Supplementary Materials for

Fully portable and wireless universal brain-machine interfaces enabled by flexible scalp electronics and deep-learning algorithm

Musa Mahmood[1], Deogratias Mzurikwao[2], Yun-Soung Kim[1], Yongkuk Lee[3], Saswat Mishra[1], Robert Herbert[1], Audrey Duarte[4], Chee Siang Ang[2] and Woon-Hong Yeo[1,5,6,*]

[1]George W. Woodruff School of Mechanical Engineering, Institute for Electronics and Nanotechnology, Georgia Institute of Technology, Atlanta, Georgia 30332, USA.

[2] School of Engineering and Digital Arts, University of Kent, Canterbury, Kent CT2 7NT, United Kingdom

[3]Department of Biomedical Engineering, Wichita State University, Wichita, Kansas 67260, USA

[4]School of Psychology, College of Sciences, Georgia Institute of Technology, Atlanta, Georgia 30332, USA

[5]Wallace H. Coulter Department of Biomedical Engineering, Parker H. Petit Institute for Bioengineering and Biosciences, Georgia Institute of Technology, Atlanta, Georgia 30332, USA.

[6]Neural Engineering Center, Center for Flexible and Wearable Electronics Advanced Research, Institute for Materials, Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, Georgia 30332, USA.

*Corresponding author. Email: whyeo@gatech.edu (W.-H.Y.)

Section S1. Topographic Scalp Mapping using Deep Learning

When using as few as two EEG data channels (3 measurement electrodes), it is important to verify that the electrode locations are optimal for that subject. Therefore, 4-class SSVEP examinations were undertaken with a 32-channel EEG (ActiveTwo System, Biosemi B.V., Amsterdam), with setup shown in **Fig. 1a**. A convolutional neural network (CNN) is then trained to classify the data. From the trained network, relevant weights correlated to information presence may be extracted and associated to specific channels. Electrode placement is then decided based on the channel locations with the most significant weights. The architecture used is a 2-layer deep convolutional neural, *2-CNN-Mapping*, with max pooling using only vector kernels, detailed in **Table S1**. The input was split into epochs of 300 data points (1.172 seconds at 256 Hz). Each epoch was high-pass filtered at 4 Hz and normalized from the range of 0 to 1. Therefore, the inputs were 32 channels x 300 data points, and convolutions were performed along the time domain signal to prevent data mixing across channels. The weights are extracted from the feature maps formed in the first convolution layer. The sum of weights of all feature maps was obtained as we have 32 feature maps in the first convolution layer. We used the first convolution layer as the feature maps are still in the same shape as the input before max-pooling. The weights for each feature map were calculated using the following formula. w_l stands for window length, which is 300 and p is the number of channels.

$W_p = \sum_{q=0}^{q=w_l} w_{(p,q)}$, where $0 \leq p < 32$ and $0 \leq q < w_l$

To get the total weight W for each channel p per participant, we then sum the weights of all feature maps r in the first convolution layer which has 32 feature maps as expressed in equation: $W = \sum_{r=1}^{32} (W_p)_r$. A resulting scale showing the activation weights for each channel over the 32 feature maps is shown in **Fig. S1b**. The topographic maps created from the weights extracted from the trained model shows the regions with highest weights are situated around the virtual cortex region of the human brain, as can be seen in **Fig**. **S1c-d**. This exercise is meant to be a proof-of-concept method for mapping biopotential signals from a range of sources with the intention of reducing the number of electrodes and amount of data to process.

Section S2. Fabrication procedures.

Device fabrication procedure is illustrated in **Fig. S3**. Detail procedures involved with device preparation is described here:

a) Microfabrication of the thin-film circuit boards

1. Spincoat PDMS on a cleaned silicon wafer at 3000 rpm for 30 sec.
2. Cure the PDMS-coated wafer on a 150 °C hot plate for 5 min.
3. Perform oxygen plasma treatment to render the PDMS hydrophilic.
4. Spincoat polyimide at 4000 rpm for 1 min and bake in a vacuum oven at 250 °C for 3 hr including ramping steps.
5. Sputter 500 nm copper for the 1st metal layer.
6. Spincoat photoresist (SC1827) at 3000 rpm for 30 sec and bake it on a 110 °C hot plate for 1 min.
7. Expose UV with the 1st metal pattern (ground) using a mask aligner (MA6).
8. Develop the exposed photoresist with developer (MF-319).
9. Etch exposed copper with copper etchant (APS-100, diluted 1:1 with DI water) and strip photoresist.
10. Spincoat polyimide at 850 rpm for 1 min bake in a vacuum oven at 250 °C for 3 hr including ramping steps.
11. Spincoat polyimide at 850 rpm for 1 min bake in a vacuum oven at 250 °C for 3 hr including ramping steps.
12. Spincoat photoresist (AZP4620) at 1000 rpm for 30 sec and bake it on a 95 °C hot plate for 4 min.
13. Expose UV with the 1st via pattern using a mask aligner (MA6).
14. Develop the exposed photoresist with developer (AZ400K, diluted with 4 parts of DI water).
15. Oxygen plasma etch exposed PI using reactive ion etching (Plasma-Therm) and strip photoresist.
16. Sputter 1.5 μm copper for the 2nd metal layer.
17. Spincoat photoresist (AZP462) at 3000 rpm for 30 sec and bake it on a 95 °C hot plate for 4 min.
18. Expose UV with the 2nd metal pattern using a mask aligner (MA6).
19. Develop the exposed photoresist with developer (AZ400K, diluted with 4 parts of DI water).
20. Etch exposed copper with copper etchant (APS-100, diluted 1:1 with DI water) and strip photoresist.
21. Spincoat polyimide at 4000 rpm for 1 min and bake in a vacuum oven at 250 °C for 3 hr including ramping steps.
22. Spincoat photoresist (AZP462) at 2000 rpm for 30 sec and bake it on a 95 °C hot plate for 4 min.
23. Expose UV with the solder pad exposure pattern using a mask aligner (MA6).
24. Develop the exposed photoresist with developer (AZ400K, diluted with 4 parts of DI water).
25. Oxygen plasma etch exposed PI using reactive ion etching (Plasma-Therm) and strip photoresist.
26. Completed circuits are peeled-off from the PDMS wafer and transferred onto a glass slide for chip assembly.

b) Flexible circuit board assembly using reflow soldering
    1. Openings of the stainless steel stencil is aligned to the exposed copper pads.
    2. Solder paste is dispensed onto the stencil is dragged across the stencil with a razor blade.
    3. Chip components are placed on the board.
    4. Using a programmable hot plate, the components are reflow soldered by ramping the hot plate from 100 ℃ to 170 ℃ at an increment of 10 ℃ with a 1 min dwell period at each temperature.
    5. Soldered board is rinsed with acetone and isopropyl alcohol and dried with a stream of nitrogen.
    6. The board is detached from the glass slide and is attached to a thin layer of elastomer (Ecoflex).

c) Dry electrode integration and device encapsulation
    1. Flexible film cables are obtained by cutting the cables into individual strips.
    2. PDMS is applied on the backside of the strips, where the conductive trace is exposed, and cured.
    3. Using silver paint, the flexible device and the dry electrode are bonded on the two ends of the cables.
    4. Small amount of epoxy is applied to the tip of the cable on the dry electrode side to enhance the attachment.
    5. Ecoflex is applied on top of the circuit for full device encapsulation.

Section S3. Fabrication of Skin Electrodes via Aerosol Jet Printing.

Silver skin electrodes were printed using an Aerosol Jet Printer (AJ200, Optomec, Albuqeurque, NM). This additive manufacturing process involves depositing polyimide and silver nanoparticle ink in xylene onto a glass slide prepared with spin-coated Polymethyl methacrylate (PMMA A7). The PMMA is then dissolved for 15 minutes in an acetone bath at room temperature, and the finished electrodes are transferred to an elastomer (Ecoflex Gel, Smooth-On, USA) using water-soluble poly-vinyl alcohol (PVA) tape (3M, Maplewood, Minnesota).

Step-by step instructions:

1. Clean glass slide or silicon wafer with acetone and IPA.
2. Plasma clean at 18 W for 30 seconds.
3. Spin coat PMMA in Anisole at 4000 RPM.
4. Bake the 700 nm PMMA at 180 °C for 2.5 minutes.
5. Align sample onto platen stage of the AJ200 and leave platen stage at 70 °C.
6. Change deposition head to 300 µm nozzle.
7. Enable the pneumatic atomizer to print polyimide 2545 5:1 mixture with N-methyl-2-pyrrolidone.
8. Turn on sheath flow rate and 100 SCCM, atomizer flow rate at 1500 SCCM, and exhaust at 1470 SCCM.
9. Print two to five layers of the polyimide patterns at 10 mm/s on the PMMA substrate.
10. Ensure atomization is dense on the side of the pneumatic atomizer.
11. Cure the polyimide at 250°C in an ambient oven for 1 hour.
12. Change deposition head to 150 µm nozzle.
13. Align fiducial markers of the polyimide layer to deposit the silver layer.
14. Enable the ultrasonic bath on the AJ200 at 30°C.
15. Turn on sheath flow rate between 5 SCCM and 30 SCCM.
16. Turn on atomizer flow rate between 15 SCCM and 30 SCCM
17. Turn on atomizer current at .6 Amps.
18. Run the silver ink pattern at 10 mm/s on the polyimide patterns.
19. Sinter the silver ink at 200 °C for 1 hour
20. Place a texwipe and a glass slide to sandwhich the samples for the acetone bath.
21. Place the samples in an acetone bath for 1 hour at 60 °C.
22. Prepare a sheet of PVA by laminating onto a glass slide.
23. Heat treat at 100 °C for 1 minute to remove wrinkles and release stress.
24. Spin coat ecoflex gel (1:1) at 2000 RPM for 1 min.
25. Let cure at room temperature for 2 hours.

26. Remove the samples from the acetone bath and apply water soluble tape onto the electrodes.
27. Transfer the electrodes onto the ecoflex gel elastomer and dissolve the tape.
28. Attach anisotropic conductive film (ACF) wires to the electrode pads using silver paint.

Section S4. Resistance measurement of electrode compression

For cyclic compression of the hair electrode a motorized tension/compression test stand (ESM303, Mark-10, USA) was used to push the electrode with a controlled displacement of 2 mm. Two conductive film wires (HST-9805210, Elform, USA) were attached with silver paint (PELCO Conductive Silver Paint, Ted Pella, USA) to monitor the resistance across one of the flexible legs using a digital multimeter (DMM7510, Keithley Instruments, USA). Placed between two metal compression plates, the hair electrode underwent a repeated compression of 1000 cycles, with each cycle lasting approximately 6 seconds. **Fig. S6** shows the testing setup used for this experiment with multimeter and laptop for recording results (**Fig. S6a**), and the positioning of the electrode between the compression plates. (**Fig. S6b**). The results from this test are shown in **Fig. 7**, with the entire captured resistance data shown in **Fig. 7a**, showing a general decrease in resistance over the test period. A single cycle in this process is expanded in **Fig. 7b**, showing how the resistance increases under compression for the first three seconds, and returns to normal on decompression. A moving average is computed using a window of 90 samples (approximately 6 seconds) to show the reduction in resistance over 1000 cycles, showing a drop of -4.46kΩ.

Section S5. Frequency component analysis with CCA.

