nature electronics

ARTICLES

https://doi.org/10.1038/s41928-020-00510-8

Check for updates

A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition

Ali Moin[1,5]✉, Andy Zhou[1,5], Abbas Rahimi[2], Alisha Menon[1], Simone Benatti[3], George Alexandrov[1], Senam Tamakloe[1], Jonathan Ting[1], Natasha Yamamoto[1], Yasser Khan[1], Fred Burghardt[1], Luca Benini[3,4], Ana C. Arias[1] and Jan M. Rabaey[1]✉

Wearable devices that monitor muscle activity based on surface electromyography could be of use in the development of hand gesture recognition applications. Such devices typically use machine-learning models, either locally or externally, for gesture classification. However, most devices with local processing cannot offer training and updating of the machine-learning model during use, resulting in suboptimal performance under practical conditions. Here we report a wearable surface electromyography biosensing system that is based on a screen-printed, conformal electrode array and has in-sensor adaptive learning capabilities. Our system implements a neuro-inspired hyperdimensional computing algorithm locally for real-time gesture classification, as well as model training and updating under variable conditions such as different arm positions and sensor replacement. The system can classify 13 hand gestures with 97.12% accuracy for two participants when training with a single trial per gesture. A high accuracy (92.87%) is preserved on expanding to 21 gestures, and accuracy is recovered by 9.5% by implementing model updates in response to varying conditions, without additional computation on an external device.

Wearable sensors are increasingly prevalent in health monitoring and human–machine interface applications as a result of improvements in their size and comfort[1–4]. Recent advances in flexible electronics have enabled the fabrication of wearable sensors that can mechanically bend and conform to non-planar and dynamic surfaces of the human body, allowing physiological signals of low bandwidth to be measured[5–10]. Hybrid systems that combine flexible sensors with rigid computational components on a separate substrate have also been developed[8,10]. Such systems are useful for applications that require local signal processing and miniaturized form factors.

To provide a real-time analysis of physiological signals, wearable biosensors can implement machine-learning models for signal processing. Local (in-sensor) processing of signals from biosensors has advantages over wirelessly streaming raw data to an external computational device, including reduced communication link bandwidth and radio power requirements. Processing the signals locally can also offer improved latency and security. Machine learning models for in-sensor processing are, however, typically trained offline before they are implemented in low-power embedded processors[11,12].

Modern, lightweight machine-learning algorithms perform well when the training data align with the conditions that are expected during deployment, but, when the initial training of a classification model fails to capture a wide set of conditions, the classification accuracy of the model degrades, resulting in suboptimal performance or poor user experience[13–18] (Supplementary Fig. 5). Therefore, the ability to train and update an in-sensor classification model during practical application is desirable.

Gesture recognition using surface electromyography (sEMG)[19–24] could benefit from a hybrid system with in-sensor processing. sEMG-based gesture recognition devices can measure electrical muscle activity from electrodes placed on the surface of the skin and perform pattern recognition on features that are extracted from these signals[11,25]. These devices should ideally have a small standalone form factor for autonomy, a high channel count and high-density electrode placement for more comprehensive spatial coverage and improved classification accuracy[26,27], and in-sensor intelligence that is generalizable or adaptable to various wear conditions. Existing devices are either in non-wearable form factors[26], dependent on external devices for computation[28] or reliant on a small number of precisely positioned, bulky electrodes[12,29,30]. Furthermore, the systems that are capable of in-sensor (online) training do not support in-sensor model updates[29,30]. These systems thus cannot adapt to signal variations from sweating, fatigue, varying muscle contraction effort, and electrode displacement due to changing situational contexts such as limb and body position or device doffing and donning[13–18].

In this Article, we report a wearable, high-density sEMG biosensing system (Fig. 1a) that uses hyperdimensional (HD) computing to implement in-sensor adaptive learning and real-time inference for hand gesture classification. HD computing is an emerging computing paradigm that supports fast and simple learning and is inherently robust against noise and errors[31]. HD computing takes advantage of information being represented by very high-dimensional vectors (hypervectors) to perform otherwise complex tasks, such as classification or reasoning, using simple computational operations[31]. This approach has already shown promising results in classification tasks

[1]Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA, USA. [2]IBM Research Zurich, Ruschlikon, Switzerland. [3]Department of Electrical, Electronic and Information Engineering, University of Bologna, Bologna, Italy. [4]Department of Information Technology and Electrical Engineering, ETH Zürich, Zurich, Switzerland. [5]These authors contributed equally: Ali Moin, Andy Zhou. ✉e-mail: moin@berkeley.edu; jan_rabaey@berkeley.edu

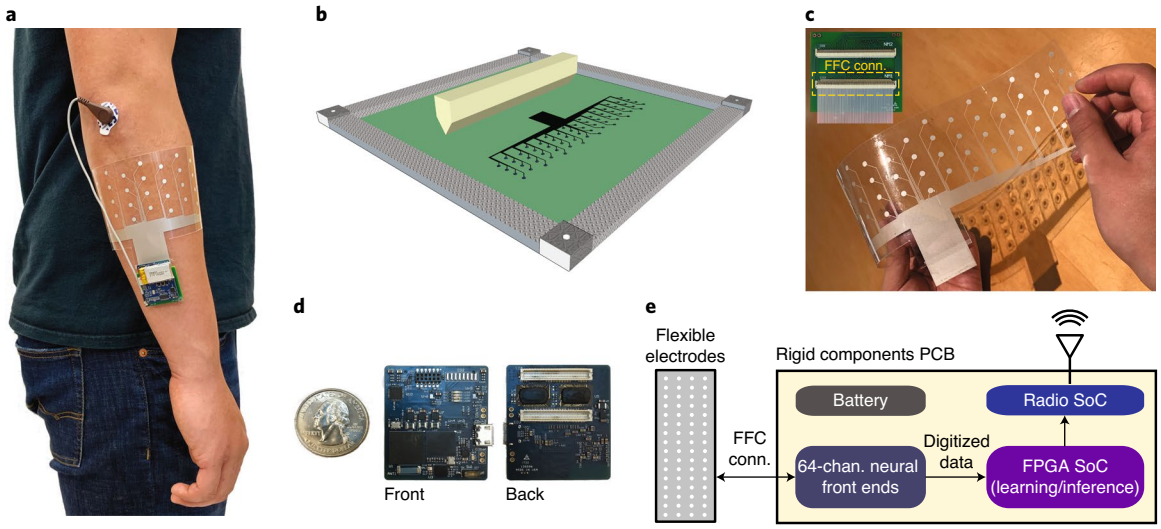

Fig. 1 | Wearable biosensing system for sEMG. a, The device on the forearm of a participant. **b**, Illustration of the screen-printing process. A squeegee and a screen are used to print sEMG electrode arrays onto flexible substrates. The electrode pattern is defined by features on the screen. **c**, The custom-designed, flexible 16 × 4 array of electrodes that conforms to the forearm to provide high-density, large-area sEMG recordings without individual wires. An adapter PCB with a flat flexible connector (FFC conn.) is used to interface the flexible electrode array with the rigid components. **d**, The miniaturized, eight-layer PCB that accommodates the complex, rigid components responsible for sensing, processing and telemetry. **e**, Block diagram of the main components constituting the wearable system.

for physiological signals such as sEMG[32], electroencephalography (EEG)[33] and electrocorticography (ECoG)[34], discriminating up to five classes on offline datasets.

To build our system, we used a hybrid method for interfacing soft conformal sensors and hard silicon-based integrated circuits (ICs). In particular, we combined a sensing electrode array printed on a flexible substrate (Fig. 1b,c) with a miniaturized printed circuit board (PCB) that includes complex sensing, processing and telemetry components (Fig. 1d,e). The large-area, high-density electrode array has 64 channels and was fabricated by screen printing using conductive and dielectric inks on a thin polyethylene terephthalate (PET) sheet. The PET sheet is highly flexible and conforms well to the complex three-dimensional (3D) form of forearm muscles during contractions and relaxations. The system uses a low-power custom application-specific integrated circuit (ASIC) to collect the sEMG data for classification. The custom-designed ASIC[35] can record and digitize the sEMG signals from multiple channels close to the electrode sites, reducing the need for bulky cable connections between electrodes and signal conditioning circuitry. We validated our biosensing device through a series of real-time experiments that model signal variations from real-world use cases. We obtained a hand gesture classification accuracy of 97.12% when classifying 13 hand gestures using only a 4 s window of training data per gesture. We demonstrated that the classification model can be updated locally to classify a larger set of hand gestures, as well as to recover from accuracy degradation due to theparticipant moving to a new arm position, replacing the device or wearing the device for a prolonged period of time.

