VAE to encode continuous Speech Graphics' gestures into discrete articulatory-gesture units[27]. A VQ-VAE is composed of an encoder network that maps a continuous feature space to a learned discrete codebook and a decoder network that reconstructs the input using the encoded sequence of discrete units. The encoder was composed of 3 layers of 1D convolutional units with 40 filters, a kernel size of 4 and a stride of 2. Rectified linear unit (ReLU) activations followed the second and third of these layers. After this step, we applied a 1D convolution, with 1 filter and a kernel size and stride of 1, to generate the predicted codebook embedding. We then used nearest-neighbour lookup to predict the discrete articulatory-gesture units. We used a codebook with 40 different 1D vectors, in which the index of the codebook entry with the smallest distance to the encoder's output served as the discretized unit for that entry. We trained the VQ-VAE's decoder to convert discrete sequences of units back to continuous articulatory gestures by associating each unit with the value of the corresponding continuous 1D codebook vector. Next we applied a 1D convolution layer, with 40 filters and a kernel size and stride of 1, to increase the dimensionality. Then, we applied 3 layers of 1D transpose convolutions, with 40 filters, a kernel size of 4 and a stride of 2, to upsample the reconstructed articulatory gestures back to their original length and sampling rate. ReLU activations followed the first and second of these layers. The final 1D transpose convolution had the same number of kernels as the input signal (16). We used the output of the final layer as the reconstructed input signal during training.

To encourage the VQ-VAE units to decode the most critical gestures (such as jaw opening) rather than focusing on those that are less important (such as nostril flare), we weighted the mean-squared error loss for the most important gestures more highly. We upweighted the jaw opening's mean-squared error loss by a factor of 20, and the gestures associated with important tongue movements (tongue-body raise, tongue advance, tongue retraction and tongue-tip raise) and lip movements (rounding and retraction) by a factor of 5. We trained the VQ-VAE using all of the reference articulatory gestures from the 50-phrase-AAC, 529-phrase-AAC and 1024-word-General sentence sets. We excluded from VQ-VAE training any sentence that was used during the evaluations with the 1024-word-General set.

To create the CTC decoder, we trained a bidirectional RNN to predict reference discretized articulatory-gesture units given neural activity. We first downsampled the ECoG features by a factor of 6 to 33.33 Hz. We then normalized these features to have an L2 norm of 1 at each time point across all channels. We used a time window of neural activity spanning from 0.5 s before to 7.5 s after the go cue for the 1024-word-General set and from 0.5 s before to 5.5 s after for the

50-phrase-AAC and 529-phrase-AAC sets. The RNN then processed these neural features using the following components: a 1D convolution layer, with 256 filters with kernel size and stride of 2; three layers of gated recurrent units, each with a hidden dimension size of 512; and a dense layer, which produced a 41D output. We then used the softmax activation function to output the probability of the 40 possible discrete units (determined by the VQ-VAE) as well as the CTC blank token. See Method 5 in Supplementary Methods for full training details for the VQ-VAE and CTC decoder. The model hyperparameters stated here are for the 1024-word-General sentence set (see Supplementary Table 14 for other hyperparameter values).

During inference, the RNN yielded a predicted probability of each discretized articulatory-gesture unit every 60 ms. To transform these output probabilities into a sequence of discretized units, we retained only the most probable unit at each time step. We used the decoder module of the frozen VQ-VAE to transform collapsed sequences of predicted discrete articulatory units (here, 'collapsed' means that consecutive repeats of the same unit were removed) into continuous articulatory gestures.

Real-time acoustic avatar-animation approach. During real-time testing, we animated the avatar using avatar-rendering software (referred to as SG Com; provided by Speech Graphics; Supplementary Fig. 18). This software converts a stream of speech audio into synchronized facial animation with a latency of 50 ms. It carries out this conversion in two steps: first, it uses a custom speech-to-gesture model to map speech audio to a time series of articulatory-gesture activations; then, it carries out a forward mapping from articulatory-gesture activations to animation parameters on a 3D MetaHuman character created by Epic Games. The output animation was rendered using Unreal Engine 4.26 (Method 5 in Supplementary Methods; ref. 69).

For every 10 ms of input audio, the speech-to-gesture model produces a vector of articulatory-gesture activation values, each between 0 and 1 (for which 0 is fully relaxed and 1 is fully contracted). The forward mapping converts these activations into deformations, simulating the effects of the articulatory gestures on the avatar face. As each articulatory gesture approximates the superficial effect of some atomic action, such as opening the jaw or pursing the lips, the gestures are analogous to the Action Units of the Facial Action Coding System[70], a well-known method for taxonomizing human facial movements. However, these articulatory gestures from Speech Graphics are more oriented towards speech articulation and also include tongue movements, containing 16 speech-related articulatory gestures (10 for lips, 4 for tongue, 1 for jaw and 1 for nostril). The system does not generate values for aspects of the vocal tract that are not externally visible, such as the velum, pharynx or larynx.

To provide avatar feedback to the participant during real-time testing in the audio-visual task condition, we streamed 10-ms chunks of decoded audio over an Ethernet cable to a separate machine running the avatar processes to animate the avatar in synchrony with audio synthesis. We imposed a 200-ms delay on the audio output in real time to improve perceived synchronization with the avatar.

The avatar-rendering system also generates non-verbal motion, such as emotional expressions, head motion, eye blinks and eye darts. These are synthesized using a superset of the articulatory gestures involving the entire face and head. These non-verbal motions are used during the audio-visual task condition and emotional-expression real-time decoding.

Speech-related animation evaluation. To evaluate the perceptual accuracy of the decoded avatar animations, we used a crowd-sourcing platform (Amazon Mechanical Turk) to design and conduct a perceptual assessment of the animations. Each decoded animation was assessed by six unique evaluators. Each evaluation consisted of playback of the decoded animation (with no audio) and textual presentation of the target (ground-truth) sentence and a randomly chosen other sentence from the same sentence set. Evaluators were instructed to identify the phrase that they thought the avatar was trying to say (Method 5 in Supplementary Methods). We computed the median accuracy of the evaluations across evaluators for each sentence and treated that as the accuracy for a given trial and then computed the final accuracy distribution using the pseudo-block strategy described above.

Separately, we used the dlib software package[28] to extract 72 facial keypoints for each frame in avatar-rendered and healthy-speaker videos (sampled at 30 frames per second). To obtain videos of healthy speakers, we recorded video and audio of eight volunteers as they produced the same sentences used during real-time testing in the audio-visual task condition. We normalized the keypoint positions relative to other keypoints to account for head movements and rotation: we computed jaw movement as the distance between the keypoint at the bottom of the jaw and the nose, lip aperture as the distance between the keypoints at the top and bottom of the lips, and mouth width as the distance between the keypoints at either corner of the mouth (Method 5 in Supplementary Methods and Supplementary Fig. 19). To compare avatar keypoint movements to those for healthy speakers, and to compare among healthy speakers, we first applied dynamic time warping to the movement time series and then computed the Pearson's correlation between the pair of warped time series. We held out 10 of 200 1024-word-General avatar videos from final evaluation as they were used to select parameters to automatically trim the dlib traces to speech onset and offset. We did this because our automatic segmentation method relied on the acoustic onset and offset, which is absent from direct-avatar-decoding videos.

Articulatory-movement decoding. To collect training data for non-verbal orofacial-movement decoding, we used the articulatory-movement task. Before data collection, the participant viewed a video of an avatar carrying out the following six movements: open mouth, pucker lips, lips back (smiling or lip retraction), raise tongue, lower tongue and close mouth (rest or idle). Then, the participant carried out the prompted-target task containing these movements as targets (presented as text). We instructed the participant to smoothly transition from neutral to the peak of the movement and then back to neutral, all within approximately 2 s starting at the go cue.