Canonical correlation analysis (CCA) is a well-established method for decoding SSVEP signals using frequency component analysis[1-3]. For two multidimensional random variables, X and Y, and corresponding linear combinations $x = X^T W_x$ and $y = X^T W_y$ CCA generates pairs of linear combinations W_x and W_y, for sets X and Y such that the correlation between the two canonical variables is maximized. CCA may be used to classify where an unlabeled dataset may be classified using 'template' datasets for each class. In this case, the unlabeled SSVEP dataset is the recorded training data, and that is compared against template reference signals which is generated using the sine and cosine functions, including only the first and second harmonics.

$$Y_f = \begin{pmatrix} \sin(2\pi ft) \\ \cos(2\pi ft) \\ \sin(2\cdot 2\pi ft) \\ \cos(2\cdot 2\pi ft) \end{pmatrix}$$

Therefore, we find the maximum canonical correlation, ρ, with respect to W_x and W_y

$$\max_{W_x, W_y} \rho(x, y) = \frac{E\left[W_x^T X Y^T W_y\right]}{\sqrt{E\left[W_x^T X X^T W_x\right] E\left[W_y^T Y Y^T W_y\right]}}.$$

To determine the output class, we find the index of the maximum correlation from ρ:

$$y_{fit} = \underset{i}{\operatorname{argmax}}\ \rho_i, \qquad i = 0,\ 1,\ 2,\ ...,\ \mathrm{K}$$

where classes *i* correspond to frequencies from reference signals $f_0, f_1, \ldots, f_K$.[1, 3]

Section S6. Power-spectral density analysis (PSDA).

For filtering, baseline drift and DC offset were removed using a 3rd order Butterworth high-pass filter, with a cutoff frequency of 4 Hz. The main features of importance to SSVEP data is the frequency content above 5 Hz. To effectively use highly variable EEG signals for classification, signals must be preprocessed in a manner that highly normalizes its features. In this respect, FFT or power-spectral density estimation (PSD) are typically used as feature extraction methods for SSVEP[4-6]. All power-spectral density estimation methods in this paper use taper or multitaper methods using a Hann window to reduce bias from the initial Fourier spectral estimate. Hann windows are selected as they offer a reliable middle-ground between spectral leakage and smearing bias[7, 8]. Bartlett described how to estimate power spectra and reduce noise (smoothing) in periodograms by splitting a signal into non-overlapping segments and summing the corresponding spectrums[9]. We begin our analysis with a case where the window length is the same as the entire signal, which we will refer to as a single windowed periodogram. A standard single-windowed periodogram spectral estimate without time averaging is quantified by:

$$\hat{S}_{xx}(f) = \frac{2 \cdot \mathrm{FFT}(\mathrm{x} \cdot \mathrm{w}) \cdot conj(\mathrm{FFT}(\mathrm{x} \cdot \mathrm{w}))}{f_s \cdot \sum w_i^2}$$

Such a method is sensitive to random zero-mean noise, so time-averaging is often required for more representative estimates. Welch's method, which involves time-averaging through 50% overlapping windows aims to alleviate some of these issues. Welch's method aims to reduce noise in exchange for a reduction in frequency resolution. This is done by using a window *w*, that is smaller than the length of discrete time signal *x*, and overlapping the windows by 50%[10].

We calculate the individual spectrums:

$$\hat{S}_{xx}^{(k)}(f) = \frac{2 \cdot \mathrm{FFT}(\mathrm{x}^{(k)} \cdot \mathrm{w}) \cdot conj(\mathrm{FFT}(\mathrm{x}^{(k)} \cdot \mathrm{w}))}{f_s \cdot \sum w_i^2}$$

Then sum up the spectra, where N is the length of the original signal, and L is the length of the window:

$$\hat{S}_{xx}(f) = \sum_{k=1}^{\lfloor N/L \rfloor} \frac{S_{xx}^{(k)}}{\lfloor N/L \rfloor}$$

Note that Welch's method is not used for 0.512s inputs, as the frequency resolution is excessively reduced, and cannot be classified.

Section S7. Cross-spectral density analysis (CSDA).

Along with single-spectra periodograms and power-spectral density estimations using Welch's method, cross-spectrum power-spectral density is also used. These cross-spectra are estimated using Welch's method. For *x* and *y* signals of length *N*, we may break x and y into overlapping segments of size L. Where w is a tapering Hann window of specified length, where FFT () is the discrete Fourier transform using only the real positive frequencies.

This way, the equivalent block of cross-spectrums is estimated by

$$\hat{S}_{xy}^{(k)}(f) = \frac{2 \cdot \mathrm{FFT}\left(\mathrm{x}^{(k)} \cdot \mathrm{w}\right) \cdot conj\left(\mathrm{FFT}(\mathrm{y}^{(k)} \cdot \mathrm{w})\right)}{f_s \cdot \sum w_i^2}$$

The full cross spectrum is achieved by averaging the terms in the block of cross-spectrums:

$$\hat{S}_{xy}(f) = \sum_{k=1}^{\lfloor N/\mathrm{L} \rfloor} \frac{S_{xy}^{(k)}}{\lfloor N/L \rfloor}$$

This method is shown to closely estimate PSD which is calculated by integrating across the signal, for known power spectra[11]. Cross-spectral estimation has been used in analysis of electrophysiological applications in the past[12, 13]. Nolte et al. and others have used cross-spectrum estimation to establish linear relationships (coherence) between two or more EEG channels using the Fourier transform of the data[14, 15]. Note that Welch's method is not used for 0.512 sec inputs, as the frequency resolution is excessively reduced, and cannot be classified.

Section S8. Optimization of Convolutional Neural Network via Grid Search.

For optimizing the CNN components and basic hyperparameters, the design was constrained to 1 convolutional layer for simplicity. Here, a deep-learning based CNN was modeled on an input of time-domain, PSDA, or CSDA SSVEP data with features normalized between 0 and 1. Window sizes of 128, 192, 256, 384, 512 data points (corresponding to 0.512, 0.768, 1.024, 1.536, 2.048 sec data windows) were selected to gauge changes in accuracy over different window sizes. The inputs for two channels were as [*w* x 2] for time-domain data, where *w* is the length of the window in data points. For the PSDA, the input size was [*w*/2 x 2], as the power spectrum is half the size of the window used to create it, scaled from 0 Hz to the frequency range; 125 Hz in this case. Consequently, the CSDA data contains a cross-spectrum in addition to the power spectrums, and therefore the input [*w*/2 x 3] was required for this method. The baseline model was determined through a grid search approach to determine which model components and hyperparameters provide the best performance for a single-layer convolutional neural network (CNN).

The optimal learning rate was discovered to be on the order of 1e-3, as seen in **Fig. S12a**. The number of units in the fully connected layer units was altered between 64 and 4096 as seen in **Fig. S12b**. It was found that 1024 fully connected units was preferred as values greater than 1024 did not offer any marked improvement but resulted in extended training and evaluation times. Dropout regularization was then added to reduce overfitting, as it has proven highly effective in image classification[16]. As seen in **Fig. S12c**, dropout regularization considerably decreases the number of training steps required to converge and the best result of 50% dropout was found to improve training time considerably. Next, nonlinearities were introduced separately into the convolutional layers and the fully connected layers. Activation nonlinearity via LeakyReLU activations (LReLU) was found to be mostly detrimental in the fully connected layer, as shown in **Fig. S12d**. However, in the convolutional layers, activation nonlinearity was particularly useful in decreasing the amount of time required to converge toward optimized models. As seen in **Fig. S12e**, ELU activations increased the convergence rate the most, followed by increased α values with LeakyReLU activations. Finally, different convolution kernel sizes were attempted, varying filter sizes from [2 x 2] up to [12 x 12].

The optimal configuration for time-domain inputs of length 128 (0.512 seconds) is described in **Table S3**, using [8 x 8] convolution kernels instead of [2 x 2]. Selected results for these experiments are shown in **Table S4**, where [8 x 8] and [12 x 12] filters provided similar results. The varied convolution kernels for CSDA inputs are shown in **Table S5**, where it was shown that [4 x 4] kernels were most effective. At this stage, CSDA inputs achieve accuracies up to 92.33 ± 1.05% versus time-domain (92.81 ± 0.92%). This result suggests that there is no clear advantage to using either method, and further optimization is required. An analysis of the hidden layers of a single-layer CNN is shown in **Fig. S13**. Note that the level of activity in the fully connected layers of PSDA and CSDA versus that of the time domain model. The time-domain model appears to suffer from lower neuronal activity versus the frequency domain models. This is an expected result of higher entropy data source with fewer low-level detectable features.

The accuracies of this model may be improved significantly by adjusting the 2D convolution stride path. Using convolutions with increased stride is a reliable alternative to down sampling via max pooling[17]. As the data is densely packed, excessive down sampling destroys relevant information. Therefore, it was determined that the optimal model contains two convolutional layers with a stride of [1, 2] on the first convolutional layer, and a stride of [2, 2] on the second, in conjunction with square filters. However, this method requires many more filters than the single-layer models previously mentioned. In this case 50 and 100 filters were used with the two convolutional layers,

as seen in **Table S9** for time-domain inputs, and **Table S10** for CSDA inputs. Cross-validating across each of six subjects, where data from the remaining five were used to train, the time domain model achieved 94.54 ± 0.90% (ITR of 122.1 ± 3.53 bits min^{-1}, w=0.512s + 0.492s gaze shift) for 0.512 second inputs, versus 88.88 ± 2.17% with the baseline model. Comprehensive results for time domain can be seen in **Table S11** and **Table S12**. The optimized 2-layer CSDA model achieved 92.58 ± 1.07% for 0.512 second inputs, versus 91.98 ± 1.12% using the baseline model, for a maximum ITR of and 112.25 ± 6.81 bits min^{-1}. The detailed results for CSDA inputs may be found in **Table S13** and **Table S14** for CSDA inputs. These values may be compared to other significant results in published literature in **Table 1**.

The hidden layers for the 2-layer optimized model for CSDA inputs is shown in **Fig. S14**, showing a grey-scale representative of the outputs at each of the convolutional layers. This shows how only a small part of the input signal is deemed to be relevant on the left side. The most significant confusion occurred between alpha rhythms and SSVEP classes, a problem that occurred in both models. The confusion matrices shown in **Fig. S15 (a-b)** suggest that further performance improvements are to be gained from using frequencies that do not conflict with alpha rhythm frequencies. The average fully connected layer feature maps in **Fig. S15 (c)** show that many active features are similar with the alpha rhythm and 11.1 Hz classes. This suggests that the low-level features are very similar in these classes, and the convolutional neural network is unable to adequately separate them. It is worth noting that this also occurs with CSDA data, as seen in **Fig. S15 (d)**, but there is far less activity on the CSDA fully connected layer overall due to the similarities in the CSDA data.