Hybrid flexible biosensor design and fabrication

To create the flexible skin-worn interface, we screen-printed a uniform 4 × 16 array of circular electrodes (diameter r, 4.3 mm) and connective traces on a flexible PET substrate using conductive silver ink (NovaCentrix FG57b) sintered by photonic curing (Fig. 1b,c)[36]. The silver ink formulation included binders to help with cohesion of the silver microflakes as well as adhesion to the substrate. Compared to high-density electrode array fabrication methods reported in the literature[28,32,37,38], our screen-printing solution with photonic sintering allows the use of temperature-sensitive substrate materials and requires only subsecond curing time for cheaper and faster large-scale production. A dielectric encapsulation layer (NovaCentrix DE-SP1) was subsequently printed with via holes for exposing the electrode pads while insulating the conductive traces from the skin. Each printed layer was 15 µm thick. Overall dimensions (29.3 cm × 8.2 cm) were chosen to wrap around the entire circumference of an above-average-sized forearm, capturing activity of the extrinsic flexor and extensor muscles involved in finger movements with low inter-electrode pitch in both the proximal–distal and medial–lateral directions. Four electrodes per column were spaced 14.3 mm apart, and the columns were spaced 17.8 mm apart with vertical offsets of 7.15 mm.

The flexible electrode array was interfaced to a custom, eight-layer PCB (Fig. 1d,e) using a flat flexible connector (FFC) on a two-layer adapter board (Fig. 1c). All 64 channels from the electrode array were sampled and digitized at 1 kS s^{-1} using a custom neural interface IC[35] with very low power consumption (700 µW) in a small form factor. sEMG signals were processed within a system-on-a-chip field-programmable gate array (SoC FPGA), where we implemented a full training and classification algorithm to make the system entirely standalone. The device connected wirelessly to a base station computer using a 2.4 GHz radio SoC for reconfiguration and data logging, if needed. The board was powered using a single 3.7 V, 240 mAh lithium-ion battery weighing 4 g. The total weight of the wearable system was 26 g, with the computational unit PCB weighing 6 g and the electrode array and adapter weighing 16 g. In this configuration, battery life during continuous gesture recognition was ~6 h, with power dissipation being dominated by raw data packetization and wireless transmission at greater than 1 Mbps (Supplementary Fig. 7).

On-body characterization of the electromyography acquisition

Using our device, we collected and analysed an offline sEMG dataset from five participants across multiple wear conditions simulating

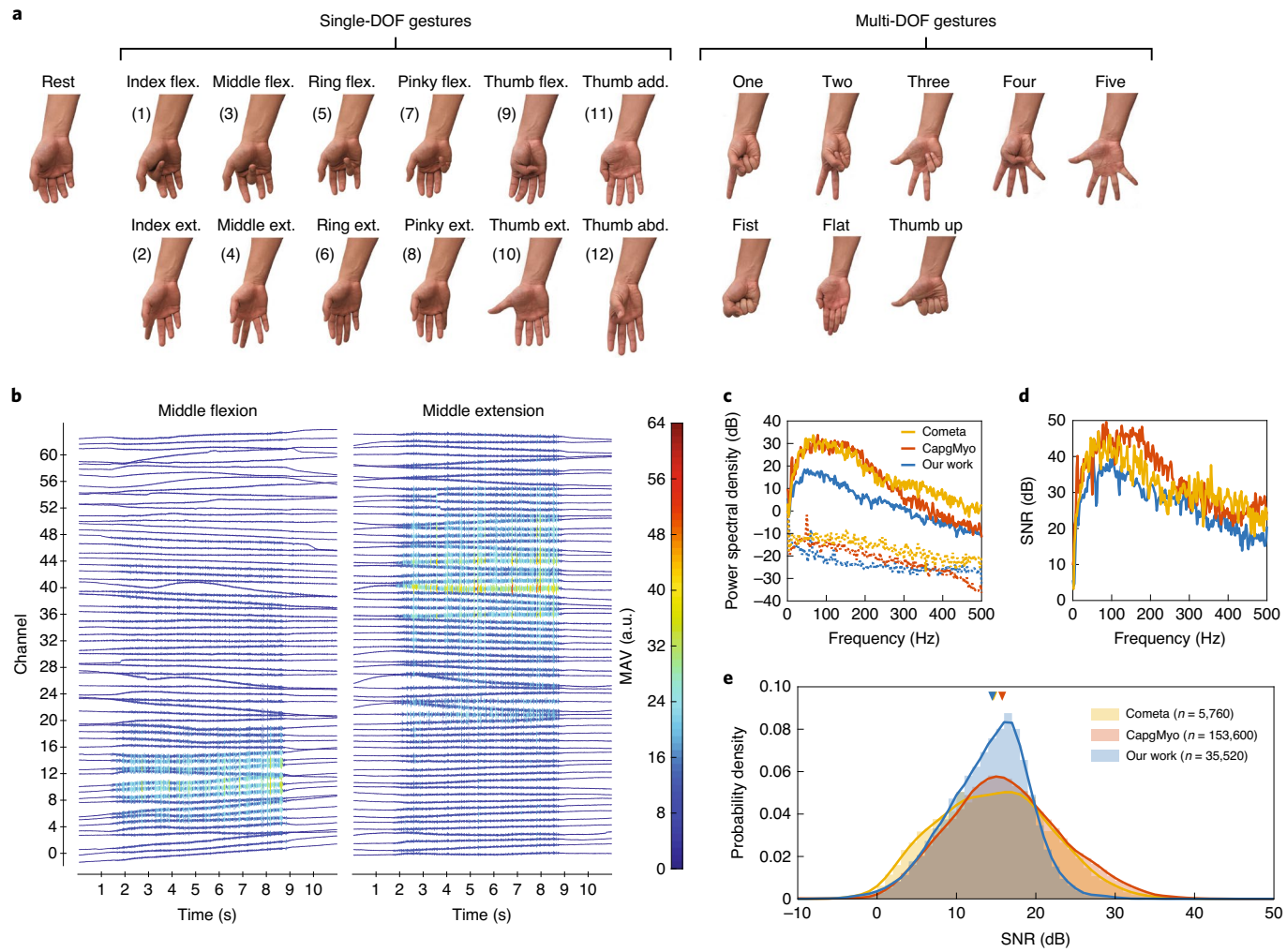

Fig. 2 | Hand gesture classes used in the study and sEMG recording characteristics. a, The single-DOF gesture subset includes individual finger flexions (flex.) and extensions (ext.) along with the 'rest' gesture. The multi-DOF gesture subset includes common, isometric hand postures involving multiple fingers. add., adduction; abd., abduction. **b**, Example raw waveforms recorded from all 64 channels during middle finger flexion and extension. Channels are organized starting with channel 0 on the surface nearest the radius, with increasing channels wrapping around the anterior side of the forearm to the surface nearest the ulna (~channel 32), and then around the posterior side back to the radius. Each 11 s gesture trial is divided into 1.5 s rest, a 2 s transition period to the gesture, a 4 s hold period, a 2 s transition period back to rest and 1.5 s rest based on the instructions given to the participant. The colour of the waveform indicates the local amplitude of the sEMG, as measured by MAV calculated over 50 ms segments. **c**, Example Welch's power spectral density estimates of sEMG recordings from single channels of three systems, including a commercial sEMG interface with individually placed bipolar electrodes (Cometa + Dormo, NinaPro DB4[39–41]), a benchtop high-density sEMG acquisition set-up (CapgMyo[42]) and our system. Solid lines represent the signal spectrum during performance of the middle extension gesture, while dotted lines represent the signal spectrum during rest. Spectral densities are normalized to the 0 Hz component of the rest spectrum for each recording system, for ease of comparison. **d**, Spot SNR for the example performances in **c**, measured as the ratio between the middle flexion spectrum and the rest spectrum. **e**, Distribution of overall SNR for the three compared systems for all channels during the performance of all single-DOF gestures. Shaded areas are the probability density histograms of SNR values. Lines are a Gaussian kernel fit to the distribution. Downward pointing triangles represent the medians.

typical use cases (Fig. 2 and Supplementary Note 1). The dataset included flexion and extension of different finger degrees of freedom (DOF; Fig. 2a), with single-degree-of-freedom (single-DOF) gestures being performed in four different situational contexts and multiple-degree-of-freedom (multi-DOF) gestures being performed in only the baseline situational context (Supplementary Note 1). The device was worn by each participant as shown in Fig. 1a, with the centre of the array roughly aligned to the ulna of their dominant arm (Methods). A single drop of conductive gel was placed on each electrode to improve skin contact and adhesion. Signals were recorded in a single-ended configuration, with an ECG Ag/AgCl electrode on the elbow used as reference. Figure 2b shows example waveforms recorded from all 64 channels during the flexion and extension of the middle finger DOF. Each 50 ms segment of each waveform is coloured based on the mean absolute value (MAV) feature derived from that segment, indicating the local amplitude of the sEMG. Activity in contrasting subsets of channels (overlaying the anterior flexor muscles or posterior extensor muscles) can be seen in the antagonistic movements.