To train and test the avatar-movement classifier (Method 5 in Supplementary Methods), we used a window of neural activity spanning from 1 s before to 3 s after the go cue for each trial. We first downsampled the ECoG features (HGA and LFSs) by a factor of 6 to 33.33 Hz. We then normalized these features to have an L2 norm of 1 at each time point across all channels separately for LFS and HGA features. Next, we extracted the mean, minimum, maximum and standard deviation across the first and second halves of the neural time window for each feature. These features were then stacked to form a 4,048D neural-feature vector (the product of 256 electrodes, 2 feature sets, 4 statistics and 2 data halves) for each trial. We then trained a multilayer perceptron consisting of 2 linear layers with 512 hidden units and ReLU activations between the first and second layers. The final layer projected the output into a 6D output vector. We then applied a softmax activation to get a probability for each of the six different gestures. We evaluated the network using tenfold cross-validation.

Emotional-expression decoding. To collect training data for non-verbal emotional-expression decoding, we used the emotional-expression task. Using the prompted-target task paradigm, we collected neural data as the participant attempted to produce three emotions (sad, happy and surprised) at three intensity levels (high, medium and low) for a total of nine unique expressions. The participant chose her three base emotional expressions from a list of 30 options per emotion, and the animations corresponding to the three intensity levels were generated from these chosen base expressions.

We instructed the participant to smoothly transition from neutral to the peak of the expression and then back to neutral, all within approximately 2 s starting at the go cue. We used the same data-windowing and neural-processing steps as for the articulatory-movement decoding. We used the same model architecture and training procedure as for the NATO-and-hand-motor classifier and our previous work[3]. We initialized the expression classifier with a pretrained NATO-and-hand-motor classifier (trained on 1,222 trials of NATO-motor task data collected before the start of collection for the emotional-expression task) and fine-tuned the weights on neural data from the emotional-expression task. See Method 5 in Supplementary Methods for further details on the data augmentation, ensembling and hyperparameter values used with this model.

We evaluated the expression classifier using 15-fold cross-validation. Within the training set of each cross-validation fold, we fitted ten unique models to ensemble predictions on the held-out test set. We applied hierarchical agglomerative clustering to the nine-way confusion matrix in Fig. 4e using SciPy[50].

Articulatory-encoding assessments

To investigate the neural representations driving speech decoding, we assessed the selectivity of each electrode to articulatory groups of phones. Specifically, we fitted a linear receptive-field encoding model to predict each electrode's HGA from phone-emission probabilities predicted by the text-decoding model during tenfold cross-validation with data recorded with the 1024-word-General sentence set. We first decimated the HGA by a factor of 24, from 200 Hz to 8.33 Hz, to match the sampling rate of the phone-emission probabilities. Then, we fitted a linear receptive-field model to predict the HGA at each electrode, using the phone-emission probabilities as time-lagged input features (39 phones and 1 aggregate token representing both the silence and CTC blank tokens). We used a ±4-sample (480-ms) receptive-field window, allowing for slight misalignment between the text decoder's bidirectional-RNN phone-emission probabilities and the underlying HGA. We fitted an independent model for each electrode. The true HGA, HGA(t), is modelled as a weighted linear combination of phone-emission probabilities (indexed by p) in the overall emissions matrix (X) over a ±4-sample window around each time point. This resulted in a learned weight matrix $w(d,p)$ in which each phone, p, has temporal coefficients $d_1 \ldots D$, in which d_1 is −4 and D is 4. During training, the squared error between the predicted HGA, HGA*(t), and the true HGA, HGA(t), is minimized, using the following formulae:

$$\mathrm{HGA}^*(t) = \sum_{d=1}^{D} \sum_{p=1}^{P} w(d,p) \times X(p, t-d)$$

$$\min \sum_{t} [\mathrm{HGA}^*(t) - \mathrm{HGA}(t)]^2$$

We implemented the model with the MNE toolbox's receptive-field ridge regression in Python[71]. We used tenfold cross-validation to select the optimal alpha ridge-regression parameter by sweeping over the values $[1\times10^{-1}, 1\times10^{0}, 1\times10^{1}, \ldots 1\times10^{5}]$, using 10% of our total data as a held-out tuning set. We then conducted another round of tenfold cross-validation on the remaining 90% of our total data to evaluate performance with the optimized alpha parameter. We averaged the coefficients for the model across the ten folds and collapsed across time samples for every phone using the maximum magnitude weight. The sign of the weight could be positive or negative. This yielded a single vector for each electrode, where each element in each vector was the maximum encoding of a given phone. Next, we pruned any electrode channels that were not significantly modulated by silent-speech attempts. For each electrode, we computed the mean HGA magnitudes in the 1-s intervals immediately before and after the go cue for each NATO code-word trial in the NATO-motor task. If an electrode did not have significantly increased HGA after the go cue compared to before, it was excluded from the remainder of this analysis (significant modulation determined using one-sided Wilcoxon signed-rank tests with an alpha level of 0.00001 after applying 253-way Holm–Bonferroni correction). We then applied a second pruning step to exclude any electrodes that had encoding values (r) less than or equal to 0.2 (Supplementary Fig. 20). We applied the centroid clustering method, a hierarchical, agglomerative clustering technique, to the encoding vectors using the SciPy Python package[72]. We carried out clustering along both the electrode and phone dimensions.

To assess any relationships between phone encodings and articulatory features, we assigned each phone to a POA feature category, similar to what was done in previous work[22,23]. Specifically, each phone was primarily articulated at the lips (labial), the front tongue, the back tongue or the larynx (vocalic). To quantify whether the unsupervised phone-encoding clusters reflected grouping by POA, we tested the null hypothesis that the observed parcellation of phones into clusters was not more organized by POA category than by chance. To test this null hypothesis, we used the following steps: (1) compute the POA linkage distances by clustering the phones by Euclidean distance into F clusters, with $F = 4$ being the number of POA categories; (2) randomly shuffle the mapping between the phone labels and the phonetic encodings; (3) for each POA category, compute the maximum number of phones within that category that appear within a single cluster; (4) repeat steps 2 and 3 over a total of 10,000 bootstrap runs; (5) compute the pairwise Euclidean distance between all combinations of the 10,000 bootstrap results; (6) repeat step 3 using the true unsupervised phone ordering and clustering; (7) compute the pairwise Euclidean distances between the result from step 6 and each bootstrap from step 4; (8) compute the one-tailed Wilcoxon rank-sum test between the results from step 7 and step 5. The resulting P value is the probability of the aforementioned null hypothesis.

To visualize population-level (across all electrodes that were not pruned from the analysis) encoding of POA features, we first computed the mean encoding of each electrode across the four POA feature groups (vocalic, front tongue, labial and back tongue). We then z scored the mean encodings for each POA feature and then applied multidimensional scaling over the electrodes to visualize each phone in a 2D space. We implemented this using the scikit-learn Python package[73].

To measure somatotopy, we computed kernel density estimations of the locations of top electrodes (the 30% of electrodes with the strongest encoding weights) for each POA category along anterior–posterior and dorsal–ventral axes (Fig. 2f). To do this, we used the seaborn Python package[74], Gaussian kernels and Scott's rule.

To quantify the magnitude of activation in response to non-verbal orofacial movements, we took the median of the evoked response potential to each action over the time window spanning from 1 s before to 2 s after the go cue. From this, we subtracted the same metric computed across all actions to account for electrodes that were non-differentially task activated. For each action, we then normalized values across electrodes to be between 0 and 1. We used ordinary least-squares linear regression, implemented by the statsmodels Python package[75], to relate phone-encoding weights with activation to attempted motor movements.

To assess whether postcentral responses largely reflected sensory feedback, we compared the time to activation between precentral and postcentral electrodes. For each speech-responsive electrode (see above), we averaged the HGA across trials (event-related potentials (ERPs)) of each of the 26 NATO code words. For each electrode, we found the time at which each code-word ERP reached its peak. Given that electrodes may have strong preferences for groups of phones (Fig. 5), we took the minimum time-to-peak across code-word ERPs for further analysis. For each electrode's optimal code-word ERP, we also calculated the time-to-onset, defined as the earliest time point at which

the HGA was statistically significantly greater than 0. We measured this with Wilcoxon rank-sum tests at a significance level of 0.05, similar to what was done in previous work[76].

