

MyoKey: Inertial Motion Sensing and Gesture-based QWERTY Keyboard for Extended Realities

Kirill A. Shatilov, Young D. Kwon, Lik-Hang Lee, Dimitris Chatzopoulos, and Pan Hui, *Fellow, IEEE*

Abstract—Usability challenges and social acceptance of textual input in a context of extended realities (XR) motivate the research of novel input modalities. We investigate the fusion of inertial measurement unit (IMU) control and surface electromyography (sEMG) gesture recognition applied to text entry using a QWERTY-layout virtual keyboard. We design, implement, and evaluate the proposed multi-modal solution named MyoKey. The user can select characters with a combination of arm movements and hand gestures. MyoKey employs a lightweight convolutional neural network classifier that can be deployed on a mobile device with insignificant inference time. We demonstrate the practicality of interruption-free text entry with MyoKey, by recruiting 12 participants and by testing three sets of grasp micro-gestures in three scenarios: empty hand text input, tripod grasp (e.g., pen), and a cylindrical grasp (e.g., umbrella). With MyoKey, users achieve an average text entry rate of 9.33 words per minute (WPM), 8.76 WPM, and 8.35 WPM for the freehand, tripod grasp, and cylindrical grasp conditions, respectively.

Index Terms—EMG, Electromyography, Micro-gestures, Mobile input techniques.

1 INTRODUCTION

Augmented, mixed, and virtual reality (AR/MR/VR), located at the different points of the extended realities (XR) spectrum, struggle to achieve wide adoption and slowly approach the so-called plateau of productivity. However, with an ongoing process of miniaturization of electronic components, increasing density of pixels in head-mounted displays, and advancements in deep learning applications, XR is getting yet another round of attention. Notable advancements include Google Glass Enterprise 2 [1], North Focals [2]; a constantly growing entertainment-centered VR/MR market [3]; and relevant research projects [4], [5], [6], [7], [8], [9].

In opposition to the growing complexity and diversity of output techniques (retinal projection [10], pupil tracking for VR/MR helmets for sharper imagery [11] or waveguide-coupling AR displays [12], [13]), input modalities are limited to a relatively conservative set of physical keyboards and touch surfaces. Small-sized touch interfaces employed by smartglasses for text input cumber interaction with the digital overlays [14] and are usually inconvenient for both the direct user and a bystander [15]. For example, Google Glass users have to continuously hold an arm at eye level to select characters. Alternative approaches, such as head-gaze, vision-based hand gestures (e.g., Microsoft HoloLens) might be tedious [16], suffer from optical occlusion, and raise questions of social acceptance [17] and bystanders' privacy [18]. Speech-based input is not suitable for password and URL

input since it might disclose sensitive information [4] and be inappropriate in noise-free environments [17].

Thus, it is necessary to explore alternative modalities and design approaches for text input on XR headsets and smart glasses. Application of Surface electromyography (sEMG), a widely applied technology in the area of active prosthetics, might have been overlooked in general-purpose human-computer interaction (HCI). sEMG is an imaging technique used to evaluate myoelectrical currents elicited in muscles [19]. It is used as a modality to recognize intended gestures of amputees to control upper [19], [20] or lower limb [21] prosthesis, and in rehabilitation and sports [22]. The popularity of wrist-worn devices [23] that can host sEMG electrodes, and the existence of commercial sEMG-enabled wearables [24], [25], demonstrate the potential of a more diverse sEMG-based HCI [26].

However, it is challenging to design and implement a robust and efficient sEMG-based text entry system: such a system might suffer from an increasing number of gestures [27]; high offline recognition accuracy will not necessarily be maintained in real-time, as the signal-to-noise ratio (SNR) of sEMG signals is affected by the user's posture, gesture's intensity, electrode's position and electrical interference from nearby devices [28]. Thus, it is impractical to directly map alphanumeric symbols to a large number of gestures. Also, such a large set of gestures will discourage new users due to a high learning burden to memorize gesture sets. Therefore, an additional modality shall be considered as an auxiliary modality to tackle these limitations. Eventually, touchless multi-modal sEMG-enabled input protocols can enrich interaction experiences in XR but also eliminate the potential source of viral and bacterial contamination [29].

In this paper, we present MyoKey, a multi-modal text entry system we implemented for textual input on XR headsets. Myokey supports three sets of grasp micro-gestures

- Kirill A. Shatilov and Pan Hui are with the Department of Computer Science and Engineering, HKUST, Hong Kong.
Corresponding author: panhui@ust.hk
- Young D. Kwon is with Cambridge University, UK.
- Lik-Hang LEE is with KAIST, S. Korea, & University of Oulu, Finland.
- Dimitris Chatzopoulos is with University College Dublin, Ireland

Manuscript received 28 June 2021; revised xxx.