Supplementary Figures

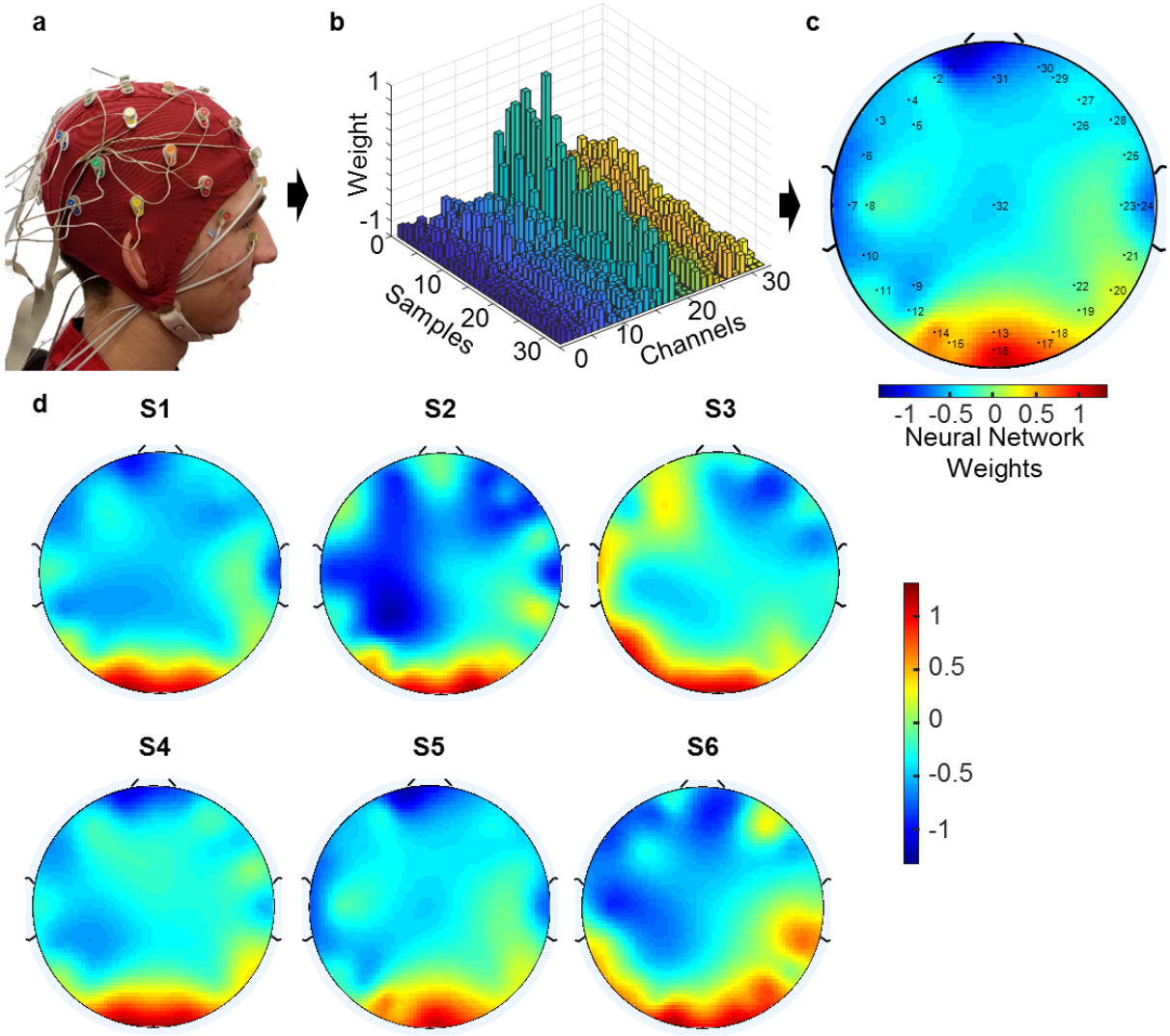


Figure S1. Topographical mapping of EEG signals using convolutional neural networks. (a) 32-channel cap EEG setup on test subject. (b) Weights extracted from the 2-CNN-Mapping convolutional neural network. From 32 convolutional filters used across all channels, the average weights for each point are summed, resulting in the weight mappings for each of the six subjects as illustrated in (c). The color scale represents the rescaled average neural network weights based on n=200 samples (40 samples per class). (d) shows the mappings generated for all 6 subjects, labeled correspondingly.

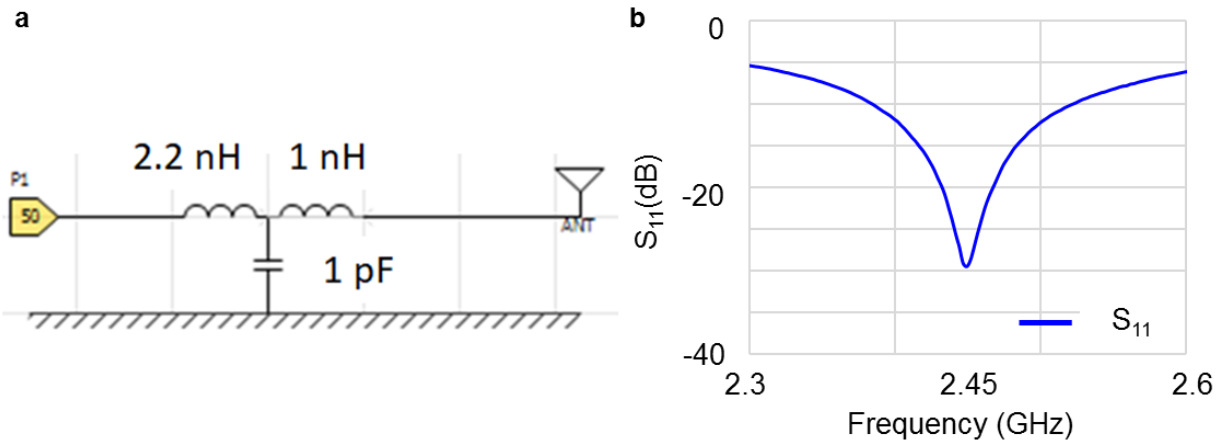


Figure S2. Antenna Matching Circuitry for SKINTRONICS wireless EEG. (a) shows the components in the antenna matching configuration optimized for use with the thin multilayer flexible SKINTRONICS format. **(b)** Shows reflection coefficient (S_{11}) measurement with a network analyzer in air, with the device placed flat on a glass slide.

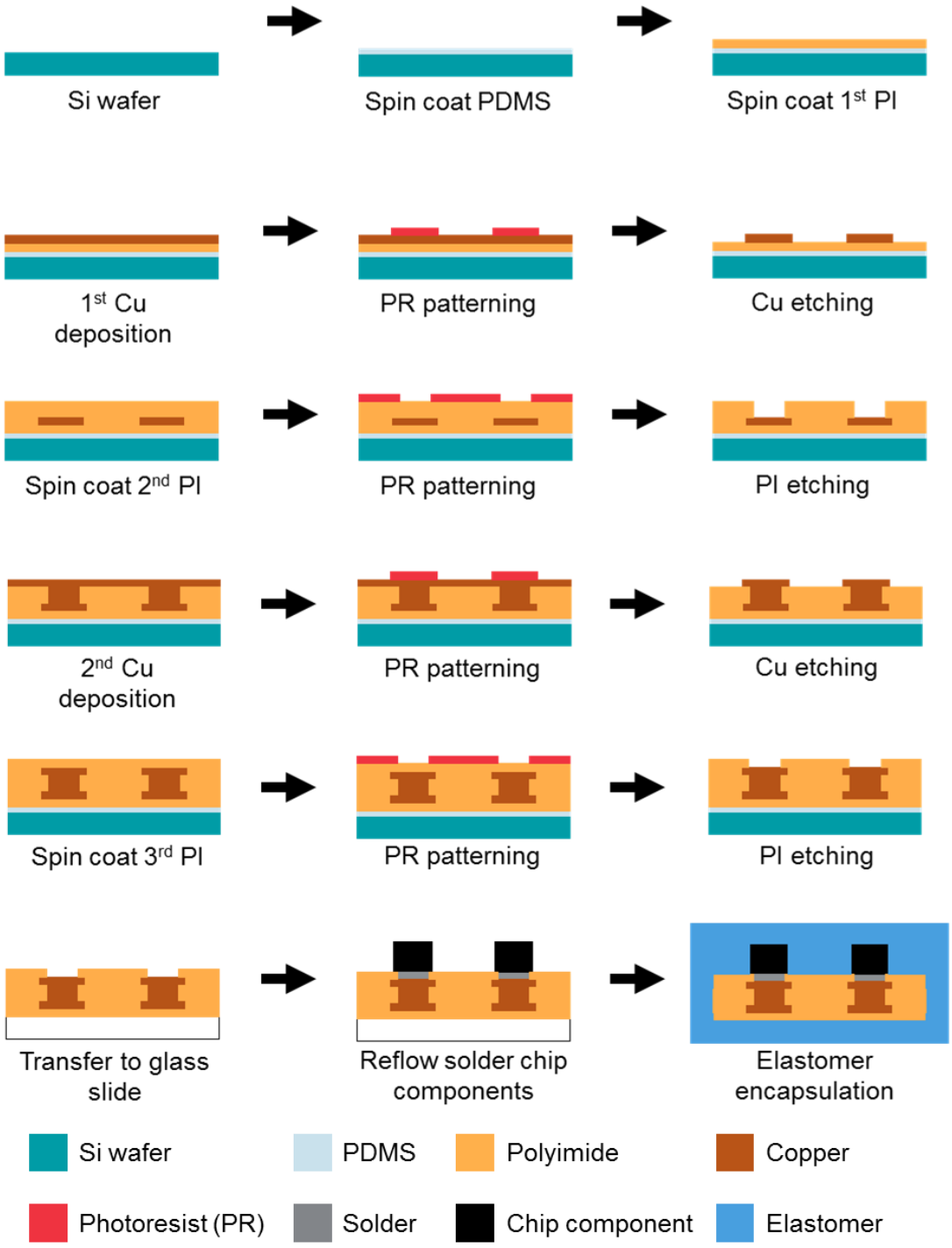


Figure S3. A schematic representation of the microfabrication and the assembly processes. Conventional microfabrication processes are used to construct the flexible circuit board. The PDMS-coated wafer surface allows the completed circuit boards to be peeled off and transferred onto a glass slide, where the reflow soldering of the chip components is carried out. The soldered board is detached from the glass slide and fully encapsulated with elastomer.