Exclusion analyses

We assigned each electrode to an anatomical region and visualized all electrodes on the pial surface using the same methods described in our previous work[77]. For the exclusion analyses, we tested the phone-based text-decoding model on the real-time evaluation trials in the text task condition with the 1024-word-General sentence set. We did not use early stopping for these analyses; we used the full 8-s time windows of neural activity for each trial. For the synthesis and direct-avatar decoding models, we tested on the real-time synthesis evaluation trials from the 1024-word-General set, and evaluation remained consistent with other analyses (Methods 4 and 5 in Supplementary Methods). Also, we tested the NATO code-word classifier by training and testing on NATO code-word trials recorded during the NATO-motor task (Supplementary Fig. 10). We used all of the NATO-motor task blocks recorded after freezing the classifier (Fig. 2h), a total of 19 blocks, as the test set.

Electrode contributions

For text, synthesis and direct-avatar decoding models, we measured the contribution of each electrode to the model's predictions. We computed the derivative of each model's loss function with respect to the HGA and LFS features of each electrode across time[32]. We then computed the L1 norm of these values across time and averaged across all trials in the corresponding test set for the model. For each electrode, we then summed the resulting contribution for HGA and LFSs to obtain one aggregate contribution. For each model, contributions were then normalized to fall between 0 and 1. To compare contributions across decoding modalities, we used ordinary least-squares linear regression, implemented by the statsmodels Python package[75].

Statistical analyses

Statistical tests are fully described in the figure captions and text. To summarize, we used two-sided Mann–Whitney Wilcoxon rank-sum tests to compare unpaired distributions. Critically, these tests do not assume normally distributed data. For paired comparisons, we used two-sided Wilcoxon signed-rank tests, which also do not assume normally distributed data. When the underlying neural data were not independent across comparisons, we used the Holm–Bonferroni correction for multiple comparisons. P values < 0.01 were considered statistically significant. 99% confidence intervals were estimated using a bootstrapping approach in which we randomly sampled the distribution (for example, trials or pseudo-blocks) of interest with replacement 2,000 or 1,000 times and the desired metric was computed. The confidence interval was then computed on this distribution of the bootstrapped metric. P values associated with the Pearson correlation were computed with a permutation test in which data were randomly shuffled 1,000 times. To compare success rates of decoding during our freeform demonstration with the main real-time evaluation, we used a t-test.

Reporting summary

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

Data availability

Data relevant to this study are accessible under restricted access according to our clinical trial protocol, which enables us to share de-identified information with researchers from other institutions but prohibits us from making it publicly available. Access can be granted upon reasonable request. Requests for access to the dataset can be made online at https://doi.org/10.5281/zenodo.8200782. Response can be expected within three weeks. Any data provided must be kept confidential and cannot be shared with others unless approval is obtained. To protect the participant's anonymity, any information that could identify her will not be part of the shared data. Source data to recreate the figures in the manuscript, including error rates, statistical values and cross-validation accuracy will be publicly released upon publication of the manuscript. Source data are provided with this paper.

Code availability

Code to replicate the main findings of this study can be found on GitHub at https://github.com/UCSF-Chang-Lab-BRAVO/multimodal-decoding.

45. Shi, B., Hsu, W.-N., Lakhotia, K. & Mohamed, A. Learning audio-visual speech representation by masked multimodal cluster prediction. In *Proc. International Conference on Learning Representations* (2022).
46. Crone, N. E., Miglioretti, D. L., Gordon, B. & Lesser, R. P. Functional mapping of human sensorimotor cortex with electrocorticographic spectral analysis. II. Event-related synchronization in the gamma band. *Brain* **121**, 2301–2315 (1998).
47. Moses, D. A., Leonard, M. K. & Chang, E. F. Real-time classification of auditory sentences using evoked cortical activity in humans. *J. Neural Eng.* **15**, 036005 (2018).
48. Bird, S. & Loper, E. NLTK: The Natural Language Toolkit. In *Proc. ACL Interactive Poster and Demonstration Sessions* (ed. Scott, D.) 214–217 (Association for Computational Linguistics, 2004).
49. Danescu-Niculescu-Mizil, C. & Lee, L. Chameleons in imagined conversations: a new approach to understanding coordination of linguistic style in dialogs. *In Proc. 2nd Workshop on Cognitive Modeling and Computational Linguistics* (eds. Hovy, D. et al.) 76–87 (Association for Computational Linguistics, 2011).
50. Virtanen, P. et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**, 261–272 (2020).
51. Park, K. & Kim, J. g2pE. (2019); https://github.com/Kyubyong/g2p.
52. Graves, A., Mohamed, A. & Hinton, G. Speech recognition with deep recurrent neural networks. In *Proc. International Conference on Acoustics, Speech, and Signal Processing* (eds Ward, R. & Deng, L.) 6645–6649 (2013); https://doi.org/10.1109/ICASSP.2013.6638947.
53. Hannun, A. et al. Deep Speech: scaling up end-to-end speech recognition. Preprint at https://arXiv.org/abs/1412.5567 (2014).
54. Paszke, A. et al. Pytorch: an imperative style, high-performance deep learning library. In *Proc. Advances in Neural Information Processing Systems 32* (2019).
55. Collobert, R., Puhrsch, C. & Synnaeve, G. Wav2Letter: an end-to-end ConvNet-based speech recognition system. Preprint at https://doi.org/10.48550/arXiv.1609.03193 (2016).
56. Yang, Y.-Y. et al. Torchaudio: building blocks for audio and speech processing. In *Proc. ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (ed. Li, H.) 6982–6986 (2022); https://doi.org/10.1109/ICASSP43922.2022.9747236.
57. Jurafsky, D. & Martin, J. H. *Speech and Language Processing: an Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (Pearson Education, 2009).
58. Kneser, R. & Ney, H. Improved backing-off for M-gram language modeling. In *Proc. 1995 International Conference on Acoustics, Speech, and Signal Processing* Vol. 1 (eds Sanei. S. & Hanzo, L.) 181–184 (IEEE, 1995).
59. Heafield, K. KenLM: Faster and smaller language model queries. In *Proc. Sixth Workshop on Statistical Machine Translation*, 187–197 (Association for Computational Linguistics, 2011).
60. Panayotov, V., Chen, G., Povey, D. & Khudanpur, S. Librispeech: an ASR corpus based on public domain audio books. In *Proc. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 5206–5210 (2015); https://doi.org/10.1109/ICASSP.2015.7178964.
61. Ito, K. & Johnson, L. The LJ speech dataset (2017); https://keithito.com/LJ-Speech-Dataset/.
62. van den Oord, A. et al. WaveNet: a generative model for raw audio. Preprint at https://arXiv.org/abs/1609.03499 (2016).
63. Ott, M. et al. fairseq: a fast, extensible toolkit for sequence modeling. In *Proc. 2019 Conference of the North American Chapter of the Association for Computational Linguistics* (Demonstrations) (eds. Muresan, S., Nakov, P. & Villavicencio, A.) 48–53 (Association for Computational Linguistics, 2019).
64. Park, D. S. et al. SpecAugment: a simple data augmentation method for automatic speech recognition. In *Proc. Interspeech 2019* (eds Kubin, G. & Kačič, Z.) 2613–2617 (2019); https://doi.org/10.21437/Interspeech.2019-2680.
65. Lee, A. et al. Direct speech-to-speech translation with discrete units. In *Proc. 60th Annual Meeting of the Association for Computational Linguistics* Vol. 1, 3327–3339 (Association for Computational Linguistics, 2022).
66. Casanova, E. et al. YourTTS: towards zero-shot multi-speaker TTS and zero-shot voice conversion for everyone. In *Proc. of the 39th International Conference on Machine Learning* Vol. 162 (eds. Chaudhuri, K. et al.) 2709–2720 (PMLR, 2022).
67. Wu, P., Watanabe, S., Goldstein, L., Black, A. W. & Anumanchipalli, G. K. Deep speech synthesis from articulatory representations. In *Proc. Interspeech 2022* 779–783 (2022).
68. Kubichek, R. Mel-cepstral distance measure for objective speech quality assessment. In *Proc. IEEE Pacific Rim Conference on Communications Computers and Signal Processing* Vol. 1, 125–128 (IEEE, 1993).
69. The most powerful real-time 3D creation tool — Unreal Engine (Epic Games, 2020).
70. Ekman, P. & Friesen, W. V. Facial action coding system. APA PsycNet https://doi.org/10.1037/t27734-000 (2019).