Signal quality, as acquired using our wearable biosensing system, was compared to available recordings of the same gestures using a traditional commercial sEMG interface (Cometa used in NinaPro DB4)[39–41] as well as a benchtop high-density sEMG acquisition set-up (CapgMyo)[42]. To compare the power spectrum of the recordings, we computed the Welch's power spectral density estimate for the channel with the highest signal-to-noise ratio (SNR) while performing

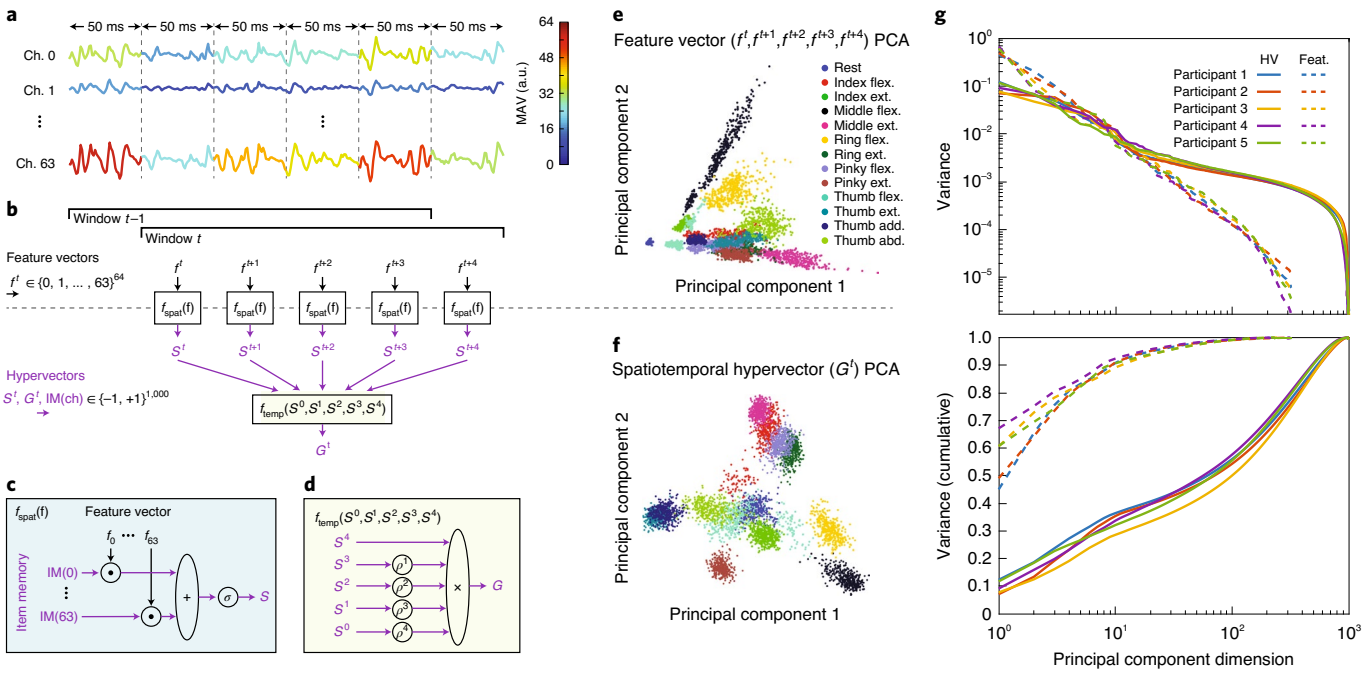

Fig. 3 | Hyperdimensional computing algorithm for projecting windows of sEMG data into hypervectors. a, Example sEMG data and MAV feature values. The three example traces shown are divided into 50 ms feature segments, with segments colour-coded based on MAV feature values. **b**, A single classification window at time t consists of five feature segments totalling 250 ms, and consecutive classification windows overlap by 80%. Each feature segment f^t is encoded into a spatial hypervector $S^t = f_{spat}(f^t)$. The five consecutive spatial hypervectors are then bound temporally to form a spatiotemporal hypervector representing the entire window $G^t = f_{temp}(S^t, S^{t+1}, S^{t+2}, S^{t+3}, S^{t+4})$. The dashed grey line shows the division between the low-dimensional feature domain and the high-dimensional hypervector domain. **c**, Spatial encoding function f_{spat} consists of computing a weighted sum of immutable item memory (IM) hypervectors IM(ch) weighted (·) by their corresponding features f_{ch}. The sum is bipolarized (σ) back to ±1. **d**, A temporal encoding function binds consecutive spatial hypervectors S^t together through permutation by k bits (ρ^k) and element-wise multiplication (×) operations. **e**,**f**, Principal component analysis for all classification windows from five trials of 13 single-DOF gestures. Analysis is shown for one participant in a baseline context. The top two principal components are plotted for features (**e**) and hypervectors (**f**) for each window. **g**, Proportion of explained variance for all principal component dimensions per dimension (top) and cumulative (bottom). Solid lines represent dimensions of the hypervector (HV) set; dashed lines represent dimensions of the feature (Feat.) set.

the middle extension gesture (gesture 4), the gesture that produced the highest SNR across all three devices (Fig. 2c). The frequency contents of the different recordings were qualitatively similar. Spot SNR was calculated by comparing the power spectrum during a gesture performance to the power spectrum during rest (Fig. 2d). Overall SNR (calculated as the integral of spot SNR) varied for different channels and different gestures (Fig. 2e). The Cometa and CapgMyo systems exhibited better peak SNR for the best channels and associated best gestures, probably because those systems consist of differential, bipolar recording configurations with improved common-mode noise rejection. Despite this, median SNRs for the three systems were approximately the same. Furthermore, SNR values for our system were similar before and after long, 2 h wear sessions, during which participants could sweat and the conductive gel could smear (Supplementary Fig. 2).

Gesture recognition algorithm with adaptive learning

We implemented an HD computing algorithm[31] for training and inference of hand gestures. Raw data or preprocessed features (which are continuous-valued and low-dimensional) are projected onto 1,000D bipolar ($\{-1,+1\}^{1,000}$, for simplicity in describing mathematical operations) or binary ($\{0,1\}^{1,000}$, for implementation with digital logic) hypervectors with information fully distributed across all bits, analogous to the way the human brain utilizes vast circuits of neurons and synapses for learning and recall. Learning for classification tasks can be implemented with low-complexity element-wise majority functions for computing cluster centroids, or prototype hypervectors, which each represent a class. Similarly, inference can be implemented as a nearest neighbour search within the learned class prototype hypervectors to a given query hypervector using Hamming distance.

Figure 3 describes the algorithm for projecting sEMG data into hypervectors. Importantly, the same projection process is used for both learning and inference, allowing reuse of hardware modules for both in-sensor training and classification[43]. Signals are processed in 250 ms sliding windows with 80% overlap, resulting in a latency of 250 ms and classification throughput of 20 predictions per second, which are suitable for real-time control[44,45] (Fig. 3a). Each window is subdivided into five 50 ms feature segments, with the MAV feature being calculated per channel for each segment. Feature selection and hyperparameter tuning are described in more detail in Supplementary Note 2.

For each segment index t, the feature vector f^t is projected into a hypervector S^t representing spatial information, that is, which electrode channels have higher feature values (Fig. 3b,c). Electrode channels are represented by unique, pseudo-random bipolar hypervectors stored in an item memory (IM), which remains unchanged throughout the training and deployment of the algorithm. Each electrode channel vector IM(ch) is scaled by its corresponding feature value f^t_{ch}, and all scaled vectors are summed element-wise. The resultant spatial hypervector is then formed as $S^t = \sigma\left(\sum_{ch} \mathrm{IM(ch)} f^t_{ch}\right)$, where σ is a bipolar threshold function that turns positive elements to +1 and negative elements to −1 (Fig. 3c).