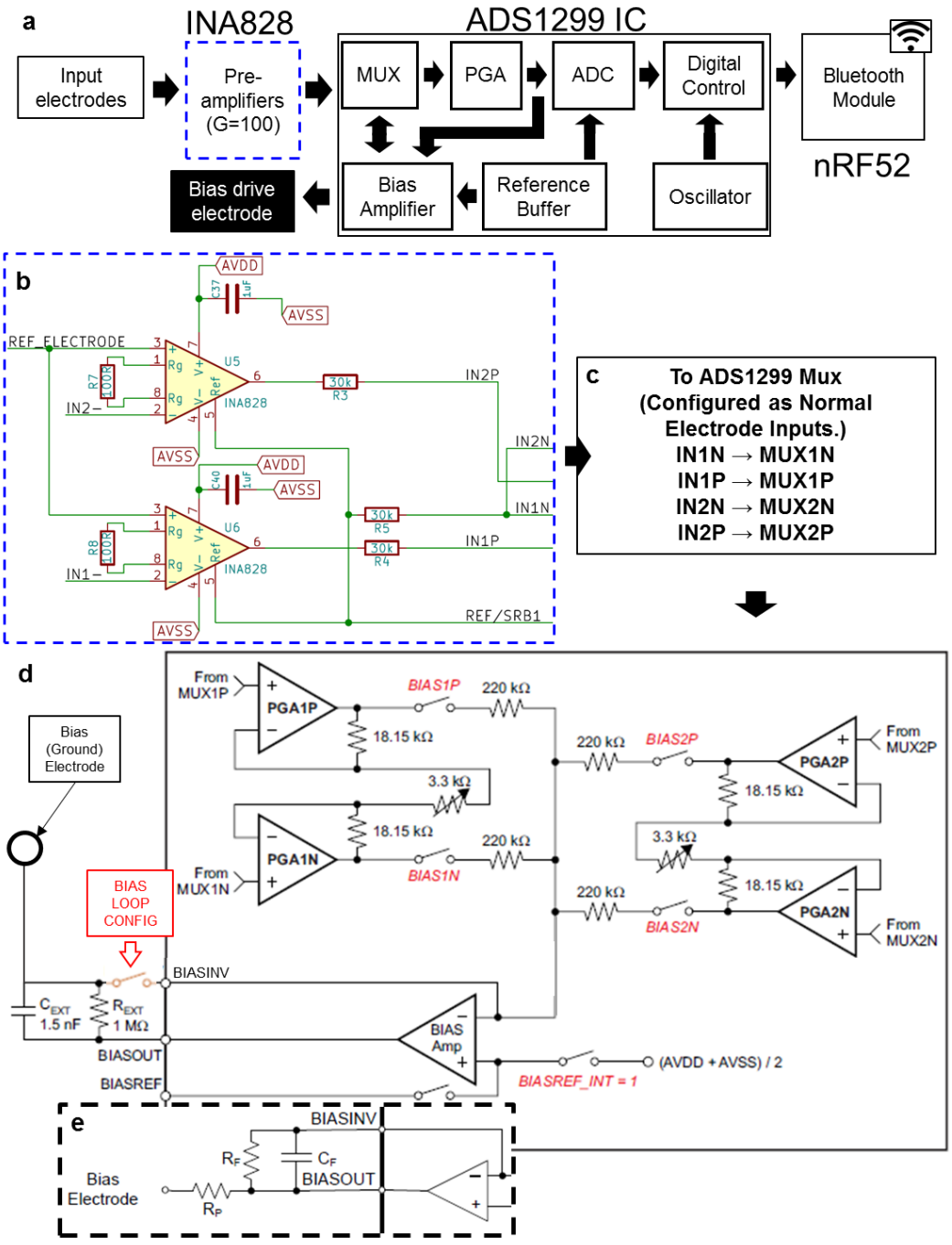


Figure S4. Block diagram and schematics of EEG amplifier configurations. (a) Block diagram of the overall flow of information from the measurement electrodes to the Bluetooth LE module. **(b)** A schematic of the pre-amplifier array configuration. **(c)** Internal routing of signals from pre-amplifier outputs **(b)** in ADC multiplexer. **(d)** A diagram of the driven bias (ground) in a conventional closed-loop configuration. **(e)** This diagram shows a switch to represent the open loop configuration of the bias amplifier, where the feedback loop is closed around the patient through the electrodes. Note: Schematic in panels **(d and e)** are adapted from TI ADS1299 datasheet, Rev C. (SBAS499C)[18]

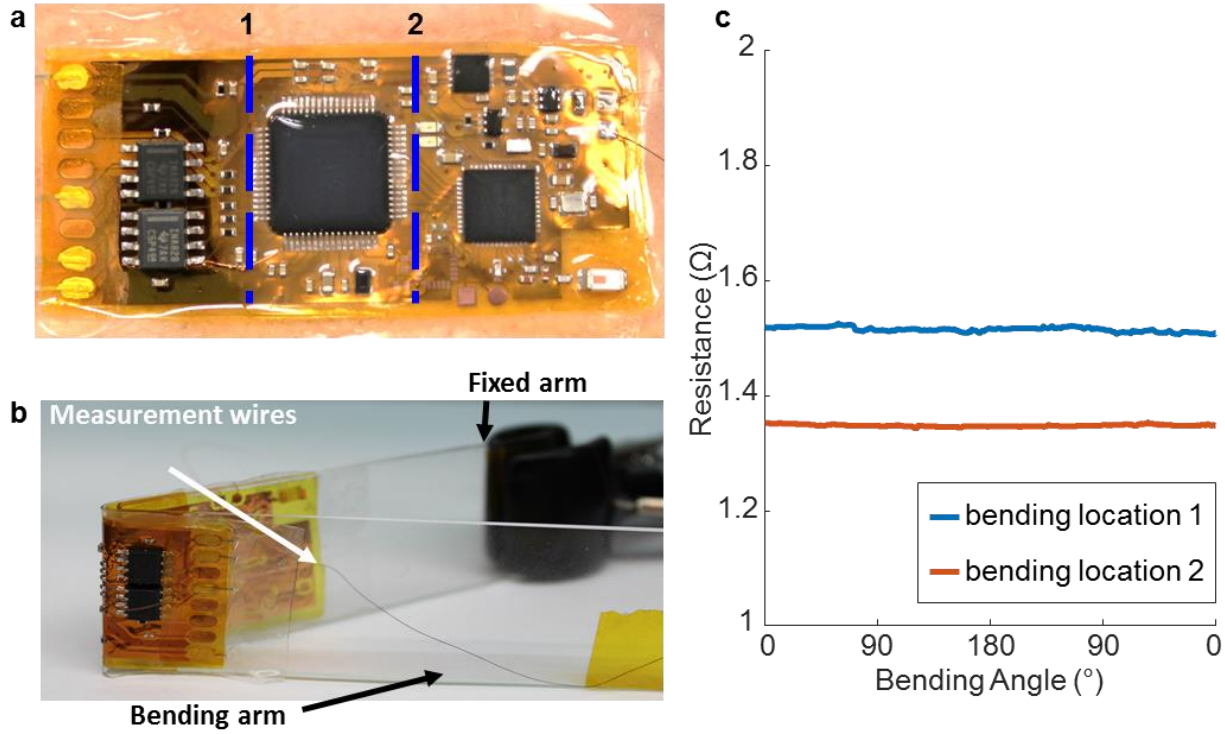


Figure S5. Mechanical Testing Setup and Additional Results. **(a)** Picture of device conformed to skin, with lines indicating the two locations of bending. **(b)** 180° bending test setup, with the device attached to glass slides, and attached wires for resistance measurements. **(c)** Resistance measurements on cyclic bending after 100 cycles completed at two bending locations, indicated by the legend.

Figure S6. Compression Test Setup for Elastomeric Hair-based Electrodes. (a) Motorized tension/compression test stand setup with benchtop digital multimeter and laptop to measure resistance. **(b)** Placement of elastomeric hair-based electrodes between compression plates of test device.

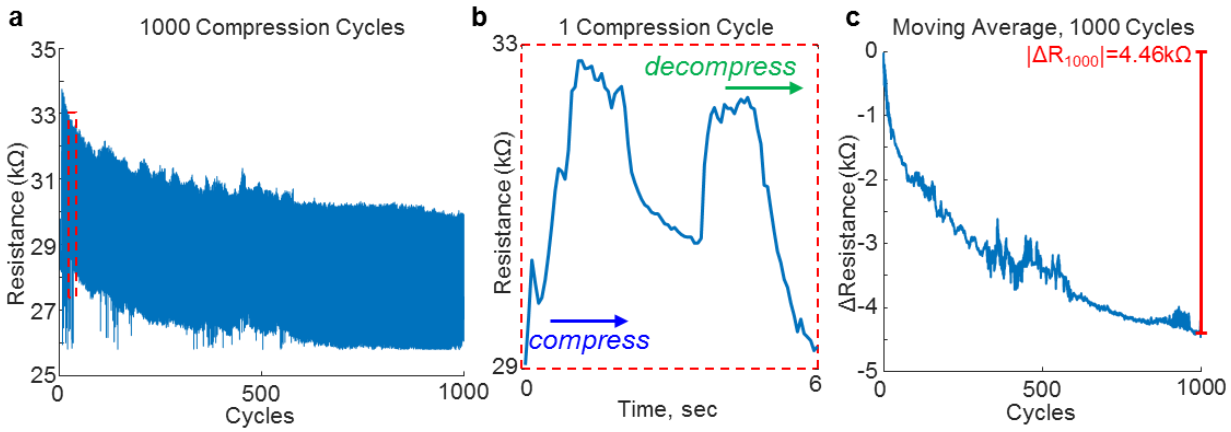


Figure S7. Data from Compression Test of Elastomeric Hair-based Electrodes. (a) Resistance data measured across single leg of the elastomeric hair-based electrodes, with **(b)** showing a representative resistance curve for a single compression cycle. **(c)** is a smoothed curve computed from a moving average of **(a)**, using a 6-second (90 sample) moving average, showing the drift in resistance over extensive mechanical usage.

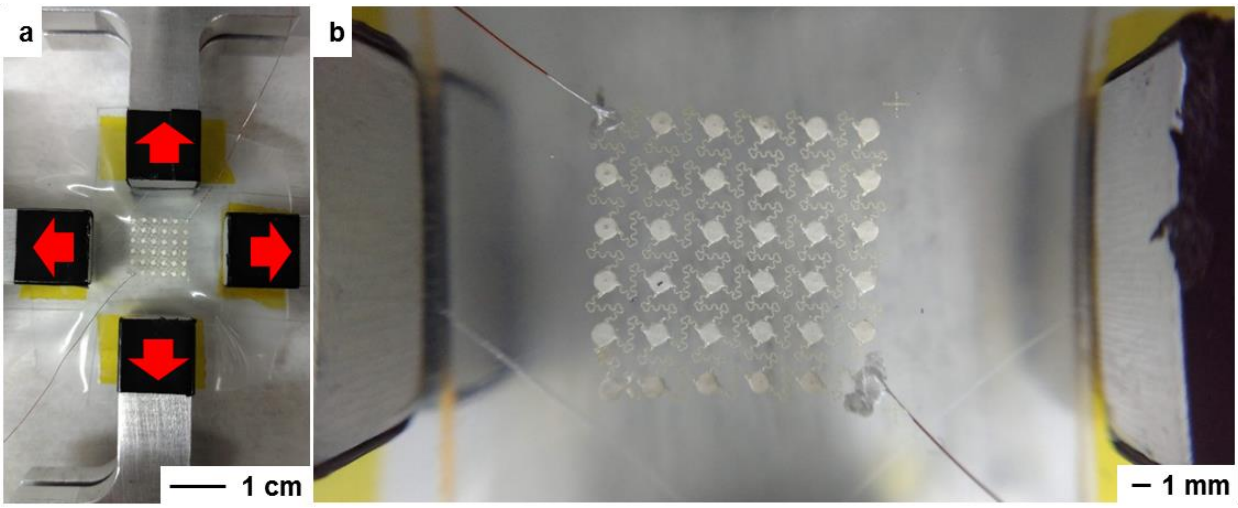


Figure S8. Biaxial Stretching Test Setup for Skin Electrodes. (a) Custom device for biaxial stretching with a flexible skin-based electrode on an elastomer placed between the 4 clamps. **(b)** shows a close-up of the device with attached copper wire (using silver paste) for resistance measurement.

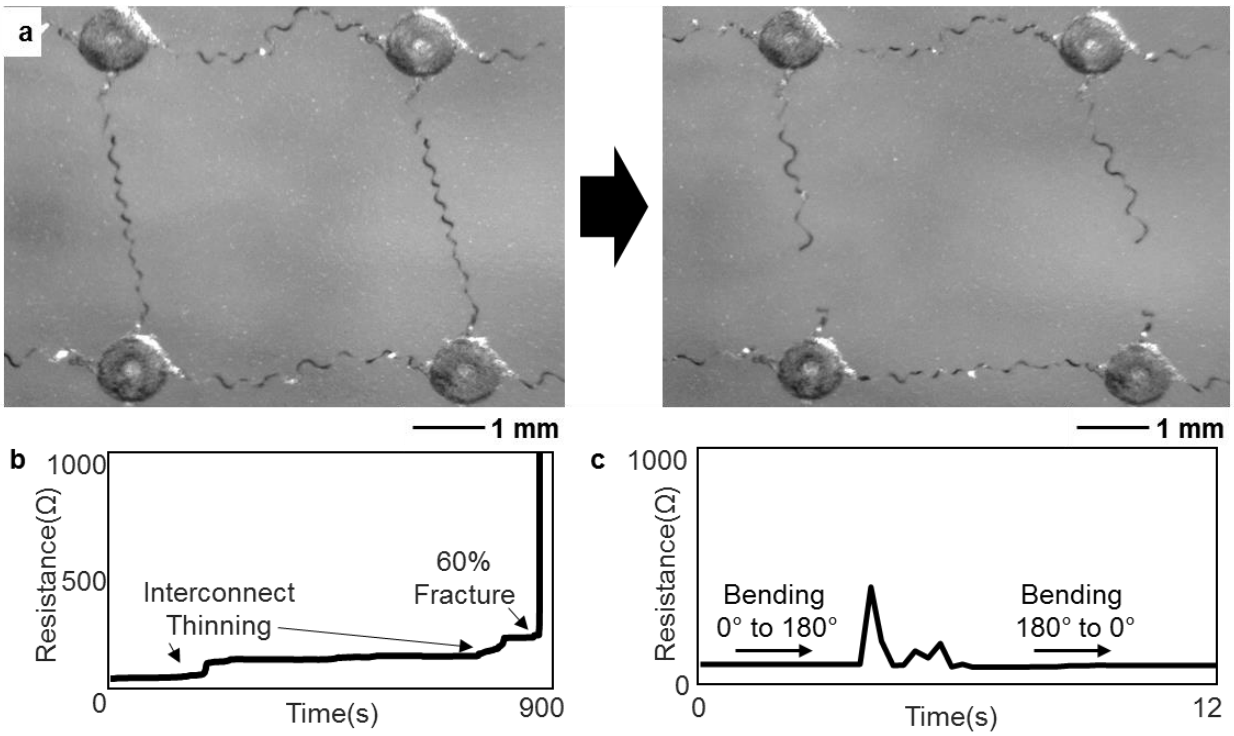


Figure S9. Data from Biaxial Stretching and Bending of Skin Electrodes. (a) electrode near 60% biaxial stretching before fracturing, as shown in **(b)**. **(c)** experimental stretching shows deformation of the silver causing an increase in resistance. **(d)** Bending 180° increases the resistance due to the deformation of the skin-electrode but releasing that stress brings the resistance back down.