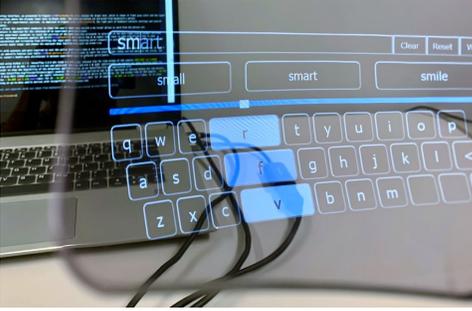


Fig. 1: The text entry layout of MyoKey

for text entry in a freehand condition and two busy-hand scenarios. MyoKey detects the user’s forearm orientation via an inertial measurement unit (IMU) and myoelectric signals by contracting muscles to select characters and words in a full QWERTY keyboard layout. In MyoKey, as depicted in Figure 1, the entire keyboard is divided into ten columns, each column contains four selection options of three characters and one (out of three) suggested word. The two modalities work complementary – the IMU-driven forearm movement first maps to a target column, and a gesture disambiguates the four options in the column. Accordingly, we design a deep learning classifier and a voting system to support robust EMG-based gesture recognition within a reasonable time window for the highly repetitive text entry task. It is important to note that we incorporate a recalibration mechanism to update the trained gesture recognition model to overcome the *session-dependency* of sEMG [30] when a user takes off MyoKey and wears it again. Additionally, MyoKey has the prominent feature of grasp micro-gestures [31]. Figure 2 depicts the gestural inputs in MyoKey under conditions of (a) freehand interaction and busy-hand interaction, e.g. holding (b) an umbrella or (c) a pen. Thus, MyoKey supports subtle interactions in the wild.

To evaluate MyoKey, we recruited a total of 12 participants. Throughout five text entry sessions, MyoKey achieves an averaged text entry rate of 9.33 words per minute (WPM) under the freehand scenario and 8.76 WPM and 8.35 WPM in the tripod and cylindrical grasp scenarios, respectively. As MyoKey is supported by a deep learning gestural recognition system, the error rate maintains at a practical level ranging from 6.88% to 9.81% for the three scenarios. It is worth mentioning that MyoKey considers user mobility by enabling grasping objects during text entry tasks outdoors.

In summary, our contributions are:

- (1) We design an IMU-driven and EMG-based text entry system for XR headsets and a corresponding QWERTY keyboard layout designated for the multi-modal solution.
- (2) We develop a lightweight sEMG pattern recognition deep learning classifier, that can run on mobile devices, achieving state-of-the-art accuracy with inference time below 10 ms on multiple gesture sets in real-time and a recalibration algorithm to mitigate potential deterioration in sEMG recognition over time.
- (3) We devise three different gesture sets for a single-handed eyes-free text entry system in several contexts, including micro-grasping gestures for interruption-free input while holding an object.

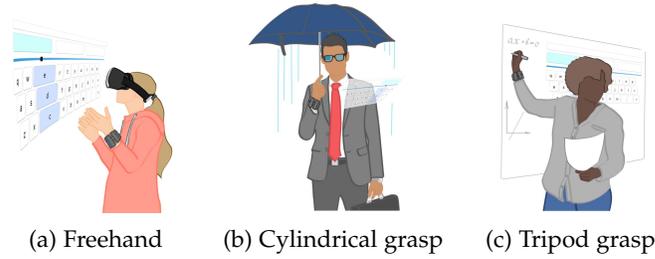


Fig. 2: Usage scenarios of the proposed solution: users can provide inputs when their hands are not occupied (Figure 2a), or when holding an object (Figure 2b, 2c).

2 RELATED WORK

Human muscles are controlled by neural impulses originating in the motor cortex of the brain and propagating across the peripheral nervous system towards motor neurons. The activity of the motor neurons is characterized by the flow of ions within muscle tissue. Intensity and polarization of the flow can be measured using needle-like electrodes implanted into the tissue itself (Electromyography) or on the adjacent surface of the skin (surface Electromyography, sEMG) [32]. sEMG capturing devices are characterized by temporal (sampling frequencies, in Hz, or samples per second) and spatial (number of electrodes and available precision) resolutions. In our experiments, we rely on the certain sEMG recording hardware, Myo band by Thalmic Labs. Myo Band samples the signal on the frequency of 200 Hz employing 8 electrodes with 16-bit resolution. sEMG signal acquired by the Myo Band for different gestures is visualized in Figure 3.

The acquired sEMG signal is classified with respect to performed gestures. Cognolato et al. [33] presented the MeganePro database, where multi-modal data (sEMG, IMU, gaze tracking, video recording, behavioral and clinical records) from intact subjects and amputees are aggregated. Kernel Regularized Least Squares (KRLS) classifier reported, for a window of 400 samples (approximately 208 ms) and 95% overlap between successive windows, 82% (for intact) and 63% (for subjects with amputated upper-limb) accuracy. Gesture recognition frameworks are also used in active prosthesis control to identify gestural intentions of amputees and micro-gestures of residual limbs [20]. Additionally, sEMG have been used to recognise held objects [34] and as a biometric modality to unlock smartphones [35].

sEMG pattern recognition accuracy can vary significantly due to several factors. One of them is the overtime smearing of the signal. To mitigate the non-stationary nature of sEMG, Zhai et al. [19] proposed a self-recalibrating classification routine with a pre-trained 2-layer convolutional neural network that recalibrates itself every session based on predictions from previous sessions. Moreover, there exists a gap between offline accuracy and real-time utility, where the gap is derived from gesture intensity, limb position, electrode shift, and transient changes in the signal [28].