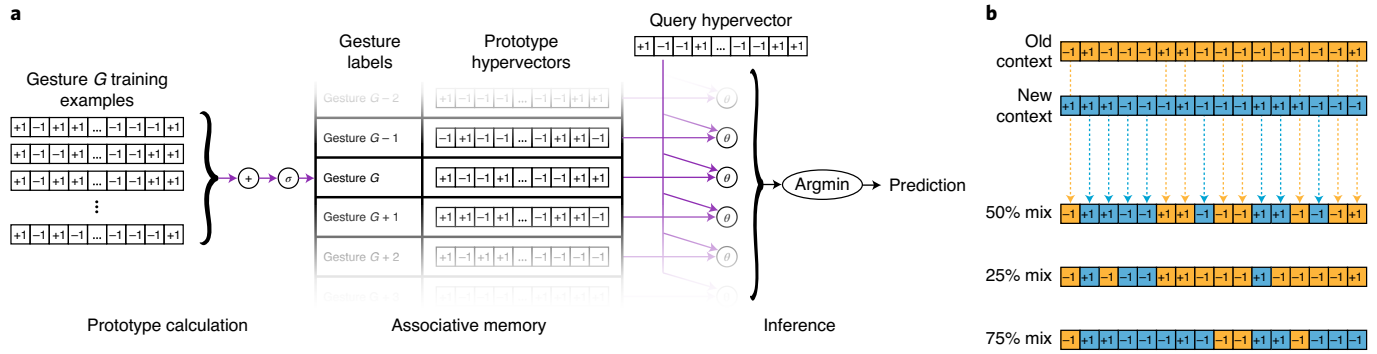

Fig. 4 | AM operations for training, accessing and contextual updating. a, Usage of the AM, from prototype calculation and storage for training (left) to Hamming distance calculation for inference (right). Prototype hypervectors are created from training examples for each gesture by taking the majority of each element across all examples. For bipolar hypervectors, this can be implemented as a sum (+) and bipolarization (σ). During classification of a query vector, the nearest neighbour among stored prototypes is found using the Hamming distance (θ) as the metric. **b**, Bitwise merge of two prototype hypervectors approximates finding weighted centroids of clusters from different contexts of the same gesture class. The updated hypervector randomly takes elements from the stored prototype hypervector and the new context prototype hypervector. The proportion of bits taken from each hypervector determines the relative weight of each context in the updated prototype.

The group of five spatial hypervectors is then encoded into a single spatiotemporal hypervector G^t (Fig. 3b,d). The order in which they occur is encoded by performing a bitwise rotation by k bits (ρ^k), with increasing k for older hypervectors (Fig. 3d). The rotated hypervectors are multiplied together element-wise, resulting in a single 1,000D bipolar hypervector representing the 250 ms window.

To visualize this projection into HD space, we performed principal component analysis (PCA) on the classification windows consisting of five feature vectors, as well as on projected spatiotemporal hypervectors from trials of the single-DOF gestures performed by one participant (Fig. 3e–g). The top two principal components (Fig. 3e,f) show general clustering of different gestures, with hypervector clusters having more circular groupings that can be better represented by their cluster centroids. The dimensionality of the overall dataset can be seen in the proportions of variance explained by each principal component dimension (Fig. 3g). For the hypervector dataset, nearly 500 principal component dimensions are required to explain 90% of the variance, whereas for feature vectors only ~10 dimensions are needed. This indicates that information is properly distributed across the large number of elements.

The encoded spatiotemporal hypervectors can be used either as training examples for creating or updating a model or as queries for inference using a trained model (Fig. 4). A prototype hypervector for each class is formed by computing the class centroid. For binary and bipolar hypervectors, this amounts to finding the majority of each element across all training examples. These prototype hypervectors are then stored in an associative memory (AM), an entirely feedforward operation with a single pass over training data (Fig. 4a). This is in contrast to other neuro-inspired approaches in which training often employs sophisticated, iterative frameworks and is much more computationally demanding than classification (for example, gradient descent with backpropagation[46]). Adding new classes to the model simply involves adding new prototype hypervectors to the AM, again differentiating HD computing from other algorithms that may require full retraining or modifications to the architecture. Once all prototypes have been computed and stored, classification involves finding the nearest-neighbouring prototype to a query hypervector (Fig. 4a).

To update a stored prototype to incorporate information about a new situational context for the same gesture, an approximated majority vote can be performed by merging elements from both the stored and the newly computed prototype hypervectors (Fig. 4b). Different proportions of new and old elements can be taken to weigh the contribution of each context. As more contexts are encountered, this proportion can be changed to tune the decay rate to avoid catastrophic forgetting of initial contexts. Very little additional overhead is required to perform model updates, compared to standard gesture recognition algorithms like support vector machines, in which new support vectors must be selected among new and old training examples[47,48], or linear discriminant analysis, in which training set statistics must be recalculated to a relatively high degree of precision[49,50].

Real-time in-sensor implementation of adaptive learning and classification

We tuned model hyperparameters and validated the HD algorithm using the offline dataset (Supplementary Note 2) before optimizing it for efficient implementation using hardware description language (HDL) for synthesis on the device's FPGA. Raw 15-bit analogue-to-digital converter codes were used as the input for feature extraction, with incremental calculations of the 50-sample MAV performed with each new sample. The implemented arithmetic operations consisted only of addition, two's complement inversion, and arithmetic right shift for division. Features were quantized and saturated to 6-bit integers based on analysis of the offline dataset, optimizing for the dynamic range (range divided by step size) given the arithmetic requirements.

The 1,000-dimensional item memory elements were generated sequentially using a cellular automaton with a hardcoded seed[51] for a smaller memory footprint. Subsequent processing steps were exactly as described in the previous section on HD classification architecture, replacing algebraic operations on bipolar hypervectors with Boolean operations on binary vectors[52]. A shift register consisting of 21 hypervectors was used as the AM to store trained prototype hypervectors and search for the closest class during inference. A single contextual update with 50% weighting was enabled for each AM entry by merging a predetermined set of 500 bits from a newly trained prototype into the stored one.

The implementation utilized 84% of resources available on our FPGA (Supplementary Note 3). Simulations verified that the algorithm (after acquiring the last sample of the last 50 ms feature window) had a latency of 539 cycles, or 26 µs when running at 20.48 MHz (Supplementary Note 4). The FPGA modules for