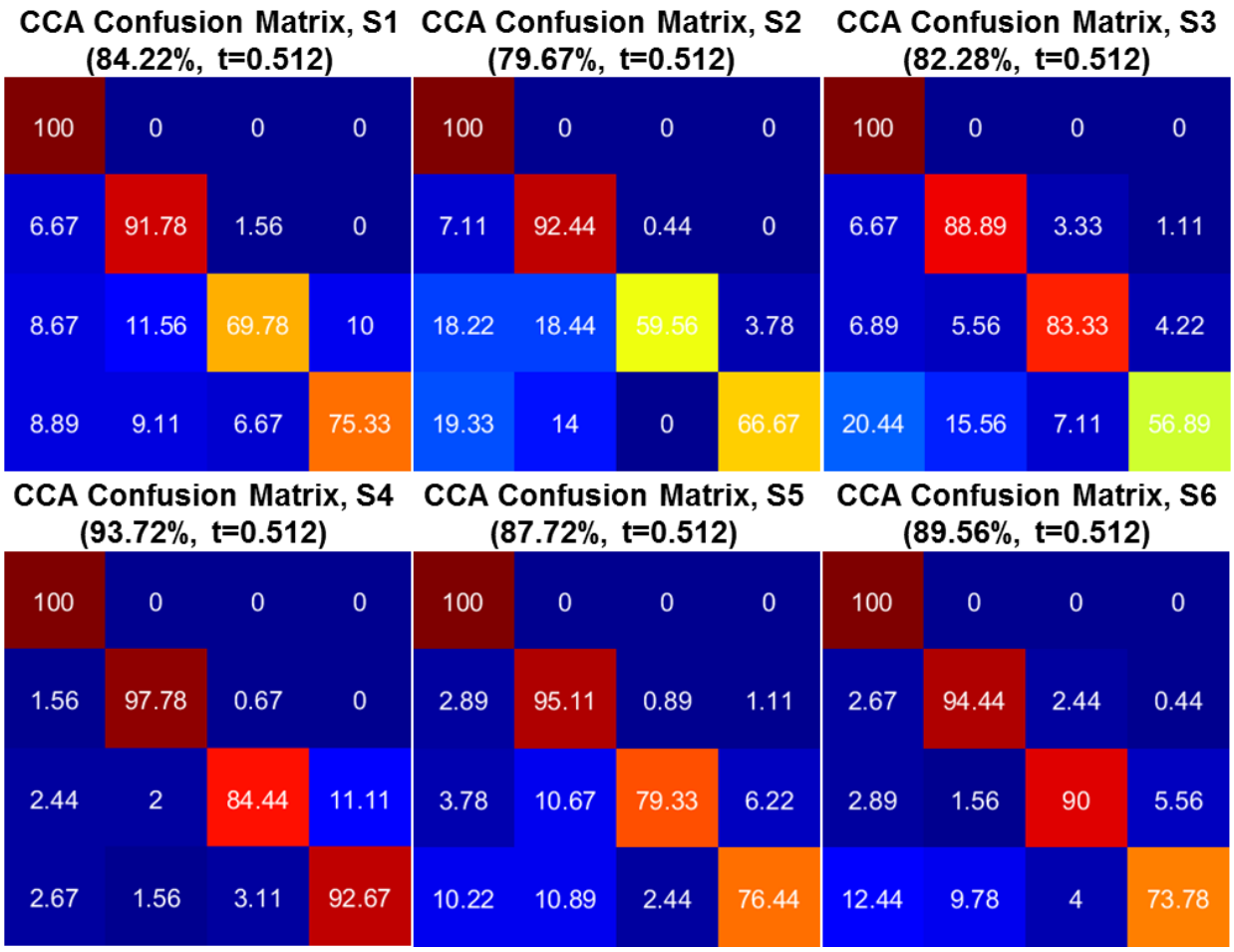


Figure S10. Accuracy Data from CCA of 2-channel SKINTRONICS SSVEP Dataset. Individual confusion matrices showing CCA results for each subject at data length of t=0.512s.

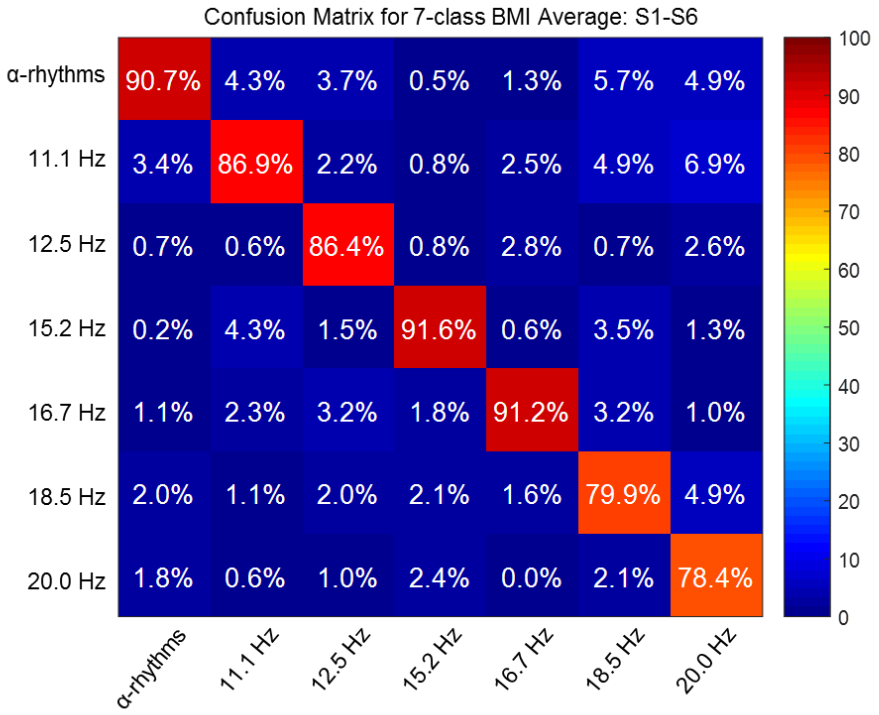


Figure S11. Confusion matrix for mean accuracy for 7-class SSVEP system for all subjects.

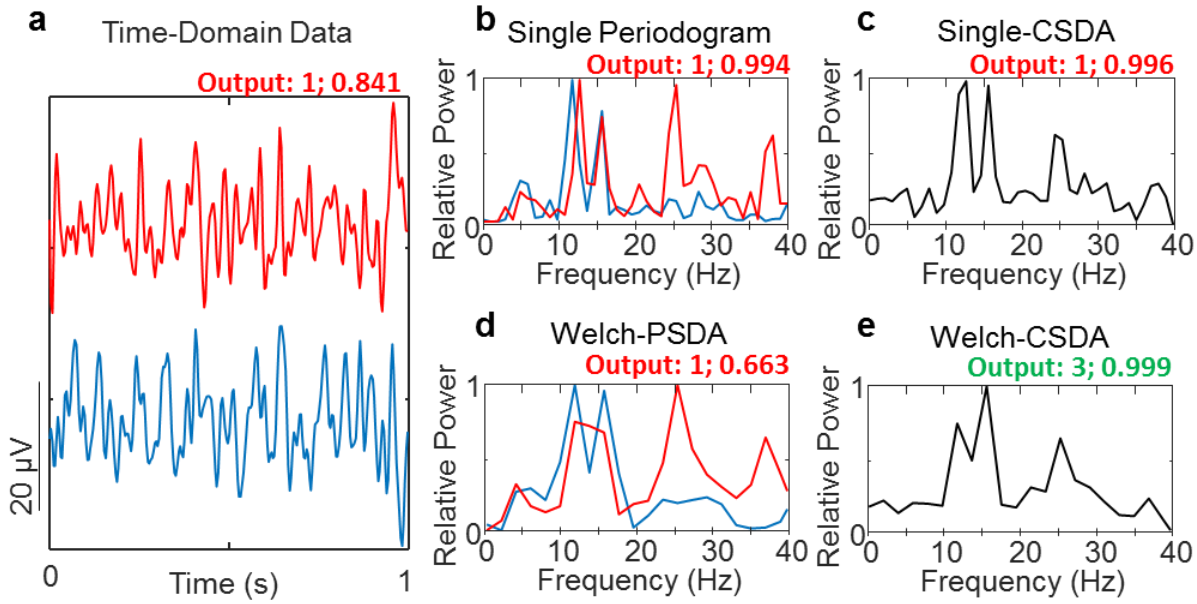


Figure S12. Feature representations using power and cross-spectral analyses. **(a)** A filtered EEG sample encoded with a noisy 15.2 Hz SSVEP signal in channels O1 and O2. **(b)** A simple periodogram using a window of equivalent length (single window periodogram). **(c)** The cross-spectral estimate of the single-windowed periodograms. This result may be improved by using windows of 0.5-second length with 50% overlap (Welch's method), as seen in **(d)**. Now, the 15.2 Hz peak is better represented, but still contains a peak around 11Hz. Taking the cross spectral estimate with Welch's method gives us **(e)**, which most closely represents the intended frequency content. In this example, the baseline CNN trained for classifying cross-spectral density data predicts the correct class. The labels show the predicted class and softmax output probabilities, where red represents a false classification output, and green represents the correct output.