71. Gramfort, A. et al. MEG and EEG data analysis with MNE-Python. *Front. Neurosci.* https://doi.org/10.3389/fnins.2013.00267 (2013).
72. Müllner, D. Modern hierarchical, agglomerative clustering algorithms. Preprint at https://arXiv.org/abs/1109.2378 (2011).
73. Pedregosa, F. et al. Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
74. Waskom, M. seaborn: statistical data visualization. *J. Open Source Softw.* **6**, 3021 (2021).
75. Seabold, S. & Perktold, J. Statsmodels: econometric and statistical modeling with Python. In *Proc. 9th Python in Science Conference* (eds. van der Walt, S. & Millman, J.) 92–96 (2010); https://doi.org/10.25080/Majora-92bf1922-011.
76. Cheung, C., Hamilton, L. S., Johnson, K. & Chang, E. F. The auditory representation of speech sounds in human motor cortex. *eLife* **5**, e12577 (2016).
77. Hamilton, L. S., Chang, D. L., Lee, M. B. & Chang, E. F. Semi-automated anatomical labeling and inter-subject warping of high-density intracranial recording electrodes in electrocorticography. *Front. Neuroinform.* **11**, 62 (2017).

Acknowledgements We thank our participant, Bravo-3, for her incredible dedication and commitment. We thank T. Dubnicoff for video editing, K. Probst for illustrations, members of the laboratory of E.F.C. for feedback, V. Her for administrative support, and the participant's family and caregivers for logistic support. We thank C. Kurtz-Miott, V. Anderson and S. Brosler for help with data collection with our participant. We thank T. Li for assistance with generating the participant's personalized voice. We thank P. Liu for conducting a comprehensive speech–language pathology assessment, B. Speidel for help with imaging reconstruction of the patient's pial surface and electrode array, the volunteers for the healthy-speaker video recordings, and I. Garner for manuscript review. We also thank the Speech Graphics team, specifically D. Palaz and G. Clarke, for adapting and supporting the technology used in this work and for providing articulatory data. Last, we thank L. Sugrue for preoperative functional magnetic resonance imaging planning. For this work, the National Institutes of Health (grant NINDS 5U01DC018671), Joan and Sandy Weill Foundation, Susan and Bill Oberndorf, Ron Conway, David Krane, Graham and Christina Spencer, and William K. Bowes, Jr. Foundation supported S.L.M., K.T.L., D.A.M., M.P.S., R.W., M.E.D., J.R.L., G.K.A. and E.F.C. K.T.L., P.W. and G.K.A. are also supported by the Rose Hills Foundation and the Noyce Foundation. A.B.S. is supported by the National Institute of General Medical Sciences Medical Scientist Training Program, grant no. T32GM007618. K.T.L. is supported by the National Science Foundation GRFP. A.T.-C. and K.G. did not have relevant funding for this work.

Author contributions S.L.M. designed and, along with A.B.S., trained and optimized the text decoder, NATO-and-hand-motor classifier and language model. K.T.L. and R.W. designed, trained and optimized the speech-synthesis models with input from G.K.A. K.T.L., M.A.B., D.A.M. and M.E.D. developed software and a user interface to support real-time and offline avatar animation and, along with E.F.C., G.K.A. and S.L.M., designed the avatar-decoding methodology. S.L.M. and K.T.L. implemented the direct-decoding approach to decode articulatory gestures and, along with A.B.S., developed models to classify non-verbal orofacial movements and emotional expressions. D.A.M. managed and coordinated the research project and implemented real-time software infrastructure to collect data and run the tasks. D.A.M., S.L.M. and K.T.L. implemented real-time software to enable real-time text, speech and avatar decoding and designed the tasks and sentence sets. K.T.L. and I.Z. designed and conducted perceptual-accuracy evaluations with decoded speech and avatar animations. K.T.L., P.W. and G.K.A. developed the personalized voice synthesizer. K.T.L., S.L.M. and I.Z. analysed human-speaker data for avatar–human comparisons. A.B.S. and S.L.M. designed and carried out the phone-encoding analyses. A.B.S. designed and carried out the articulatory-encoding analyses. A.B.S., S.L.M. and K.T.L. carried out offline exclusion analyses to assess the importance of electrode density, feature sets and anatomical regions. J.R.L. designed and trained the speech detection model used in freeform text decoding. S.L.M., K.T.L. and A.B.S. carried out statistical analyses and, along with D.A.M., generated figures. D.A.M., K.T.L. and M.E.D. designed graphical user interfaces for text, speech and avatar decoding. S.L.M., K.T.L., A.B.S., D.A.M., M.P.S. and E.F.C. prepared the manuscript with input from other authors. M.P.S., M.E.D. and D.A.M. led the data-collection efforts with help from S.L.M., K.T.L., A.B.S., R.W. and J.R.L. M.P.S. recruited the participant, handled logistical coordination with the participant throughout the study and, together with A.T.-C., maintained and updated the clinical-trial protocol. M.P.S., A.T.-C., K.G. and E.F.C. carried out regulatory and clinical supervision. E.F.C. conceived, designed and supervised the study.

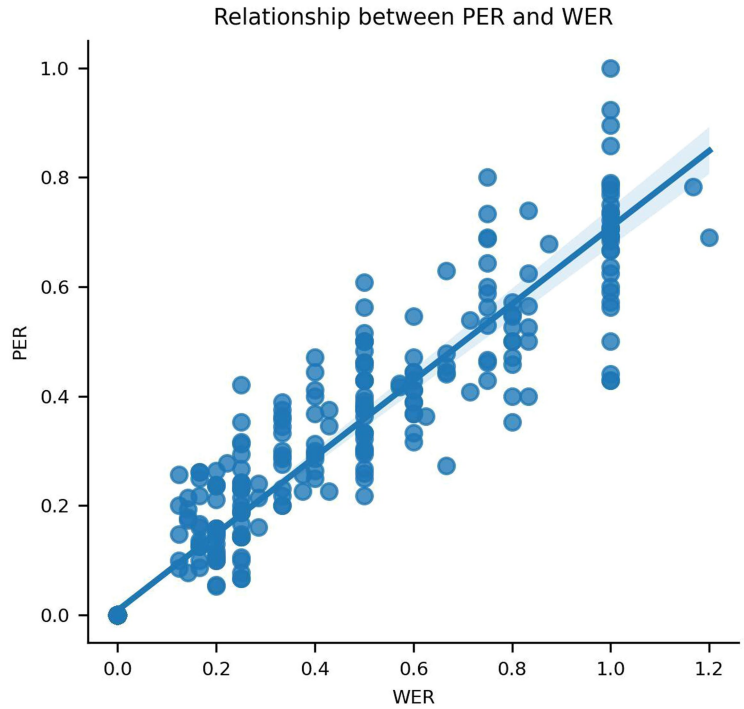
Competing interests S.L.M., D.A.M., J.R.L. and E.F.C. are inventors on a pending provisional UCSF patent application that is relevant to the neural-decoding approaches used in this work. G.K.A. and E.F.C. are inventors on patent application PCT/US2020/028926, D.A.M. and E.F.C. are inventors on patent application PCT/US2020/043706, and E.F.C. is an inventor on patent US9905239B2, which are broadly relevant to the neural-decoding approaches in this work. M.A.B. is chief technical officer at Speech Graphics. All other authors declare no competing interests.