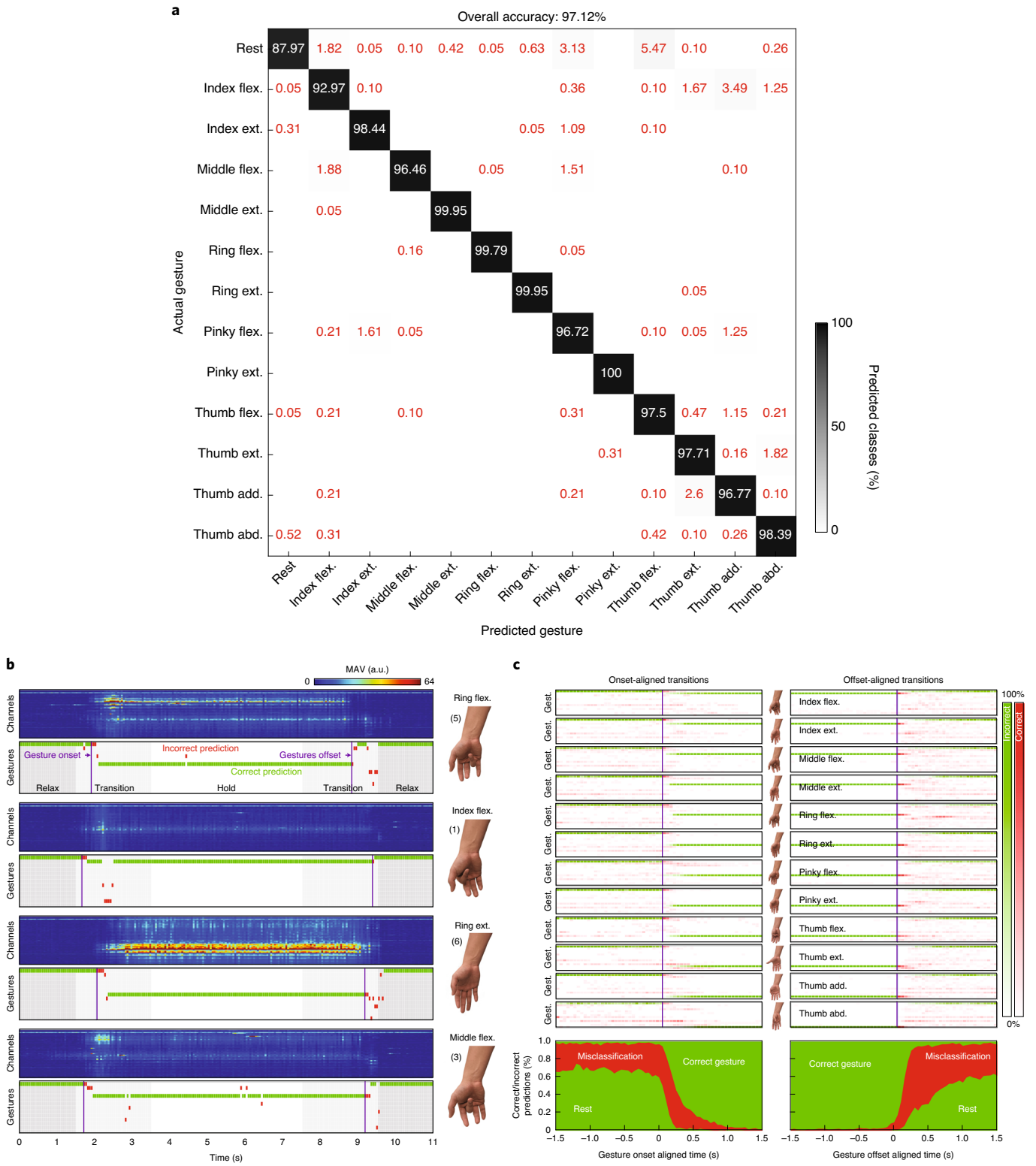

Fig. 5 | Real-time, in-sensor classification performance in the baseline context. a, Classification confusion matrix during 4 s hold periods of baseline context testing for real-time experiments. White text values are percentages of correct predictions, red text value are percentages of incorrect predictions. The greyscale coloured background represents the proportion of predicted classes. **b**, Examples of real-time prediction outputs for four gesture trials. For each gesture trial, the top plot displays the features calculated over 50 ms segments for all channels. The bottom plot shows the 20 Hz classification results relative to trial timing. Purple vertical lines denote offline-calculated gesture onset in the first transition period and gesture offset in the second transition period. The vertical position of predictions represents gesture class, and predictions are colour-coded based on accuracy with respect to offline-estimated ground truth labels. **c**, Gesture onset-aligned (left) and offset-aligned (right) prediction statistics over all baseline context tests. Each of the first 13 rows shows the averaged 20 Hz classification results for a given gesture, with more darkly coloured blocks indicating a higher proportion. Green blocks indicate correct predictions; red blocks indicate incorrect predictions. The bottom plots show the percentages of correct (green) and incorrect (red) classifications over time, relative to the onset and offset times.

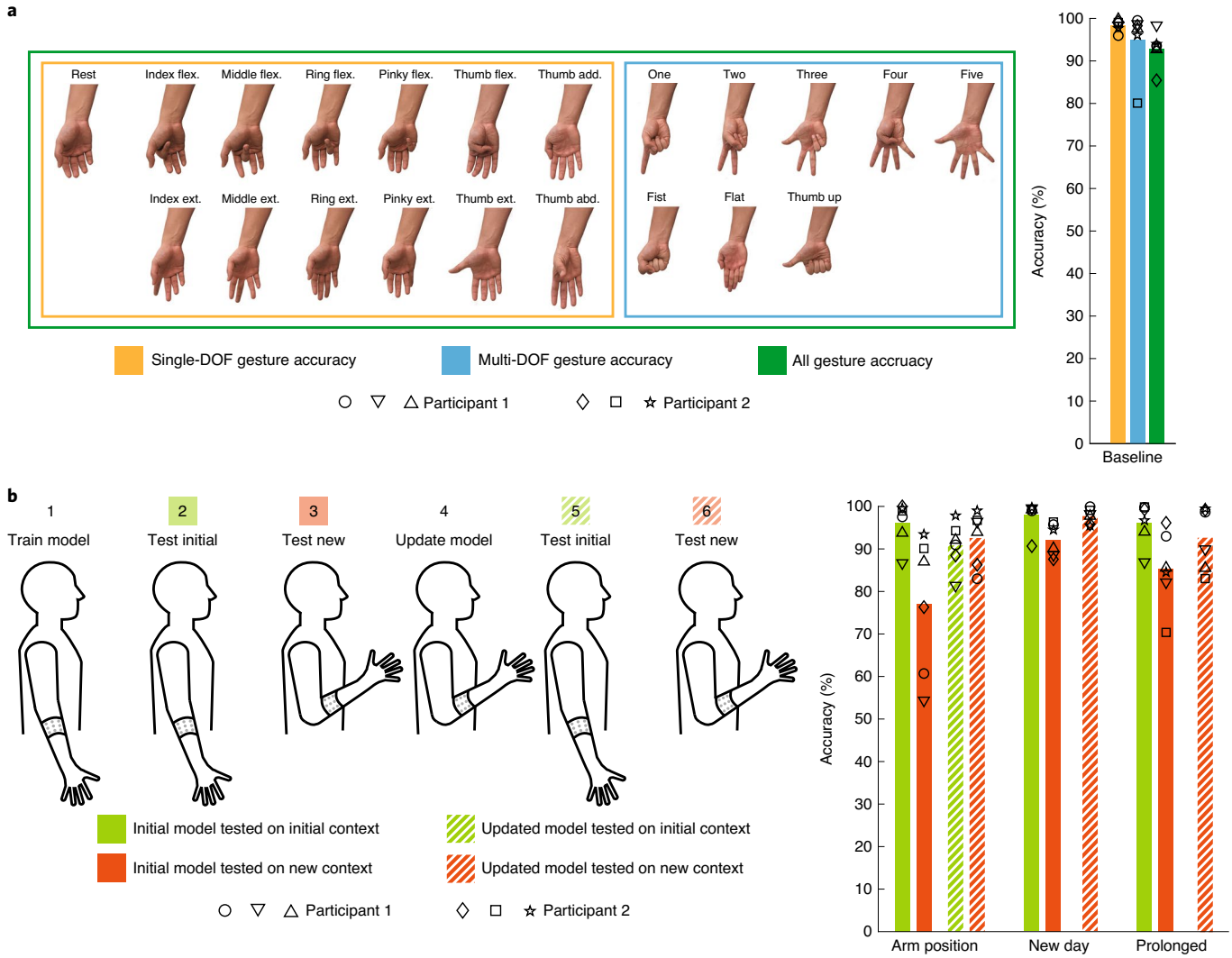

Fig. 6 | In-sensor training, update and classification results. Two participants performed each experiment three times. Bars represent the mean accuracy across all six trials, with overlaid data points for each individual trial. **a**, The ability to update the HD classification model with new gestures. An initial model was trained and tested on single-DOF gestures only (yellow bar). The model was then updated with multi-DOF gestures to cover all 21 gestures (green bar). Results for a separate model trained and tested on multi-DOF gestures only are also illustrated (blue bar). **b**, The ability to update the HD classification model with new contexts using only the single-DOF gestures. For three different context changes (arm position, new wear session and prolonged wear), an initial model was trained on the initial context (step 1) and tested on both the initial (step 2) and the new (step 3) contexts (solid bars). The model was then updated using a single trial of each gesture in the new context (step 4), and again tested (steps 5 and 6) on both contexts (striped bars). Updated models were tested only in the new context for the new wear session and prolonged wear experiments because the old context was no longer available.

classification were thus clock-gated to only be active during the first 26 µs after receiving a new sample, consuming no dynamic energy during the remaining time. Simulations and source meter measurements showed that the in-sensor HD algorithm operates with 2.437 µJ per sEMG sample and 4.39 µJ per classification every 50 samples (Supplementary Note 5). Model training and updating consumed similar energy to classification steps. In all, the on-board algorithm accounted for less than 6% of the overall device power.

We verified the online, standalone operation of our device with two participants who were familiar with myocontrol. Four types of experiment were each conducted three times to test varying device usage conditions with all training, inference and updates performed completely on board and in real time. A custom-made graphical user interface (GUI) was used to instruct the user on what gesture to perform, transmit the correct gesture label and the operation mode (train/infer/update) to the device, and log data. Both raw sEMG data and the classified gesture class were streamed back to the GUI and displayed in real time, giving the participant visual feedback during inference modes.

Figure 5 shows real-time classification accuracy and prediction time series from the device during the performance of single-DOF gestures in the baseline context immediately after device donning and with a relaxed arm. Each gesture was performed once during training, including a trial at the beginning for the rest gesture. Participants were instructed to begin each gesture within a 2 s transition period, hold the gesture for a 4 s steady-state period, and relax back to the rest gesture within a second transition period, with 3 s of relaxation between each gesture (Fig. 5b). The GUI was configured to only enable the training mode on the device during the 4 s steady-state period. Exactly the same procedure was then repeated with the device in inference mode. Figure 5a shows the classification accuracy and confusion matrix for the 4 s steady-state periods of each gesture.