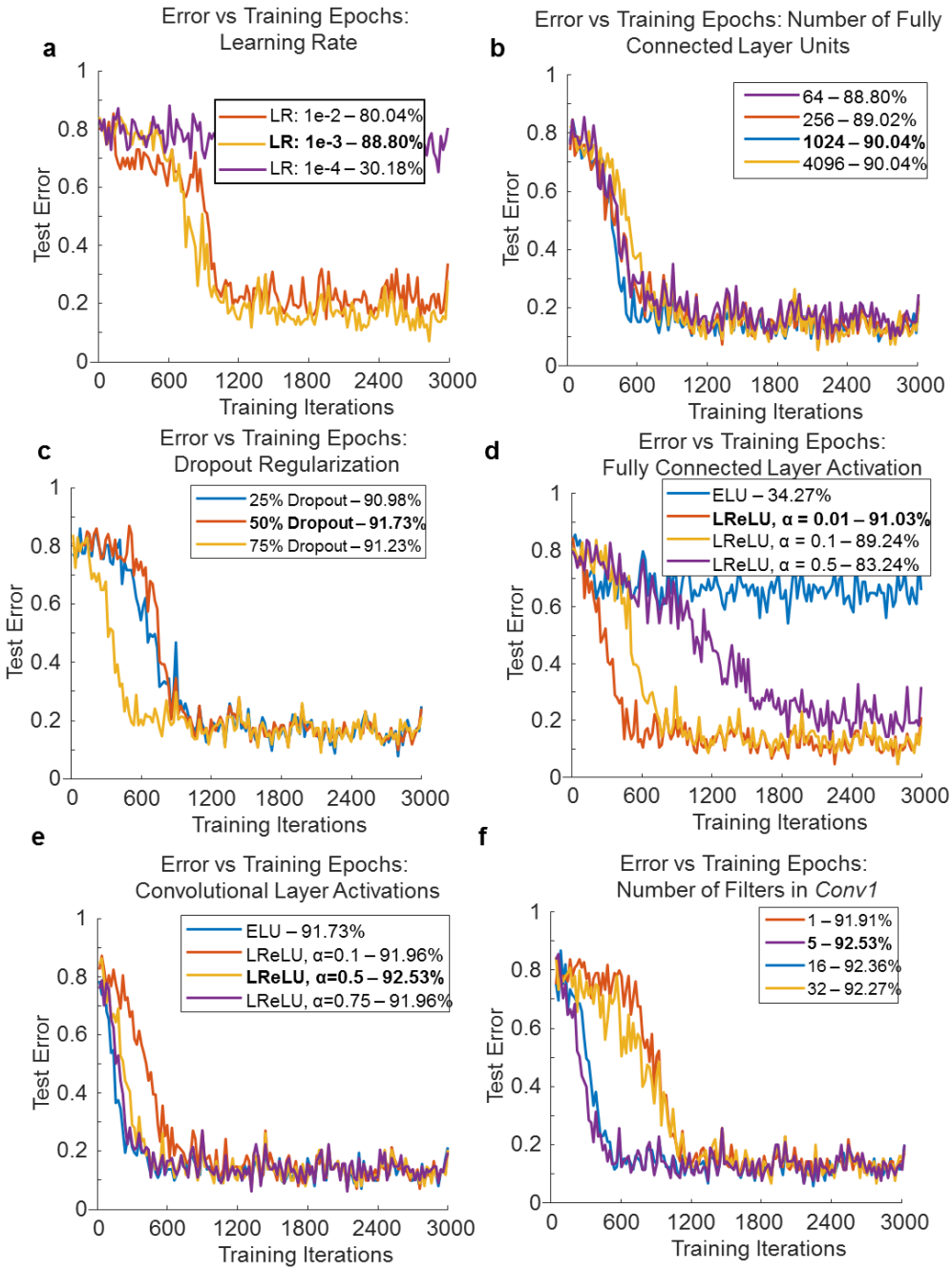


Figure S13. Hyperparameter Optimization of 1-Layer Convolutional Neural Network. Learning rates were initially tested on a single convolutional model with [2x2] convolution kernels and a stride of [1x1] for inputs of 0.512s of high-pass filtered time domain data. **(a)** Different learning rates between 1e-2 and 1e-4 were tested in, followed by number of units in the fully connected layer in **(b). (c)** Dropout regularization in between 25% and 75% is then added and analyzed. **(d)** Different activations in the fully connected layer are added. It appears that ReLU activation performs more optimally in the fully connected layer. **(e)** The activations following the convolution operation were varied, with LeakyReLU performing most optimally. **(f)** The number of filters was varied from 1 to 32, with 5 filters providing the most optimal performance with 92.53 ± 1.01%.

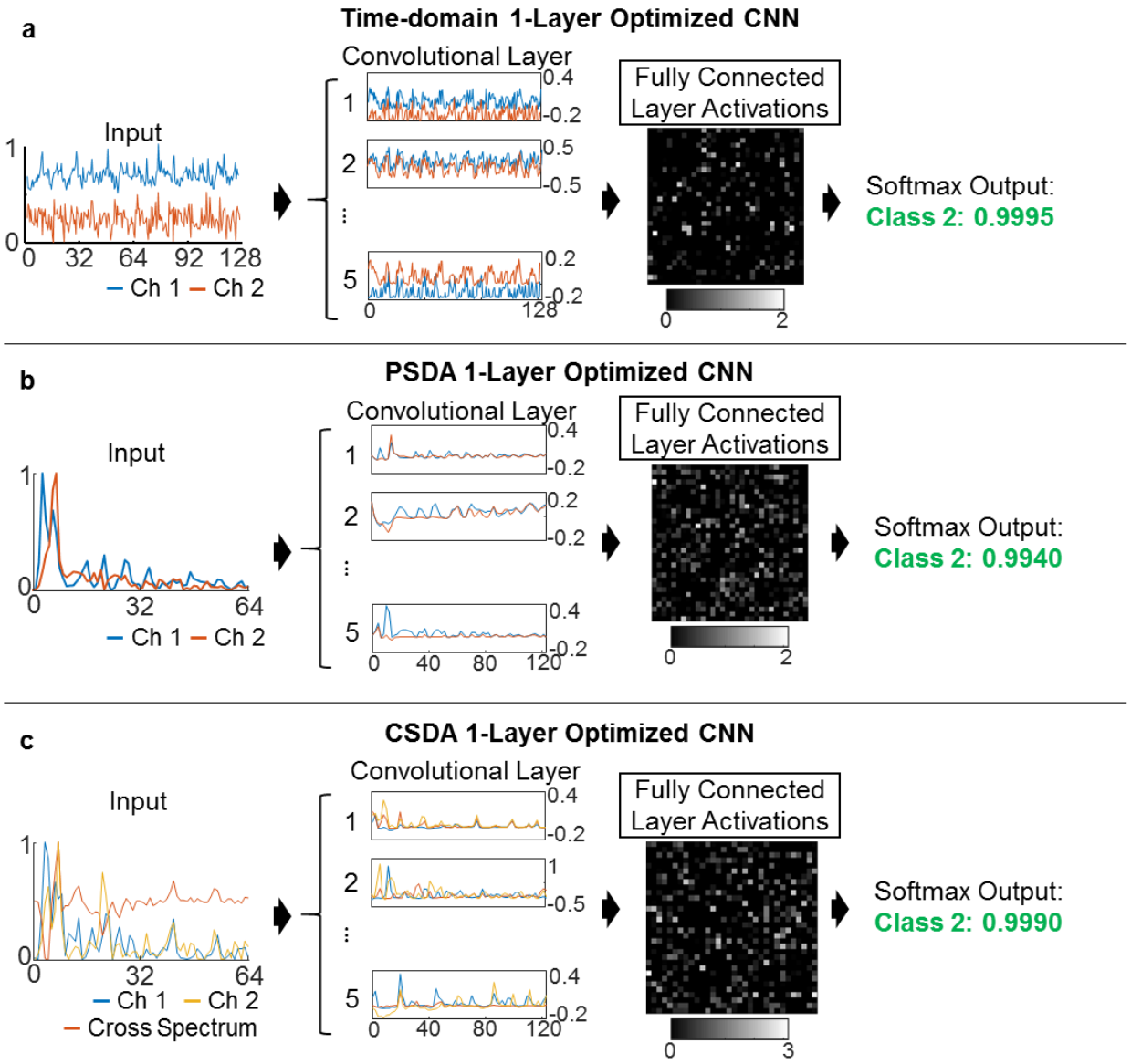


Figure S14. Convolution and activations from 1-layer optimized convolutional neural network. (a-c) shows the hidden layers for the time-domain model, PSDA model, and CSDA model respectively. The hidden layers illustrate the type of transformations that occur using convolution operations. The architectures for the models are shown in **Table 3**, for all inputs. For time-domain, [8 x 8] convolution kernels were found to be optimal, whereas for PSDA and CSDA [4 x 4] was found to be the most optimal. Note that the cross-spectral estimate includes the original power spectral estimate for each channel, so there are three inputs in total. The x-axis on the graphs represent data points.

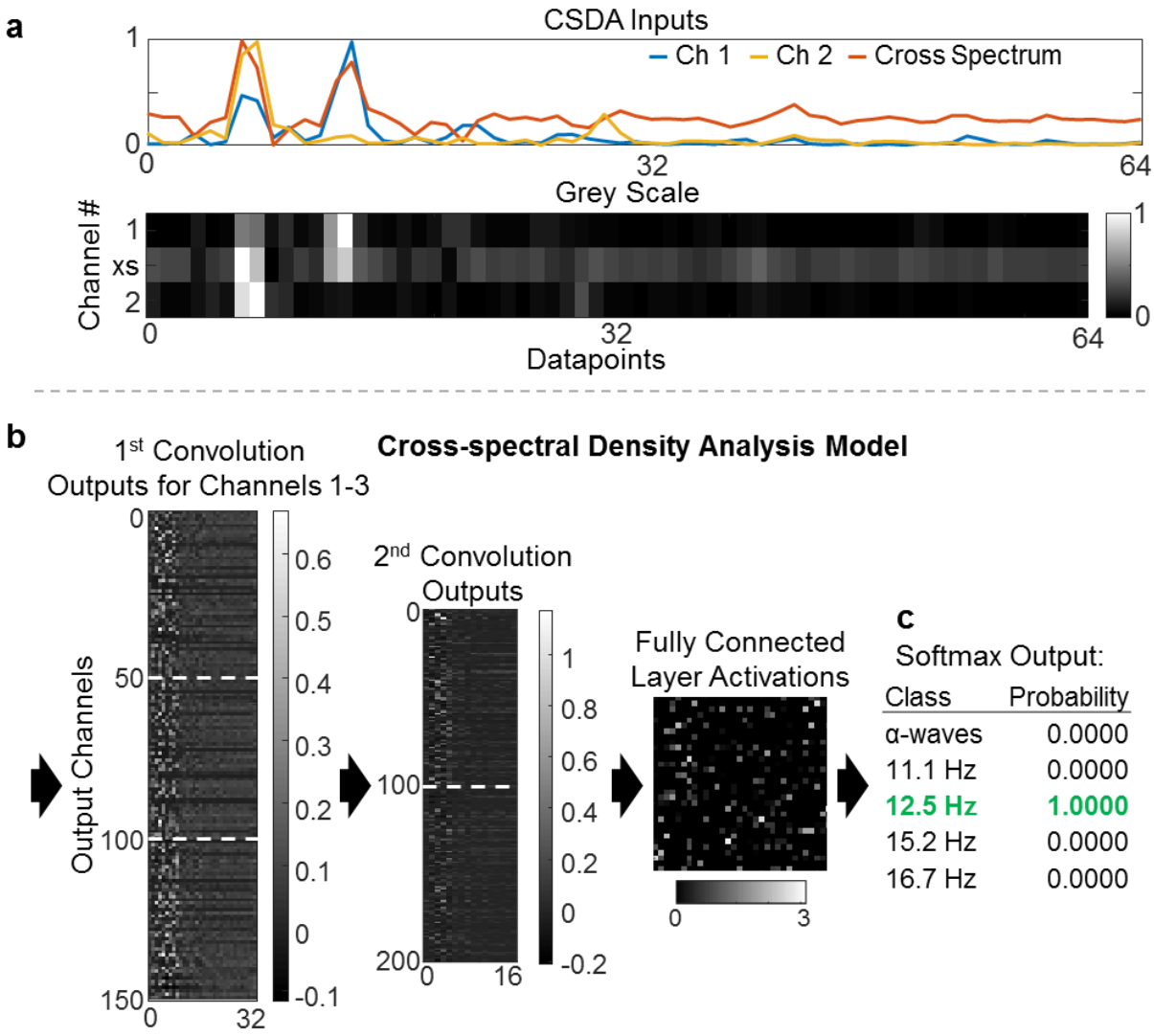


Figure S15. EEG Classification with Convolutional Neural Networks (Cross-spectral density analysis data). (a) Cross-spectral density input data, with greyscale images. Note that channel 2 is the cross-spectrum, represented by 'xs' on the greyscale image. **(b)** The hidden layers for the CSDA 2-CNN are shown, along with the fully connected layer activations, and **(c)** the softmax output. In **Fig. 5**, the analogous analysis for time domain data are shown.