Additional information

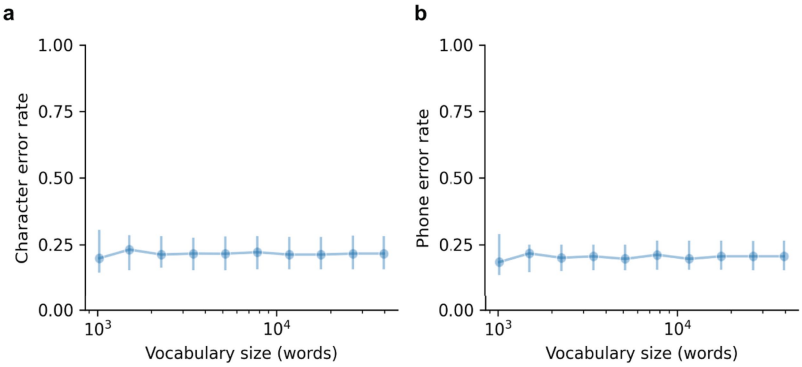
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41586-023-06443-4.

Correspondence and requests for materials should be addressed to Edward F. Chang.

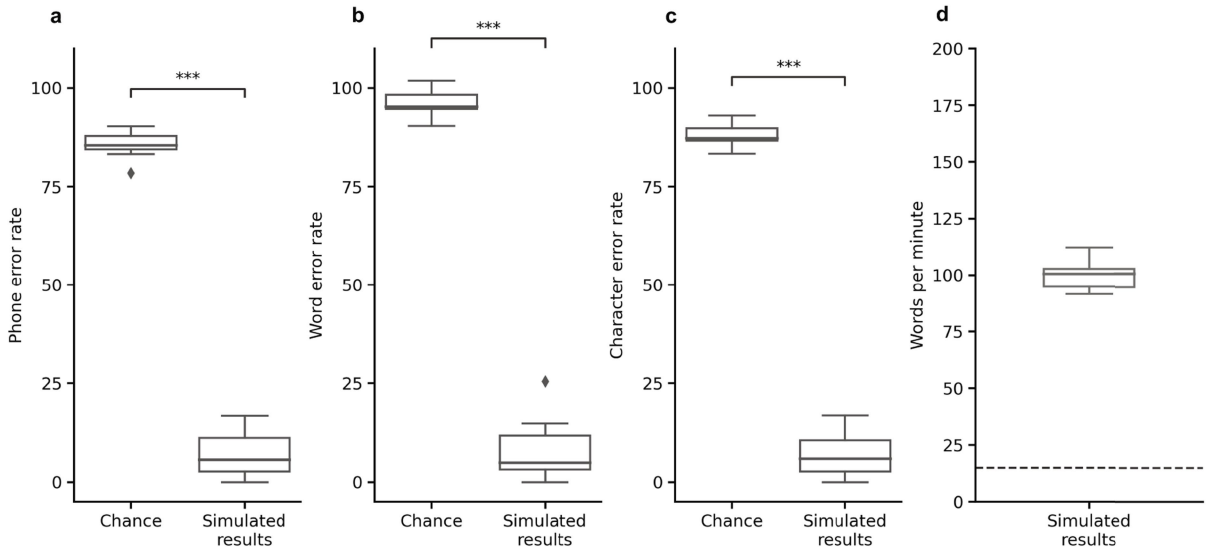
Peer review information Nature thanks Taylor Abel, Nicholas Hatsopoulos, Parag Patil, Betts Peters and Nick Ramsey for their contribution to the peer review of this work.



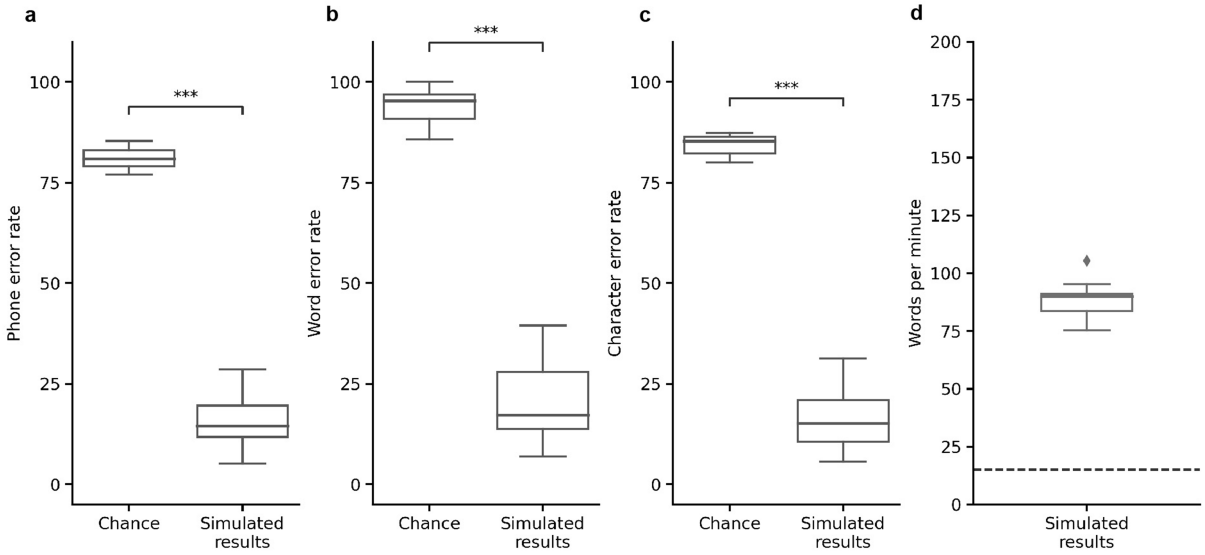
Extended Data Fig. 1 | Relationship between PER and WER. Relationship between phone error rate and word error rate across n = 549 points. Each point represents the phone and word error rate for all sentences used during model evaluation for all evaluation sets. The points display a linear trend, with the linear equation corresponding with an R^2 of .925. Shading denotes 99% confidence interval which was calculated using bootstrapping over 2000 iterations.



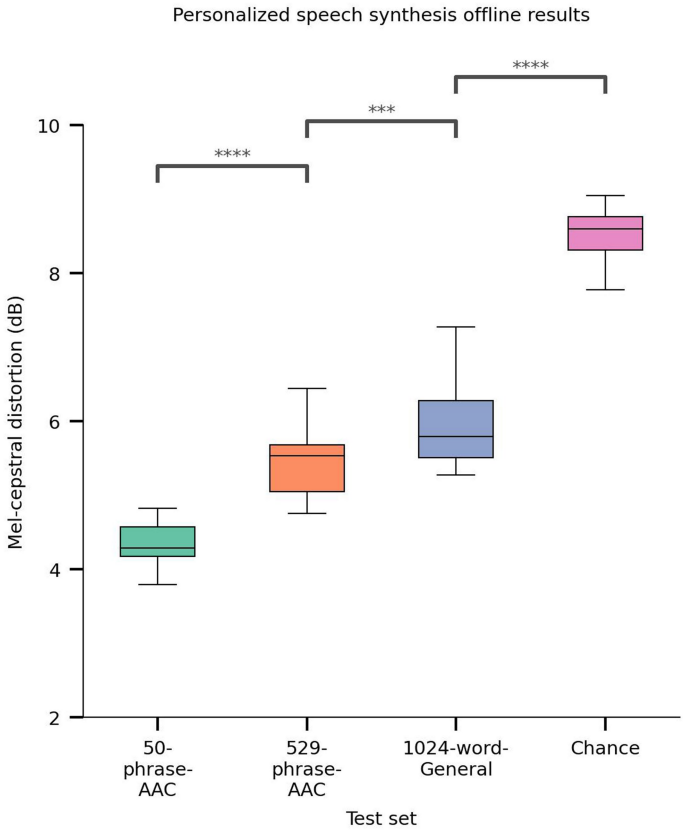
Extended Data Fig. 2 | Character and phone error rates for simulated text decoding with larger vocabularies. a,b, We computed character (**a**) and phone (**b**) error rates on sentences obtained by simulating text decoding with the 1024-word-General sentence set using log-spaced vocabularies of 1,506, 2,269, 3,419, 5,152, 7,763, 11,696, 17,621, 26,549, and 39,378 words, and we compared performance to the real-time results using our 1,024 word vocabulary. Each point represents the median character or phone error rate across n = 25 real-time evaluation pseudo-blocks, and error bars represent 99% confidence intervals of the median. With our largest 39,378 word vocabulary, we found a median character error rate of 21.7% (99% CI [16.3%, 28.1%]), and median phone error rate of 20.6% (99% CI [15.9%, 26.1%]). We compared the WER, CER, and PER of the simulation with the largest vocabulary size to the real-time results, and found that there was no significant increase in any error rate ($P > .01$ for all comparisons. Test statistic = 48.5, 93.0, 88.0, respectively, $p = .342, .342, .239$, respectively, Wilcoxon signed-rank test with 3-way Holm-Bonferroni correction).