Table 1 | Comparison of this work with other sEMG systems

	Amma et al. (2015)[26]	Liu et al. (2017)[12]	Moin et al. (2018)[32]	Pancholi and Joshi (2019)[29]	Benatti et al. (2019)[30]	Cerone et al. (2019)[28]	This work
Wearable form factor	✗	✓	✓	✓	✗	✓	✓
Electrode type	Flexible PCB	Individual surface patches	Flexible PCB	Ag-AgCl electrode band	Individual Ag-AgCl	Flexible PCB	Flexible screen-printed
Electrodes per array	192	4	64	8	8	32	64
Wireless streaming	✗	✓	✓	✗	✓	✓	✓
In-sensor classification	✗	✓	✗	✓	✓	✗	✓
In-sensor model training	✗	✗	✗	✓	✓	✗	✓
In-sensor adaptive update	✗	✗	✗	✗	✗	✗	✓
Number of classes	27	10	5	6	11	-	21

Actual gesture onset and offset times were determined offline with a combination of visual and changepoint analysis methods so as to estimate a ground truth label (Fig. 5b). Although the classification model was not trained on data that included gesture transitions or relaxation between trials, the device still performed inference during these periods. Gesture onset- and offset-aligned predictions across baseline context tests in all 24 experiments are shown in Fig. 5c. A higher error rate during relax periods (during which the prediction should be the rest gesture) is seen both before and after gesture performance, suggesting that the short periods between gesture movements may be included as training data to improve performance. Classification of the rest state is inherently more difficult, given that our hypervector projection algorithm encodes relative feature values rather than absolute feature values. Additionally, gesture transitions produced errors, requiring about 500 ms from gesture onset (including algorithm latency) to achieve 80% classification accuracy. The larger amount of misclassification following gesture offset is in part due to unintended movements in the opposite direction or of other finger degrees of freedom during transitions back to rest.

Figure 6 shows the steady-state gesture classification accuracy for the different experiments requiring online, in-sensor training and model updates. The first experiment (Fig. 6a) demonstrated the ability to update the HD classification model with new gestures. We trained and tested an initial model with only the single-DOF gesture subset (Fig. 6a, yellow bar) and achieved an average classification accuracy of 98.34%. We then updated the model with the multi-DOF gesture subset and tested on all 21 gestures (Fig. 6a, green bar) with 92.87% accuracy—a 5.47% degradation. Training and testing on the multi-DOF gesture only resulted in a similar accuracy of 95.04% (Fig. 6a, blue bar).

In the next three experiments, we demonstrated the ability to update the HD classification model with new contexts using only the single-DOF gestures. The context changes we explored were new arm position (going from the relaxed position with the arm at the side to the elbow rested on an armrest in an arm-wrestling position), new day with the device doffed and re-donned in between (>16 h apart, mean 19 h), and prolonged wear with the device worn while going about daily activities for 2 h. An initial model was trained and tested in the baseline context (Fig. 6b, solid green bar). The model was then tested in the new context, prior to updating (Fig. 6b, solid red bar). This resulted in, on average, an 11.89% accuracy degradation. The model was then updated in the new context and tested again (Fig. 6b, striped red bar). In the case of arm position, the updated model was also tested in the initial context (Fig. 6b, striped green bar). In each case, accuracy was recovered after updating the model for the new context, with an average improvement of 9.50% compared to the initial model.

A comparison of our device to existing state-of-the-art sEMG systems is presented in Table 1. Our system is the only one tested in an incremental learning framework with in-sensor training, inference in multiple new situational contexts, and incremental updates to adapt to new contexts and recover accuracy. Compared to the systems that are capable of in-sensor classification, ours has the most electrode channels by at least a factor of eight. The custom-designed flexible electrode array and low-power integrated circuit front ends allow us to maintain a small form factor and low power consumption, comparable to systems with a smaller number of channels, which are important for device wearability. A more detailed comparison of device implementations is provided in Supplementary Table 5. Our custom hardware implementation of the algorithm on an FPGA achieved the best classification energy efficiency, even with more gesture classes. Note that the total energy required for the processing (feature extraction and classification) was slightly higher compared to other reports because of the larger number of channels.

Conclusions

We have reported a self-contained, wearable sEMG biosensing system that uses HD computing to process and classify hand gestures. All the functionalities of our system, including data acquisition using a printed flexible electrode interface, as well as training and classification with a machine-learning model, are incorporated into a compact device that requires no external computation. Our biosensing system is comfortable to wear and offers fast initial training as well as on-the-fly adaptation, which are crucial for wearable human–machine interface applications in which physiological signals vary from user to user and are not stationary. In contrast to other state-of-the-art gesture-recognition systems, our device can perform training, inference and model updates locally and in real time to adapt to changing situational contexts.

Our results are promising for the development of standalone wearable devices with full, in-sensor machine-learning capabilities. Two natural extensions of this work are the incorporation of additional situational contexts (such as a larger variety of arm positions) and the incorporation of gesture transitions, which can either be treated as new contexts or as separate classes. As more information is encoded into gesture prototype hypervectors, it may eventually become beneficial to trade off memory footprint for improved classification performance and to represent gesture classes with more than one prototype hypervector. Finally, although our electrode array geometry was designed for forearm sEMG acquisition, the

low-cost and low-complexity screen-printing process can be used to create patterns for interfacing with other parts of the human body. This could enable a wider range of applications requiring low-latency, adaptive processing of physiological signals, such as electrocardiography or electroencephalography.

Methods

Hybrid interface for flexible and rigid components. An adapter PCB (Fig. 1c) with a 64-position FFC (XF2W-6415-1AE, Omron Electronics) on one side and two DF12 connectors (Hirose Electric) on the other side was fabricated to interface the flexible array traces with the rigid ICs for sEMG signal acquisition and processing.

Rigid components PCB fabrication. A custom wireless neural recording and processing module (Fig. 1d,e) capable of recording and processing 64 channels of electrical physiological signal data as well as wirelessly streaming raw signals and classified gestures back to a base station was attached to the flexible electrode array. This recorded and digitized the sEMG signals from the electrodes using a custom neuromodulation IC[35] (Cortera Neurotechnologies) with d.c.-coupled front ends and a 1 kHz sampling rate. Data aggregation and preprocessing as well as HD model training, inference and update were performed on an SoC FPGA (SmartFusion2 M2S060T, Microsemi). A 2.4 GHz radio (nRF51822, Nordic Semiconductor) streamed out the raw sEMG signals and the classified labels and distances. These components enabled small form factor and low-power operation (~6 h of classification using a standard 240 mAh 3.7 V Li-ion battery), while eliminating the need for bulky individual cables connecting each electrode to the neural front end, making the device comfortably wearable for extended periods with enhanced signal quality. Moreover, digitizing the signal next to its source increases the SNR.

The fabricated eight-layer PCB also contained a six-axis accelerometer and gyroscope (MPU-6050, InvenSense), a 512 Mb low-power SDRAM (MT46H32M16LFBF-5, Micron Technology) and an extra 64-channel neuromodulation IC[35] (Cortera Neurotechnologies) for the purpose of providing inertial sensor data, more on-board memory and an extended number of sEMG channels to 128, respectively, although they are unused in this work.

Graphical user interface. A custom GUI developed in Python 3.7 was run on the base station for receiving and logging the streamed data, configuring the sEMG signal acquisition system and providing participants with gesture information and timing. The GUI enabled experimenters to select gesture subsets, update metadata for the saved files and update gesture timing prior to each recording session. Moreover, it provided visual feedback to participants about the current classified gesture.

Array application. Before wrapping the array around the participant's forearm, a small drop of conductive hydrogel (SignaGel Electrode Gel, Parker Laboratories) was applied to each electrode surface individually to improve the skin–electrode interface impedance. The array was laid on a flat surface with electrodes facing up, and the participant was instructed to roughly align the centre of the array to the ulna of their dominant hand. The ends of the array were then wrapped around each side of the forearm while the participant remained at rest, and the ends were joined together with strips of adhesive tape. A single commercial Ag/AgCl electrode (H124SG, Covidien Kendall) was attached to the participant's elbow on the same arm as a reference potential for the voltage measurements (Fig. 1a).

Gesture recognition experiments. Participants were asked to perform a total of 21 different gestures consisting of a rest/relax position and two subsets of finger movements and positions: (1) single-DOF flexions and extensions of individual fingers and (2) multi-DOF hand postures involving multiple fingers (Fig. 2a). Each participant performed a total of four experiments, with each experiment repeated three times.

Experiment 1 was designed to test a baseline accuracy for the single- and multi-DOF gesture subsets separately, and an updated model containing all 21 gestures by adding multi-DOF gestures to the existing single-DOF trained model. Experiment 2 then introduced an arm position context variation to the single-DOF gestures. The participant performed each gesture with their elbow rested on an armrest in an arm-wrestling position. For experiment 3, the model was first trained and tested on day 1, and then updated on day 2 after doffing and re-donning the device in approximately the same location on the arm. This introduced a new wear session context variation. In experiment 4, participants were given a 2 h break between the initial training and subsequent update, during which they wore the sEMG acquisition device while going about their daily activities. This introduced a prolonged wear contextual variation.