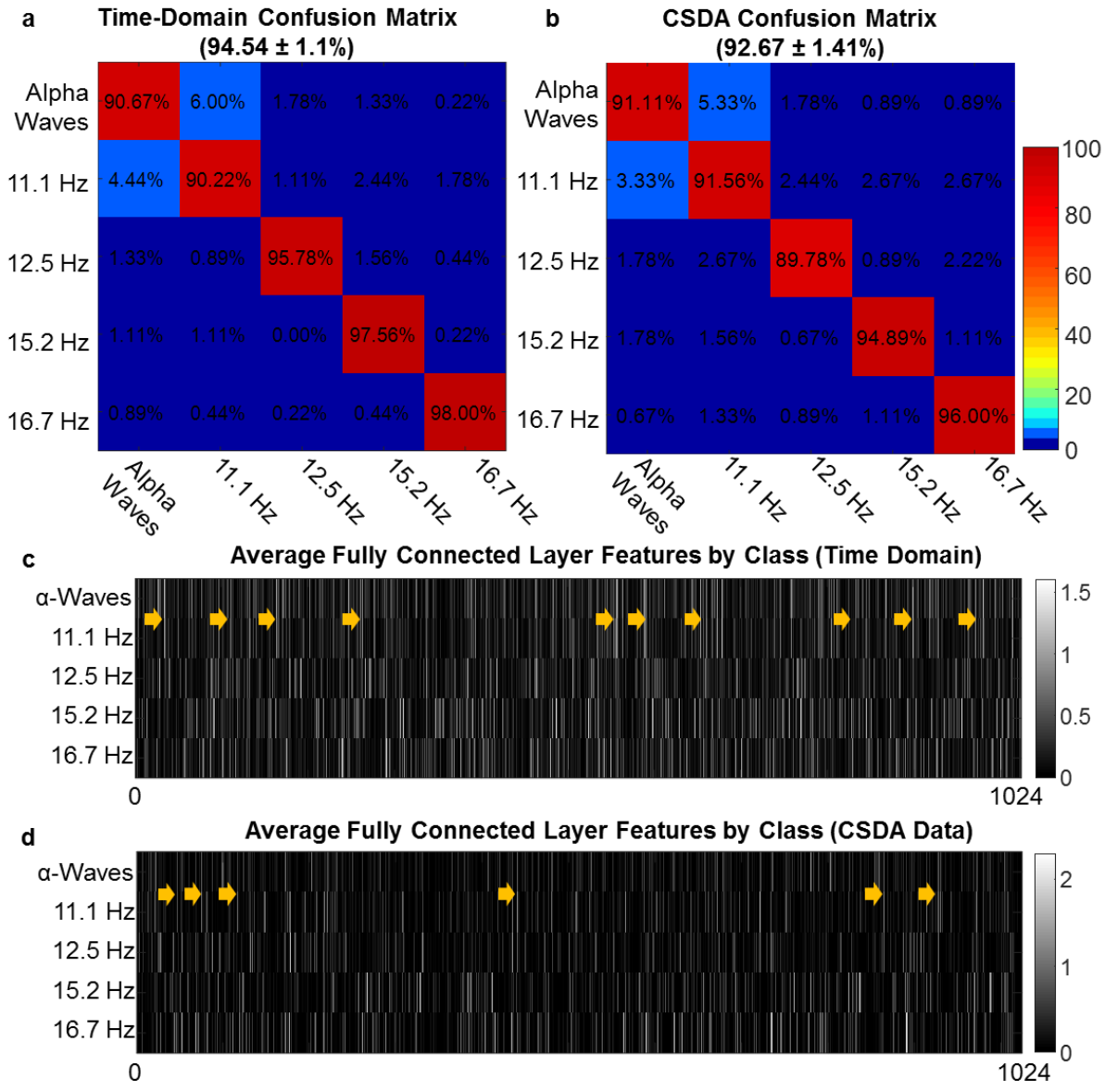


Figure S16. Confusion analysis of 2-layer convolutional neural network. **(a)** Confusion matrix for time-domain data, showing confusion between the first two classes. **(b)** Confusion matrix for cross-spectral density data inputs. **(c)** A greyscale map of average features output from the fully connected layer showing all 1024 units for each class (n=450), with arrows pointing out indices containing similar values for the first two classes. **(d)** a greyscale map of average features output from the CSDA 2-CNN model showing weaker feature maps for the classes of α-Waves and 11.1Hz, with some overlapping activations, indicated by the arrows.

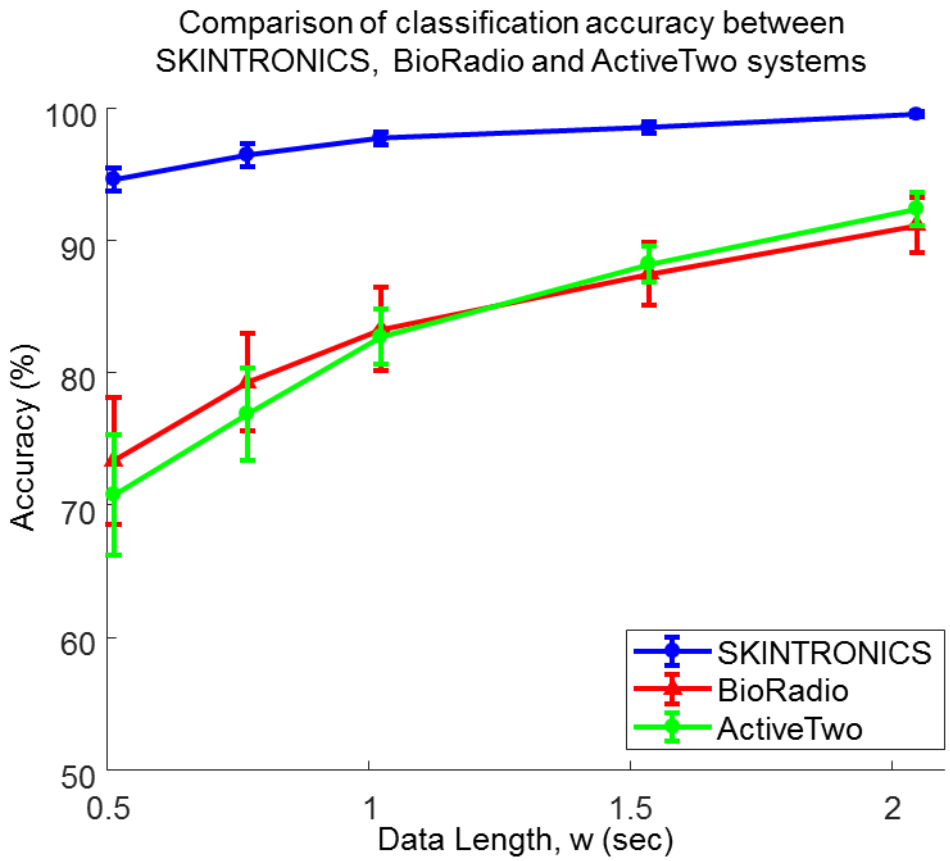


Figure S17. Comparison of classification accuracies of time domain data from SKINTRONICS, BioRadio and ActiveTwo systems using optimized CNN model.

Table S1. 2-CNN-Mapping: Convolutional Neural Network Model for Weight Extraction from 32-channel SSVEP Datasets. Using time domain data, we are able to optimize classification of real-time EEG data. Although the average accuracy was low across six subjects, we can generate mappings consistent with published literature regarding SSVEP.

Layer Name	**Layer Description**
Input	Time Domain Input: [32 x length(**w**)]
Conv1	[1 x 8] conv, 32 ReLU Units, Stride [1 x 1]
Pool1	[1 x 2] Max Pooling, Stride [1 x 2]
Conv2	[1 x 4] conv, 64 ReLU Units, Stride [1 x 1]
Pool2	[1 x 2] Max Pooling, Stride [1 x 2]
FC	Fully Connected Layer, 1024 ReLU Units, 50% Dropout
Softmax	5-way softmax output

Table S2. 1-CNN-Base: Baseline Classification Model. For time-domain, a smaller square filter was preferred over rectangular or vector filters. This model achieved an average accuracy of 88.89 ± 2.17%, when tested against all subjects with a window length of 128 (0.512 seconds).

Layer Name	**Layer Description**
Input	Time Domain Input: [2 x length(**w**)]
Conv1	[2 x 2] conv, 5 ReLU Units, Stride [1 x 1]
FC	Fully Connected Layer, 1024 ReLU Units, 50% Dropout
Softmax	5-way softmax output

Table S3. 1-CNN-Optimized: Optimized Time Domain Classification Model. For time-domain, a smaller square filter was preferred over rectangular or vector filters. Additionally, a larger nonlinearity was preferred with the parametric ReLU activations. This model achieved an average accuracy of **92.84 ± 0.92%**, when tested against all subjects with a window length of 128 (0.512 seconds).

Layer Name	**Layer Description**
Input	Time Domain Input: [2 x length(**w**)]
Conv1	[n x m] conv, 5 LeakyReLU Units ($\alpha = 0.5$), Stride [1 x 1]
FC	Fully Connected Layer, 1024 ReLU Units, 50% Dropout
Softmax	5-way softmax output

Table S4. Selected test accuracy test results for time-domain inputs with varied filter sizes. The model parameters are displayed in **Table 3**. Here we vary the filter dimensions.

Filter x dimension	Filter y dimension	Test Accuracy	Standard Error (n=6)
2	2	92.53%	1.01%
4	4	92.41%	0.97%
8	1	91.63%	0.92%
1	8	91.42%	0.69%
8	2	92.41%	1.21%
2	8	92.03%	1.33%
8	**8**	**92.84%**	**0.92%**
12	12	92.84%	1.51%

Table S5. Selected test accuracy test results with varied filter sizes using manually optimized single-layer convolutional neural network. The model parameters are displayed in **Table 3**. Here we vary the filter dimensions.

Filter x dimension	Filter y dimension	Test Accuracy	Standard Error (n=6)
2	2	91.98%	0.61%
4	**4**	**92.33%**	**1.05%**
8	1	91.11%	1.17%
1	8	91.02%	1.91%
8	2	92.04%	0.96%
2	8	92.00%	1.25%
8	8	90.76%	2.18%
12	12	90.71%	2.69%

Table S6. Classification Test Results Comparing CNN to SVM for Time Domain. Mean cross-validation (6-folds, one for each subject) test results for all subjects validated for each subject using 2-ch time domain inputs. Error represents standard error of the mean.

Data Length (sec)	**Time Domain** (n=6 Subjects)				
	2-CNN-TD	**Linear-SVM**	**Quad-SVM**	**Cubic-SVM**	**Gaussian-SVM**
0.512	**94.54%**	21.96%	82.67%	85.93%	81.95%
0.768	**96.40%**	22.79%	84.29%	86.69%	83.59%
1.024	**97.69%**	22.19%	86.54%	90.41%	87.34%
1.536	**98.50%**	22.59%	87.92%	90.96%	88.41%
2.048	**99.48%**	22.95%	84.86%	92.51%	93.55%

Table S7. Classification Test Results Comparing CNN to SVM for PSDA. Mean cross-validation (6-folds, one for each subject) test results for all subjects validated for each subject using 2-ch PSDA inputs. Error represents standard error of the mean.