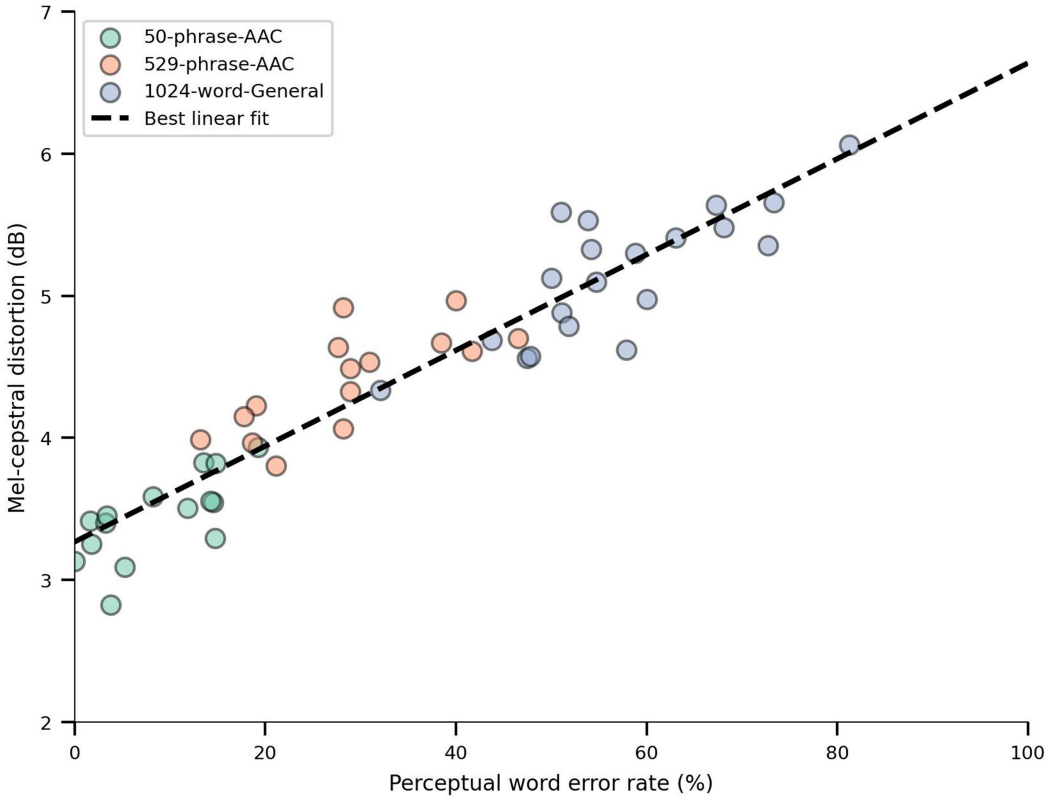
Extended Data Fig. 3 | Simulated text decoding results on the 50-phrase-AAC sentence set. a–**c**, We computed phone (**a**), word (**b**), and character (**c**) error rates on simulated text-decoding results with the real-time 50-phrase-AAC blocks used for evaluation of the synthesis models. Across n = 15 pseudo-blocks, we observed a median PER of 5.63% (99% CI [2.10, 12.0]), median WER of 4.92% (99% CI [3.18, 14.0]) and median CER of 5.91% (99% CI [2.21, 11.4]). The PER, WER, and CER were also significantly better than chance (P < .001 for all metrics, Wilcoxon signed-rank test with 3-way Holm-Bonferonni Correction for multiple comparisons). Statistics compare n = 15 total pseudo-blocks. For PER: stat = 0, P = 1.83e-4. For CER: stat = 0, P = 1.83e-4. For WER: stat = 0, P = 1.83e-4. **d**, Speech was decoded at high rates with a median WPM of 101 (99% CI [95.6, 103]).



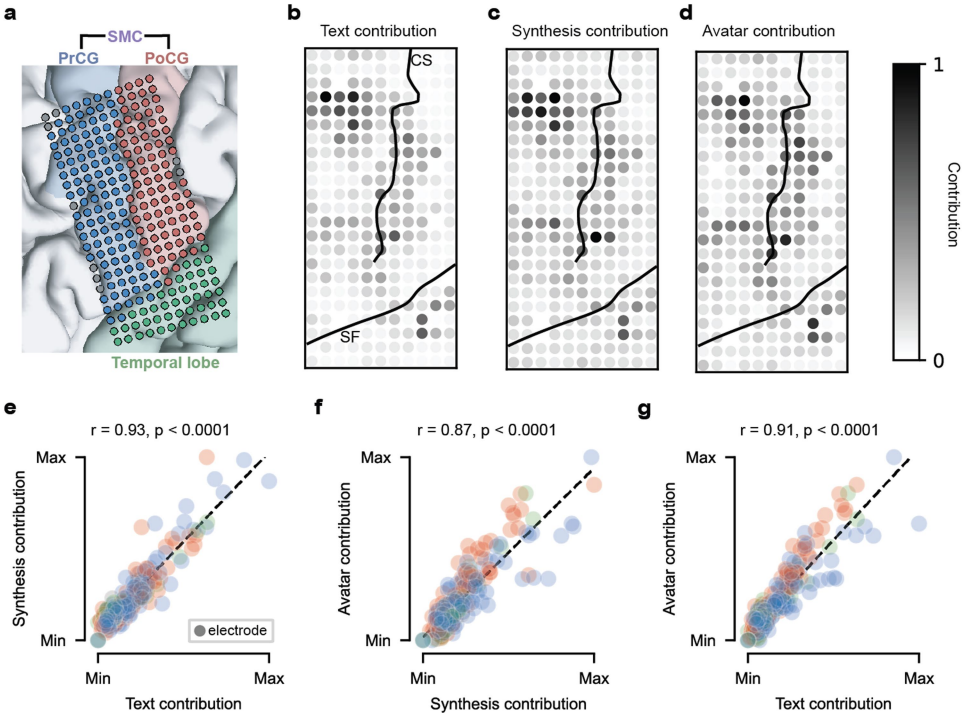
Extended Data Fig. 4 | Simulated text decoding results on the 529-phrase-AAC sentence set. a–**c**, We computed phone (**a**), word (**b**), and character (**c**) error rates on simulated text-decoding results with the real-time 529-phrase-AAC blocks used for evaluation of the synthesis models. Across n = 15 pseudo-blocks, we observed a median PER of 17.3 (99% CI [12.6, 20.1]), median WER of 17.1% (99% CI [8.89, 28.9]) and median CER of 15.2% (99% CI [10.1, 22.7]). The PER, WER, and CER were also significantly better than chance ($p < .001$ for all metrics, two-sided Wilcoxon signed-rank test with 3-way Holm-Bonferonni Correction for multiple comparisons). Statistics compare n = 15 total pseudo-blocks. For PER: stat = 0, p = 1.83e-4. For CER: stat = 0, p = 1.83e-4. For WER: stat = 0, P = 1.83e-4. **d**, Speech was decoded at high rates with a median WPM of 89.9 (99% CI [83.6, 93.3]).