Each trial lasted 8 s (Fig. 2b), with 3 s of rest before the next trial. The participant was told to begin the gesture within a 2 s transition window, which contained the transient, non-stationary part of the sEMG signal for that gesture. After the 2 s transition window, the participant was asked to hold the gesture for 4 s, constituting the steady-state part of the sEMG signal. Finally, the participant was directed to return to the rest position within another 2 s transition window. These directions ensured that the steady-state portion of the gesture could easily be labelled as part of the middle 4 s segment. All experiments were performed in strict compliance with the guidelines of IRB and were approved by the Committee for Protection of Human Subjects at University of California, Berkeley (protocol title, Flex EMG Study; protocol no., 2017-10-10425). Informed consent was obtained from all participants.

Data segmentation. For this study, only data from the 4 s steady-state hold period were used for classification. During training and testing, we generated a spatiotemporal vector for every 250 ms segment of data, sliding by 50 ms MAV feature windows (200 ms overlap). Thus, within a single 4 s gesture trial, 76 different vectors were encoded either for accumulating in the AM or for making 76 different inferences. Classification accuracy was calculated as the percentage of inference results that matched the labelled gesture, without any post processing or voting. Commands were sent to the device to put it in training, inference or update mode, synchronized to the experiment and instructions.

Algorithm implementation. The in-sensor algorithm was implemented in Verilog HDL and synthesized using Libero SoC design software (Microsemi Corp.). The FPGA resource utilization and algorithm latency are provided in Supplementary Notes 3 and 4, respectively. The algorithm energy per classification is calculated based on the instantaneous power measurements shown in Supplementary Note 5 (measured by an Agilent B2902A precision source/measure unit in 20 µs intervals) and cycle counts (Supplementary Note 4).

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

Data availability

Both the offline sEMG dataset and the real-time experiment data collected for this study are available at https://github.com/flexemg/flexemg_natelec. NinaPro data were accessed via http://ninapro.hevs.ch (Dataset 4) and CapgMyo data via http://zju-capg.org/myo/data/ (DB-c).

Code availability

The source code used for offline model validation, in-sensor implementation and analysis of results is available at https://github.com/flexemg/flexemg_natelec.

Received: 16 October 2019; Accepted: 4 November 2020;
Published online: 21 December 2020

References

1. Kim, J., Campbell, A. S., de Ávila, B. E. F. & Wang, J. Wearable biosensors for healthcare monitoring. *Nat. Biotechnol.* **37**, 389–406 (2019).
2. Bandodkar, A. J., Jeerapan, I. & Wang, J. Wearable chemical sensors: present challenges and future prospects. *ACS Sens.* **1**, 464–482 (2016).
3. Khan, Y., Ostfeld, A. E., Lochner, C. M., Pierre, A. & Arias, A. C. Monitoring of vital signs with flexible and wearable medical devices. *Adv. Mater.* **28**, 4373–4395 (2016).
4. Patel, S., Park, H., Bonato, P., Chan, L. & Rodgers, M. A review of wearable sensors and systems with application in rehabilitation. *J. Neuroeng. Rehabil.* **9**, 21 (2012).
5. Nakata, S. et al. A wearable pH sensor with high sensitivity based on a flexible charge-coupled device. *Nat. Electron.* **1**, 596–603 (2018).
6. Huang, Z. et al. Three-dimensional integrated stretchable electronics. *Nat. Electron.* **1**, 473–480 (2018).
7. Gao, W. et al. Fully integrated wearable sensor arrays for multiplexed in-situ perspiration analysis. *Nature* **529**, 509–514 (2016).
8. Imani, S. et al. A wearable chemical-electrophysiological hybrid biosensing system for real-time health and fitness monitoring. *Nat. Commun.* **7**, 11650 (2016).
9. Lochner, C. M., Khan, Y., Pierre, A. & Arias, A. C. All-organic optoelectronic sensor for pulse oximetry. *Nat. Commun.* **5**, 5745 (2014).
10. Khan, Y. et al. A flexible organic reflectance oximeter array. *Proc. Natl Acad. Sci. USA* **115**, E11015–E11024 (2018).
11. Benatti, S. et al. A versatile embedded platform for EMG acquisition and gesture recognition. *IEEE Trans. Biomed. Circuits Syst.* **9**, 620–630 (2015).
12. Liu, X. et al. The virtual trackpad: an electromyography-based, wireless, real-time, low-power, embedded hand-gesture-recognition system using an event-driven artificial neural network. *IEEE Trans. Circuits Syst. II* **64**, 1257–1261 (2017).
13. Gu, Y., Yang, D., Huang, Q., Yang, W. & Liu, H. Robust EMG pattern recognition in the presence of confounding factors: features, classifiers and adaptive learning. *Expert Syst. Appl.* **96**, 208–217 (2018).
14. Young, A. J., Hargrove, L. J. & Kuiken, T. A. The effects of electrode size and orientation on the sensitivity of myoelectric pattern recognition systems to electrode shift. *IEEE Trans. Biomed. Eng.* **58**, 2537–2544 (2011).