Data Length (sec)	**PSDA**				
	2-CNN-PSDA	**Linear-SVM**	**Quad-SVM**	**Cubic-SVM**	**Gaussian-SVM**
0.512	**90.25%**	84.40%	86.11%	86.40%	86.25%
0.768	**94.89%**	91.73%	93.86%	94.50%	92.90%
1.024	**97.83%**	95.51%	97.32%	97.73%	96.80%
1.536	98.88%	98.72%	98.50%	**99.20%**	98.70%
2.048	99.52%	99.60%	99.71%	**99.71%**	99.29%

Table S8. Classification Test Results Comparing CNN to SVM for CSDA. Mean cross-validation (6-folds, one for each subject) test results for all subjects validated for each subject using 2-ch CSDA. Error represents standard error of the mean.

Data Length (sec)	**CSDA**				
	2-CNN-CSDA	**Linear-SVM**	**Quad-SVM**	**Cubic-SVM**	**Gaussian-SVM**
0.512	**92.58%**	86.93%	87.89%	90.11%	88.13%
0.768	**96.25%**	93.40%	93.61%	94.97%	95.63%
1.024	**98.41%**	94.51%	94.86%	98.16%	97.69%
1.536	**99.41%**	98.70%	99.10%	99.20%	98.95%
2.048	**99.95%**	99.59%	99.71%	99.71%	99.71%

Table S9. 2-CNN-TD: Time domain-optimized model with altered convolution stride path. The learning rate used was 1e-3, with a training batch size of 64. After training for 3000 steps, an average accuracy of **94.54 ± 1.10%** was achieved across all test subjects for 0.512 sec inputs in the time domain.

Layer Name	**Layer Description**
Input	Time Domain Input: [2 x length(**w**)]
Conv1	[8 x 8] conv, stride [1 x 2], 50 Leaky ReLU Units ($\alpha = 0.1$)
Conv2	[2 x 2] conv, stride [2 x 2], 100 Leaky ReLU Units ($\alpha = 0.2$)
FC	Fully Connected Layer, 1024 ReLU Units, 50% Dropout
Softmax	5-way softmax output

Table S10. 2-CNN-CSDA: CSDA-optimized model with altered convolution stride path. The learning rate used was 1e-3, with a training batch size of 64. After training for 3000 steps, an average accuracy of **92.58 ± 1.07%** was achieved across all test subjects for 0.512 sec inputs for cross-spectral density estimation data.

Layer Name	**Layer Description**
Input	Cross-Spectral Density Estimate: [3 x length(**w**)/2]
Conv1	[2 x 2] conv, stride [1 x 1], 20 Leaky ReLU Units ($\alpha = 0.1$)
Conv2	[2 x 2] conv, stride [2 x 2], 100 Leaky ReLU Units ($\alpha = 0.2$)
FC	Fully Connected Layer, 1024 ReLU Units, 50% Dropout
Softmax	5-way softmax output

Table S11. Classification Test Accuracy using Time Domain Data with Baseline Model. Test accuracies on data from test subjects 1-6, using unoptimized baseline model **(Table 2)**.

Data Length (sec)	**Classification Accuracy Per Subject**						**Average**	**Standard Error (±)**
	S1	**S2**	**S3**	**S4**	**S5**	**S6**		
0.512	94.00%	87.69%	80.68%	93.33%	90.89%	86.67%	**88.89%**	2.17%
0.768	92.27%	92.95%	82.49%	97.50%	93.18%	93.41%	**91.97%**	2.21%
1.024	98.37%	97.21%	88.14%	99.53%	97.44%	95.02%	**95.95%**	1.83%
1.536	99.76%	99.02%	93.41%	99.51%	98.29%	99.27%	**98.21%**	1.05%
2.048	98.97%	98.46%	93.59%	100.00%	98.97%	99.23%	**98.21%**	1.10%

Table S12. Classification Test Accuracy using Time Domain Data with Optimized Model. Test accuracies on data from test subjects 1-6, **using manually optimized model (Table 9)**

Data Length (sec)	**Classification Accuracy Per Subject**						**Average**	**Standard Error (±)**
	S1	**S2**	**S3**	**S4**	**S5**	**S6**		
0.512	95.33%	95.11%	90.09%	93.71%	96.44%	96.53%	**94.54%**	0.90%
0.768	97.73%	95.45%	92.73%	99.32%	96.59%	96.59%	**96.40%**	0.99%
1.024	98.84%	96.74%	97.00%	99.77%	96.28%	97.49%	**97.69%**	0.69%
1.536	99.27%	96.83%	98.05%	100.00%	97.56%	99.32%	**98.50%**	0.51%
2.048	99.74%	99.23%	98.46%	100.00%	99.74%	99.69%	**99.48%**	0.29%

Table S13. Classification Test Accuracy using CSDA Data with Baseline Model. Test accuracies on data from test subjects 1-6, **using unoptimized baseline model (Table 2)**.

Data Length (sec)	**Classification Accuracy Per Subject**						**Average**	**Standard Error (±)**
	S1	**S2**	**S3**	**S4**	**S5**	**S6**		
0.512	88.89%	90.22%	93.78%	94.62%	92.89%	91.50%	91.98%	1.12%
0.768	94.00%	95.68%	97.27%	98.41%	97.95%	95.02%	96.39%	0.89%
1.024	98.84%	97.91%	98.37%	99.30%	98.14%	96.79%	98.23%	0.56%
1.536	100.00%	99.76%	100.00%	99.76%	100.00%	98.54%	99.68%	0.05%
2.048	100.00%	100.00%	100.00%	100.00%	100.00%	99.74%	99.96%	0.00%

Table S14. Classification Test Accuracy using CSDA Data with Optimized Model. Test accuracies on data from test subjects 1-6, **using manually optimized model (Table 10)**

Data Length (sec)	**Classification Accuracy Per Subject**						**Average**	**Standard Error (±)**
	S1	**S2**	**S3**	**S4**	**S5**	**S6**		
0.512	89.11%	91.11%	93.11%	96.00%	94.22%	91.91%	**92.58%**	1.07%
0.768	95.45%	94.32%	97.50%	98.86%	96.82%	94.55%	**96.25%**	0.71%
1.024	98.60%	97.91%	98.84%	99.53%	98.60%	96.98%	**98.41%**	0.23%
1.536	99.51%	99.27%	100.00%	99.76%	100.00%	97.90%	**99.41%**	0.13%
2.048	100.00%	100.00%	100.00%	100.00%	100.00%	99.69%	**99.95%**	0.00%

Table S15. Classification Test Accuracy of using Time Domain Data from BioRadio with Optimized Model.

Data Length (sec)	Classification Accuracy Per Subject						Average	Standard Error (±)
	S1	S2	S3	S4	S5	S6		
0.512	74.95%	83.69%	56.88%	60.50%	90.10%	73.90%	73.34%	4.79%
0.768	80.54%	86.66%	64.24%	72.63%	91.64%	79.91%	79.27%	3.65%
1.024	86.66%	87.93%	69.71%	77.26%	93.36%	84.56%	83.25%	3.15%
1.536	91.54%	90.36%	76.66%	83.65%	94.44%	88.01%	87.44%	2.39%
2.048	93.69%	92.25%	80.81%	90.00%	96.67%	93.31%	91.12%	2.05%

Table S16. Classification Test Accuracy of using Time Domain Data from ActiveTwo system with Optimized Model.

Data Length (sec)	Classification Accuracy Per Subject						Average	Standard Error (±)
	S1	S2	S3	S4	S5	S6		
0.512	63.61%	79.63%	49.64%	73.72%	83.33%	74.54%	70.75%	4.59%
0.768	72.33%	83.51%	60.65%	79.93%	86.64%	78.15%	76.87%	3.47%
1.024	82.51%	86.63%	72.71%	81.55%	89.20%	83.51%	82.69%	2.10%
1.536	87.50%	90.15%	83.76%	85.09%	93.56%	88.91%	88.16%	1.32%
2.048	93.45%	94.16%	86.61%	89.95%	95.50%	94.41%	92.35%	1.26%

Table S17. Comparison of Cable Length between SKINTRONICS and Commercial Systems.

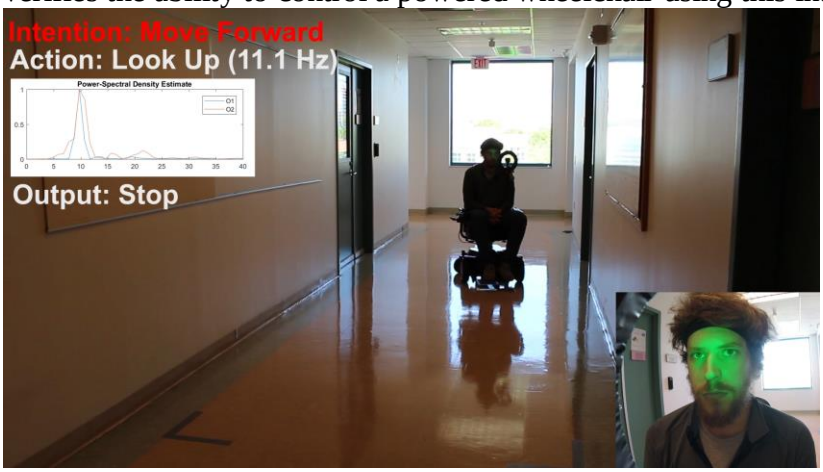
System Name	Cable Length	Electrode Type
ActiveTwo	140 cm	Active
BioRadio	100 cm	Passive
SKINTRONICS	18 cm	Passive

Table S18. Subject Information: Ethnicity and Hair Conditions.

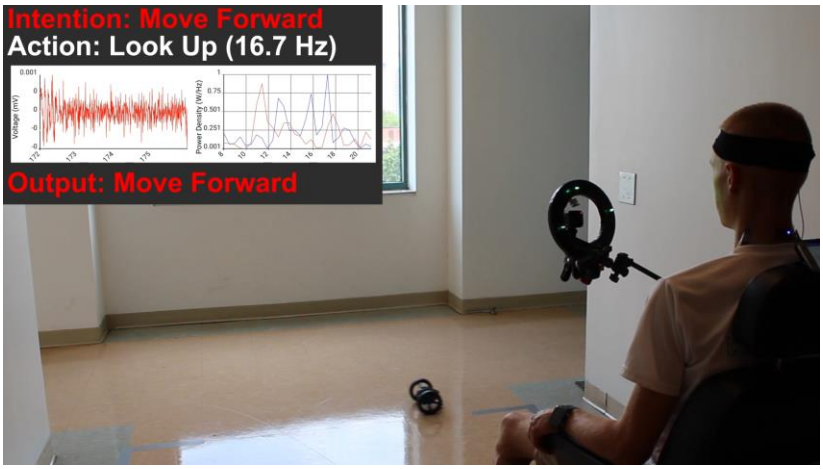
Subject Number	Ethnicity	Hair thickness	Hair Length	Mean SNR (dB)
S1	Caucasian	Sparse	Long	40.1±4.9
S2	Asian	Dense	Short	54.9±2.4
S3	Caucasian	Dense	Short	37.2±9.1
S4	Caucasian	Sparse	Very Short	60.4±2.2
S5	Caucasian	Dense	Short	64.2±1.5
S6	Asian	Dense	Short	60.3±3.9

Supplementary Videos

Video S1. Real-time demonstration of SSVEP-based control of a powered wheelchair. Video verifies the ability to control a powered wheelchair using this interface.



Video S2. Real-time demonstration of SSVEP-based control of a wireless vehicle. Video shows real-time control of a wireless RC vehicle.



Video S3. Real-time demonstration of SSVEP-based control of a PowerPoint presentation. Video shows real-time control and navigation of a PowerPoint presentation.