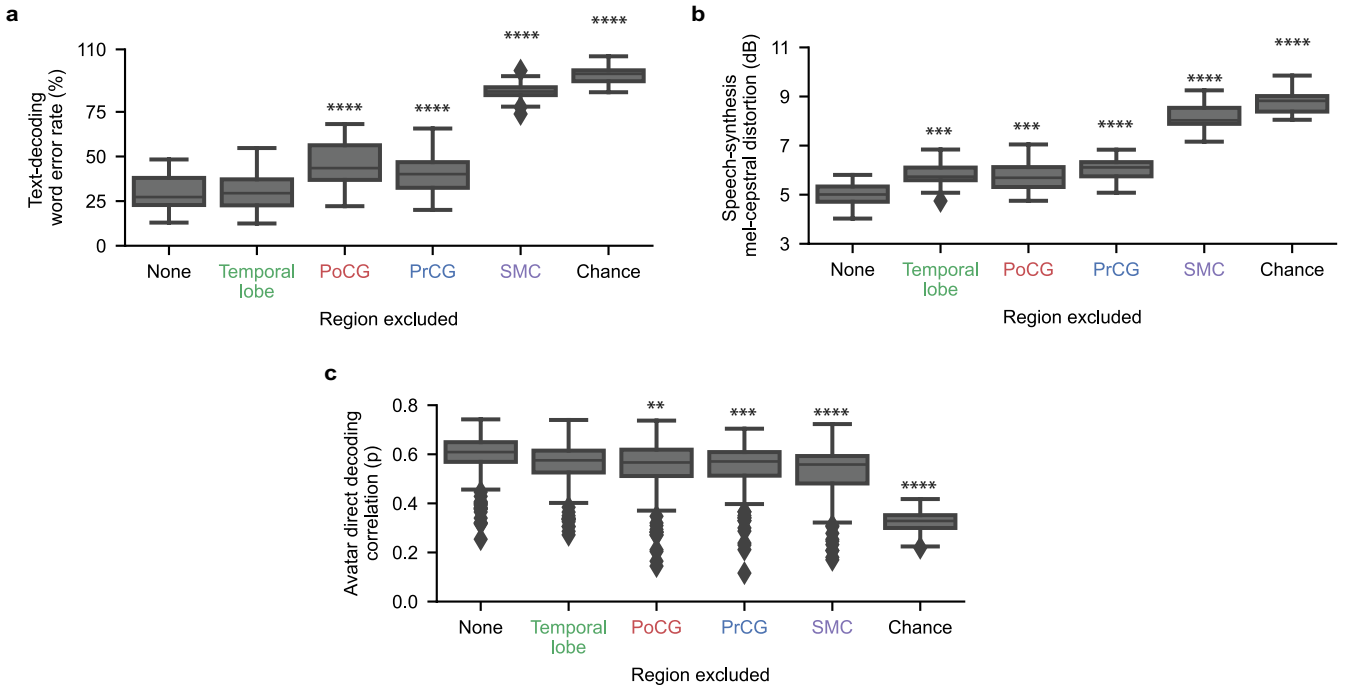
Extended Data Fig. 5 | Mel-cepstral distortions (MCDs) using a personalized voice tailored to the participant. We calculate the Mel-cepstral distortion (MCDs) between decoded speech with the participant's personalized voice and voice-converted reference waveforms for the 50-phrase-AAC, 529-phrase-AAC, and 1024-word-General set. Lower MCD indicates better performance. We achieved mean MCDs of 3.87 (99% CI [3.83, 4.45]), 5.12 (99% CI [4.41, 5.35]), and 5.57 (99% CI [5.17, 5.90]) dB for the 50-phrase-AAC (N = 15 pseudo-blocks), 529-phrase-AAC (N = 15 pseudoblocks), and 1024-word-General sets (N = 20 pseudo-blocks) Chance MCDs were computed by shuffling electrode indices in the test data with the same synthesis pipeline and computed on the 50-phrase-AAC evaluation set. The MCDs of all sets are significantly lower than the chance. 529-phrase-AAC vs. 1024-word-General $*** = P < 0.001$, otherwise all $**** = P < 0.0001$. Two-sided Wilcoxon rank-sum tests were used for comparisons within-dataset and Mann-Whitney U-test outside of dataset with 9-way Holm-Bonferroni correct.



Extended Data Fig. 6 | Comparison of perceptual word error rate and mel-cepstral distortion. Scatter plot illustrating relationship between perceptual word error rate (WER) and mel-cepstral distortion (MCD) for the 50-phraseAAC sentence set, the 529-phrase-AAC sentence set, the 1024-word-General sentence set. Each data point represents the mean accuracy from a single pseudo-block. A dashed black line indicates the best linear fit to the pseudo-blocks, providing a visual representation of the overall trend. Consistent with expectation, this plot suggests a positive correlation between WER and MCD for our speech synthesizer.



Extended Data Fig. 7 | Electrode contributions to decoding performance. **a**, MRI reconstruction of the participant's brain overlaid with the locations of implanted electrodes. Cortical regions and electrodes are colored according to anatomical region (PoCG: postcentral gyrus, PrCG: precentral gyrus, SMC: sensorimotor cortex). **b**–**d**, Electrode contributions to text decoding (**b**), speech synthesis (**c**), and avatar direct decoding (**d**). Black lines denote the central sulcus (CS) and sylvian fissure (SF). **e**–**g**, Each plot shows each electrode's contributions to two modalities as well as the Pearson correlation across electrodes and associated p-value.



Extended Data Fig. 8 | Effect of anatomical regions on decoding performance. a–c, Effect of excluding each region during training and testing on text-decoding word error rate (**a**), speech-synthesis mel-cepstral distortion (**b**), and avatar-direct-decoding correlation (**c**; average DTW correlation of jaw, lip, and mouth-width landmarks between the avatar and healthy speakers), computed using neural data as the participant attempted to silently say sentences from the 1024-word-General set. Significance markers indicate comparisons against the None condition, which uses all electrodes. *P < 0.01, **P < 0.005, ***P < 0.001, ****P < 0.0001, two-sided Wilcoxon signed-rank test with 15-way Holm-Bonferroni correction (full comparisons are given in Table S5). Distributions are over 25 pseudo-blocks for text decoding, 20 pseudo-blocks for speech synthesis, and 152 pseudo-blocks (19 pseudo-blocks each for 8 healthy speakers) for avatar direct decoding.

Extended Data Table 1 | Real-time text-decoding comparisons with the 1024-word-General sentence set

Comparison	Statistic[1]	Corrected P-value
PER Chance vs. Real-time results	0.00e+00	2.98e-07
PER Neural decoding only vs. Real-time results	0.00e+00	2.98e-07
PER Chance vs. Neural decoding only	0.00e+00	2.98e-07
WER Chance vs. Real-time results	0.00e+00	2.98e-07
CER Chance vs. Real-time results	0.00e+00	2.98e-07

Each comparison is a two-sided Wilcoxon Signed-Rank test across $n = 25$ pseudo-blocks, with 5-way Holm-Bonferroni correction. These comparisons were computed using real-time text-decoding results with the 1024-word-General sentence set, shown in Fig. 2 in the main text.