15. Zhang, X. & Huang, H. A real-time, practical sensor fault-tolerant module for robust EMG pattern recognition. *J. Neuroeng. Rehabil.* **12**, 18 (2015).
16. Hargrove, L., Englehart, K. & Hudgins, B. A training strategy to reduce classification degradation due to electrode displacements in pattern recognition based myoelectric control. *Biomed. Signal Process. Control* **3**, 175–180 (2008).
17. Tkach, D., Huang, H. & Kuiken, T. A. Study of stability of time-domain features for electromyographic pattern recognition. *J. Neuroeng. Rehabil.* **7**, 21 (2010).
18. Milosevic, B., Farella, E. & Benatti, S. Exploring arm posture and temporal variability in myoelectric hand gesture recognition. *Proc. 7th IEEE Int. Conf. on Biomedical Robotics and Biomechatronics* **2018**, 1032–1037 (2018).
19. Krishnan, K. S., Saha, A., Ramachandran, S. & Kumar, S. Recognition of human arm gestures using Myo armband for the game of hand cricket. In *IEEE International Symposium on Robotics and Intelligent Sensors* (*IRIS*) 389–394 (IEEE, 2017); https://doi.org/10.1109/IRIS.2017.8250154
20. Ploengpit, Y. & Phienthrakul, T. Rock–paper–scissors with Myo armband pose detection. In *2016 International Computer Science and Engineering Conference* 1–5 (IEEE, 2017).
21. Nyomen, K., Romarheim Haugen, M. & Jensenius, A. R. MuMYO—evaluating and exploring the MYO armband for musical interaction. In *Proc. International Conference on New Interfaces For Musical Expression* 215–218 (Louisiana State Univ., 2015).
22. Abreu, J. G., Teixeira, J. M., Figueiredo, L. S. & Teichrieb, V. Evaluating sign language recognition using the Myo armband. In *2016 XVIII Symposium on Virtual and Augmented Reality* 64–70 (IEEE, 2016).
23. Morais, G. D., Neves, L. C., Masiero, A. A. & Castro, M. C. F. Application of Myo armband system to control a robot interface. In *Proc. BIOSTEC 2016* 227–231 (2016).
24. Xu, Y., Yang, C., Liang, P., Zhao, L. & Li, Z. Development of a hybrid motion capture method using MYO armband with application to teleoperation. In *2016 IEEE International Conference on Mechatronics and Automation* 1179–1184 (IEEE, 2016).
25. Chen, X. & Wang, Z. J. Pattern recognition of number gestures based on a wireless surface EMG system. *Biomed. Signal Process. Control* **8**, 184–192 (2013).
26. Amma, C., Krings, T., Böer, J. & Schultz, T. Advancing muscle–computer interfaces with high-density electromyography. In *Proc. 33rd Annual ACM Conference on Human Factors in Computing Systems* 929–938 (2015).
27. Geng, W. et al. Gesture recognition by instantaneous surface EMG images. *Sci. Rep.* **6**, 36571 (2016).
28. Cerone, G. L., Botter, A. & Gazzoni, M. A modular, smart and wearable system for high density sEMG detection. *IEEE Trans. Biomed. Eng.* **66**, 3371–3380 (2019).
29. Pancholi, S. & Joshi, A. M. Electromyography-based hand gesture recognition system for upper limb amputees. *IEEE Sens. Lett.* **3**, 1–4 (2019).
30. Benatti, S. et al. Online learning and classification of EMG-based gestures on a parallel ultra-low power platform using hyperdimensional computing. *IEEE Trans. Biomed. Circuits Syst.* **13**, 516–528 (2019).
31. Kanerva, P. Hyperdimensional computing: an introduction to computing in distributed representation with high-dimensional random vectors. *Cogn. Comput.* **1**, 139–159 (2009).
32. Moin, A. et al. An EMG gesture recognition system with flexible high-density sensors and brain-inspired high-dimensional classifier. In *2018 IEEE International Symposium on Circuits and Systems* (*ISCAS*) 1–5 (IEEE, 2018); https://doi.org/10.1109/ISCAS.2018.8351613
33. Rahimi, A., Tchouprina, A., Kanerva, P., Millán, J. D. R. & Rabaey, J. M. Hyperdimensional computing for blind and one-shot classification of EEG error-related potentials. *Mob. Netw. Appl.* **25**, 1958–1969 (2020).
34. Burrello, A., Schindler, K., Benini, L. & Rahimi, A. One-shot learning for iEEG seizure detection using end-to-end binary operations: local binary patterns with hyperdimensional computing. In *Proc. 2018 IEEE Biomedical Circuits and Systems Conference* (*BioCAS*) 1–4 (IEEE, 2018); https://doi.org/10.1109/BIOCAS.2018.8584751
35. Johnson, B. C. et al. An implantable 700-μW 64-channel neuromodulation IC for simultaneous recording and stimulation with rapid artifact recovery. In *IEEE 2017 Symposium on VLSI Circuits, Digest of Technical Papers* C48–C49 (IEEE, 2017); https://doi.org/10.23919/VLSIC.2017.8008543
36. Cui, H. W. et al. Ultra-fast photonic curing of electrically conductive adhesives fabricated from vinyl ester resin and silver micro-flakes for printed electronics. *RSC Adv.* **4**, 15914–15922 (2014).
37. Lapatki, B. G., Van Dijk, J. P., Jonas, I. E., Zwarts, M. J. & Stegeman, D. F. A thin, flexible multielectrode grid for high-density surface EMG. *J. Appl. Physiol.* **96**, 327–336 (2004).
38. Dick, F. S., Bert, U. K., Bernd, G. L. & Johannes, P. V. D. High-density surface EMG: techniques and applications at a motor unit level. *Biocybern. Biomed. Eng.* **32**, 3–27 (2012).
39. Atzori, M. et al. Electromyography data for non-invasive naturally-controlled robotic hand prostheses. *Sci. Data* **1**, 1–13 (2014).
40. Pizzolato, S. et al. Comparison of six electromyography acquisition setups on hand movement classification tasks. *PLoS ONE* **12**, e0186132 (2017).
41. Atzori, M. et al. Building the NinaPro database: a resource for the biorobotics community. In *Proc. 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics* (*BioRob*) 1258–1265 (IEEE, 2012); https://doi.org/10.1109/BioRob.2012.6290287
42. Du, Y., Jin, W., Wei, W., Hu, Y. & Geng, W. Surface EMG-based inter-session gesture recognition enhanced by deep domain adaptation. *Sensors* **17**, 6–9 (2017).
43. Rahimi, A. et al. High-dimensional computing as a nanoscalable paradigm. *IEEE Trans. Circuits Syst.* **64**, 2508–2521 (2017).
44. Smith, L. H., Hargrove, L. J., Lock, B. A. & Kuiken, T. A. Determining the optimal window length for pattern recognition-based myoelectric control: balanŷcing the competing effects of classification error and controller delay. *IEEE Trans. Neural Syst. Rehabil. Eng.* **19**, 186–192 (2011).
45. Englehart, K. & Hudgins, B. A robust, real-time control scheme for multifunction myoelectric control. *IEEE Trans. Biomed. Eng.* **50**, 848–854 (2003).
46. Lecun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
47. Liu, J. Adaptive myoelectric pattern recognition toward improved multifunctional prosthesis control. *Med. Eng. Phys.* **37**, 424–430 (2015).
48. Diehl, C. P. & Cauwenberghs, G. SVM incremental learning, adaptation and optimization. In *Proc. International Joint Conference on Neural Networks* Vol. 4, 2685–2690 (IEEE, 2003); https://doi.org/10.1109/IJCNN.2003.1223991
49. Zhu, X., Liu, J., Zhang, D., Sheng, X. & Jiang, N. Cascaded adaptation framework for fast calibration of myoelectric control. *IEEE Trans. Neural Syst. Rehabil. Eng.* **25**, 254–264 (2017).
50. Zhang, H. et al. An adaptation strategy of using LDA classifier for EMG pattern recognition. *Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.* **2013**, 4267–4270 (2013).
51. Schmuck, M., Benini, L. & Rahimi, A. Hardware optimizations of dense binary hyperdimensional computing: rematerialization of hypervectors, binarized bundling and combinational associative memory. *ACM J. Emerg. Technol. Comput. Syst.* **15**, 1550–4832 (2019).
52. Kanerva, P. Binary spatter-coding of ordered K-tuples. in *Artificial Neural Networks — ICANN 96* (eds von der Malsburg, C. et al.) 869–873 (Lecture Notes in Computer Science Vol. 1112, Springer, 1996); https://doi.org/10.1007/3-540-61510-5_146

Acknowledgements

We thank E. Alon, A. Araujo, R. Muller, K. Lutz, H. Wu, M. Sadeghi, Cortera Neurotechnologies and Novacentrix. This work was supported in part by the CONIX Research Center, one of six centres in JUMP, a Semiconductor Research Corporation (SRC) programme sponsored by DARPA. This material is based, in part, on research sponsored by the Air Force Research Laboratory under agreement no. FA8650-15-2-5401, as conducted through the flexible hybrid electronics manufacturing innovation institute, NextFlex. The US Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory or the US Government. This research used the Savio computational cluster resource provided by the Berkeley Research Computing program at the University of California, Berkeley. Support was also received from sponsors of Berkeley Wireless Research Center, NSF Graduate Research Fellowship under grant no. 1106400, ETH Zurich Postdoctoral Fellowship programme and the Marie Curie Actions for People COFUND Program.

Author contributions

A. Moin, A.Z., A. Menon, S.B., G.A., S.T., J.T., N.Y., Y.K. and F.B. designed the hardware. A. Moin, A.Z. and A.R. developed and implemented the learning algorithm. A. Moin and A.Z. performed the experiments and analysis. L.B., A.C.A. and J.M.R. oversaw the project. A. Moin, A.Z., A.C.A. and J.M.R. wrote and edited the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41928-020-00510-8.

Correspondence and requests for materials should be addressed to A.M. or J.M.R.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

nature research

Corresponding author(s): Ali Moin, Jan M. Rabaey

Last updated by author(s): Oct 23, 2020

Reporting Summary

Nature Research 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 Research 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	A custom graphical user interface developed in Python 3.7 was used for all data collection. Firmware for the device was developed in C, Verilog, and Libero SoC v11.7. Data collection code can be found in the GitHub repository: https://github.com/flexemg/flexemg_natelec
Data analysis	Data was analyzed using Matlab R2019b (Mathworks, Inc.) and Python 3.7. Data analysis code can be found in the GitHub repository: https://github.com/flexemg/flexemg_natelec

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 Research 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 list of figures that have associated raw data
- A description of any restrictions on data availability

Both the offline sEMG dataset and the real-time experiment data collected for this study are available at https://github.com/flexemg/flexemg_v2. Ninapro data was accessed via http://ninapro.hevs.ch (Dataset 4), and Capgmyo data was accessed via http://zju-capg.org/myo/data/ (DB-c).

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	Validation of the machine learning model was performed on data collected from five subjects, with each subject performing 5 repetitions of each gesture in each context, allowing for 5-fold cross-validation studies. Real-time experiments were performed three times each for two subjects. Our experiments were intended to demonstrate device capabilities rather than show any biological or behavioral effects. Thus, multiple repeated experiments with the same two subjects were sufficient.
Data exclusions	No data were excluded from the analysis.
Replication	Each experiment was performed three times for each of two subjects. All attempts at replication were successful.
Randomization	Randomization was not relevant to our study since experiments were specific to each subject.
Blinding	Blinding was not relevant to our experiments as there were no subjective elements to assessing the device's capabilities.

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
☐	☒ Human research participants
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

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

Human research participants

Policy information about studies involving human research participants

Population characteristics	5 male able-bodied human subjects between 25 and 41 years old. The study population is a non-vulnerable highly-educated adult population who are the faculty, graduate students and staff of UC Berkeley.
Recruitment	A recruitment email was sent out that described the study and made it clear that participation is voluntary and that non-participation will have no impact.
Ethics oversight	Committee for Protection of Human Subjects at University of California, Berkeley (Protocol title: Flex EMG Study. Protocol number: 2017-10-10425). Informed consent was obtained from all participants.

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