Extended Data Table 2 | Real-time audio-visual synthesis comparisons

Dataset 1	Dataset 2	Statistic	Corrected P-value
1024-word-General MCD	50-phrase-AAC chance MCD	1.10e+03	1.68e-48
50-phrase-AAC WER	1024-word-General WER	3.94e+03	3.09e-33
50-phrase-AAC CER	1024-word-General CER	3.96e+03	4.56e-33
1024-word-General MCD	1024-word-General chance MCD	7.70e+01	7.30e-33
1024-word-General CER	1024-word-General chance CER	1.96e+02	4.01e-32
1024-word-General WER	1024-word-General chance WER	2.55e+01	1.10e-29
50-phrase-AAC MCD	1024-word-General MCD	4.53e+03	6.59e-28
50-phrase-AAC WER	50-phrase-AAC chance WER	0.00e+00	1.35e-25
50-phrase-AAC MCD	50-phrase-AAC chance MCD	1.00e+00	2.58e-25
50-phrase-AAC CER	50-phrase-AAC chance CER	3.00e+00	2.58e-25
529-phrase-AAC MCD	529-phrase-AAC chance MCD	2.70e+01	3.56e-25
529-phrase-AAC CER	529-phrase-AAC chance CER	4.10e+01	6.13e-25
529-phrase-AAC WER	529-phrase-AAC chance WER	6.20e+01	7.52e-24
529-phrase-AAC WER	1024-word-General WER	8.34e+03	3.12e-12
529-phrase-AAC CER	1024-word-General CER	9.06e+03	6.45e-10
50-phrase-AAC MCD	529-phrase-AAC MCD	7.12e+03	1.56e-07
50-phrase-AAC CER	529-phrase-AAC CER	7.79e+03	2.43e-07
50-phrase-AAC WER	529-phrase-AAC WER	7.83e+03	2.43e-07
529-phrase-AAC MCD	1024-word-General MCD	1.04e+04	7.08e-07

Across-dataset comparisons use two-sided Mann-Whitney U-tests and within-dataset comparisons use two-sided Wilcoxon signed-rank tests. All with tests are with 19-way Holm-Bonferroni correction. We use n=15 pseudo-blocks for the AAC sentence sets, and n=20 pseudo-blocks for the 1024-word-General sentence set.

nature portfolio

Corresponding author(s): Edward Chang

Last updated by author(s): Jun 3, 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	Open source: Python 3.6 and Python 3.8 Commercial software: We used hardware manufactured by Blackrock Microsystems (along with the associated firmware and the NeuroPort Central Suite software package (version 7.0.6), the proprietary software suite used to control their hardware) to acquire and pre-process data from the neural implant. Custom code: We used a real-time software package (referred to as rtNSR; described in prior literature that is referenced in the current manuscript) to process neural data in real time
Data analysis	Open source: - Python 3.8 - NumPy 1.23.3 - scikit-learn 0.24.2 - pandas 1.5.0 - SciPy 1.8.1 - PyTorch 1.10 - torchaudio 0.12.0 - NLTK 3.7 - statsmodels 0.13.5 - statannot 0.2.3 - wandb 0.13.4

- kenlm 0.0.0
- fastdtw 0.3.4
- g2p-en 2.1.0
- seaborn 0.12.1
- UnrealEngine 4.26
- SG Com 4.1.18-alpha
- fairseq 0.12.2
- lm-scorer 0.4.2
- jiwer 2.5.1
- librosa 0.9.1
- Levenshtein 0.20.2
- PyQT5 5.12
- pysptk 0.1.21
- PyAudio 0.2.12
- transformers 2.11.0
-Statsmodels 0.12.2
-Scikit-learn 0.24.2-
-NumPy 1.19.5
-Pandas 0.25.3
-SciPy 1.5.4
-PyTorch 1.12.1
-Torchaudio 0.12.1
-Kenlm
-Seaborn 0.11.2
-MNE 0.23.4

Custom code: We used custom scripts and modules to analyze the neural data and implementing real-time prediction and avatar animation. Relevant code to replicate the main findings of our work will be made available to editors and reviewers upon request, and released to the public on github and via deposition in a Zenodo repository upon publication

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

Relevant data will be made available upon reasonable request, per the guidelines from our clinical-trial protocol which allow us to share de-identified data with researchers at other institutions but precludes us from making all of the data publicly available. Requests for relevant data can be made to Dr. Edward Chang (edward.chang@ucsf.edu). Responses can be expected within 3 weeks. Any provided data should be kept confidential and should not be shared with others unless approval to do so is obtained from Dr. Chang.

Source data to re-create the manuscript figures are included with this submission.

Human research participants

Policy information about studies involving human research participants and Sex and Gender in Research.

Reporting on sex and gender	The single participant in the study is female. Sex and gender effects were not researched in this study.
Population characteristics	The single participant in the study is a female of age 47 at the start of the study with severe spastic quadriparesis and anarthria caused by brainstem stroke.
Recruitment	Participants with motor impairments secondary to neurological disorders were recruited from stroke, ALS, and Neurology clinics at UCSF. Prior to study enrollment, researchers conduct an informal phone interview to go through study information and schedule an in-person visit. This in-person office visit is then followed by three additional outpatient screening visits. We follow an iterative enrollment process, and allow the participant to learn about the study more during each visit. During the first visit, the participant has the study explained to them in detail, and has the chance to ask any questions. Should the participant wish to continue, the next in-person visit will include a physical exam and further screening to determine eligibility. After this visit, an MRI and CT of the brain is obtained for future surgical planning and eligibility purposes. As part of general pre surgical testing, an ECG and chest X-ray are also obtained. The third follow-up visit is included to review this data and address any participant questions before enrollment in the clinical trial. To be eligible for enrollment, participants must fit specific clinical characteristics (see attached clinical protocol for full

	inclusion and exclusion criteria). As this eligibility criteria is quite specific, we do not expect any significant self-selection bias in this study, due to the fact that participants who volunteer to participate will not differ from those who choose not to volunteer in any relevant clinical characteristics.
Ethics oversight	The study protocol was approved by the USFDA, UCSF IRB, and the National Institute on Deafness and Communication Disorders at the NIH.

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	Sample sizes for tests were determined by the amount of available data with the participant. The amount of data collected with the participant was dependent on our estimation of how much data (e.g. how many trials of the tasks) would be required to reasonably estimate the measurements of interest and perform statistical comparisons.
Data exclusions	After collecting the real-time testing data, we identified that due to an error, for one of the 250 sentences originally in the test set for text decoding, the participant had attempted to say the same sentence for one trial in the training set. To keep our results with the 1024-word-General set a measure of performance on previously unseen sentences, we decided to exclude this trial and report performance on the remaining 249 sentences that were not used during model training. Thus, that sentence was excluded from evaluations.
Replication	Decoding performance across pseudo-blocks of 10 sentences was reasonably consistent, as described in the manuscript. However, this does not provide definitive evidence of reproducibility. True replication of the results might require deployment of a functionally equivalent system in another participant with a similar level of paralysis and anarthria, which was not feasible here because we only have one active clinical-trial participant with the current version of our device for this proof-of-concept study at the time of writing.
Randomization	Randomization was not critical to this single-participant study.
Blinding	Because the goal of the study was to demonstrate the feasibility of a multimodal brain-computer interface controlled by a paralyzed person's silent-speech attempts, blinding was not relevant to this study.

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

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	Clinicaltrials.gov NCT03698149
Study protocol	A description of the study can be found at our clinicaltrials.gov website at https://clinicaltrials.gov/ct2/show/NCT03698149. The original and current clinical protocols are provided as a supplementary file alongside this article.

Data collection

Data collection occurred in the participant's living room. The clinical trial began in November 2018, however this participant was recruited and enrolled in the study following implantation of the study device during September 2022 at UCSF Medical Center.

The currently approved duration of the larger clinical trial is 6 years. This is with the expectation that all 8 patients will be recruited in the first 4–5 years and all other activities including data collection, device development, and analysis will be completed in 6 years. Further information on data collection procedures can be found in the attached clinical-trial protocol.

Outcomes

This clinical trial is a phase I single-center early feasibility study to evaluate the long term potential of ECoG-based neural interfaces to control advanced neuroprotheses for motor and communication restoration. This is a low participant enrollment, exploratory clinical trial, therefore secondary outcomes were not pre-defined. Our primary, pre-definied endpoint is to assess the safety and efficacy of the trial device. This is stated in the protocol as "Feasibility of control of a wearable exoskeleton device and a communication interface." With this primary end-point in mind, and with the exploratory nature of the trial guiding future directions, a variety of analysis and testing methods will be explored with trial participants throughout the trial to assess the efficacy of an ECoG-based neural interface for motor and communication restoration.