MyoKey serves as a groundwork leveraging the EMG sensors for text entry tasks on wearable computers achieving high usability and user acceptance. The prior works propose the non-standard keyboards [36] and users need to learn the new text entry layouts, in addition to the burden

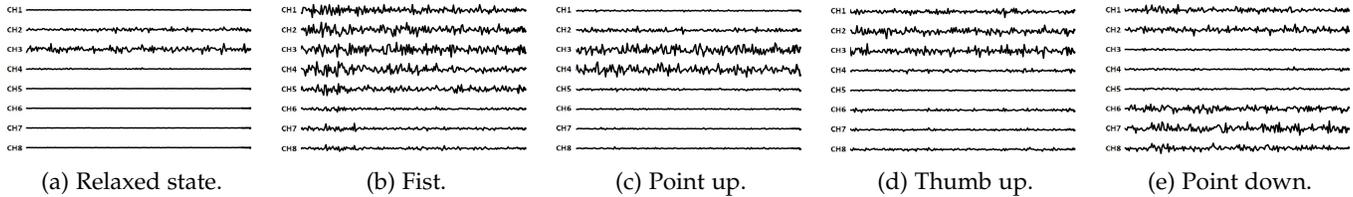


Fig. 3: Representation of sEMG signal of different gestures (Participant 1, freehand interaction).

from practicing new gestures. MyoKey leverages the user’s ingrained memory of the standard QWERTY keyboard layouts [14], in which the familiar layout can significantly reduce the learning efforts and hence improve usability.

2.1 Paradigm shifts in interaction modality

Soft keyboards for smartphones and wearable computers have received increasing attention [37]. The key challenge of typing on soft keyboards on wearable computers is the constrained size of touch interfaces [38]. Thus, many modalities have been employed for text entry on head-worn computers, including touch-based [39], IMU-driven [40], vision-based [41] [14], force-assisted [42], and electroencephalography (EEG) [43]. However, these modalities have their respective issues. It is impractical to accommodate a full-size QWERTY keyboard on a constrained-size headset with touch-based input [42]. Vision-based and IMU-driven interaction triggers lifting arms and intensive body movements, leading to ergonomic issues [14] while force-assisted and EEG-based interaction suffer from relatively low input rate [44] [42] [43].

Among the aforementioned modalities, sEMG has prominent advantages of subtle and non-invasive interaction. The advantage motivates us to explore the input methods with an alternative modality for the non-touchable interfaces on head-worn computers [45]. To the best of our knowledge, very limited works employ surface electromyography as the key modality for text entry. Existing works primarily focus on the disability and gestural inputs [46]. For example, MyoTyper [36] is an sEMG-based text entry system designated for amputees, resulting in a limited text entry rate of 2.56 wpm with a significantly high error rate (30%). In contrast, MyoKey demonstrates the practicality of EMG-based text entry for prospective usage with head-worn computers.

2.2 Alternative interfaces on shrinking headsets

Text entry interfaces have vastly evolved in various mobile devices. Besides the full traditional QWERTY keyboards, other options are being explored, like cubic layout [47]. Numerous alternative text entry systems have been proposed, in search of simplified interfaces accompanied by effective interaction for constrained wearable computers. An early work [48] considers tablets owning limited screen real estate and therefore proposes an ambiguous QWERTY keyboard shrunken to about one-third size of a full default keyboard. However, the screen real estate of head-worn computers becomes even smaller [14] in which the digital contents are not directly manipulable [42]. In PalmType [41], a digital

overlay of full QWERTY keyboard maps on the user’s palm by the see-through display of a head-worn computer. The users can tap on the character keys within the palm area, supported by an indoor time-of-flight camera decoding the tap gestures. However, PalmType not only neglects mobility but also occupies the majority of the screen and consequently hinders the seamless interaction between the users and the physical environment. Different from above vision-based [49] and force-assisted [42] approaches, this paper focuses on the interaction design of EMG-based text entry. However, the existing interaction design leveraging sEMG as an interaction modality is mainly limited to the gestural interaction, and the sEMG-based text entry interface is not appropriately addressed. A prototype named MyoType¹ allows users to input alphabetic and numerical symbols with static gestures. However, semaphoric gestures usually suffer from low input speed due to recognition overheads, and long-term usage is prone to fatigue [15].

2.3 Text Entry in the wild

Walking users and cluttered backgrounds can negatively impact the user performance of text-entry interfaces in the wild [50], [51], [52], [53]. Only a few works address the mobility issue in the user interaction with head-worn computers, which evaluate factors in real-world situations such as cluttered background [54], text reading speed in a walking path with obstacles [55], reading text [56] and visual cues [57] with peripheral vision, and text readability and legibility in a shaky condition [58]. Nevertheless, these works neglected the possibility of text entry in a common mobile scenario of object holding.

Even though the existing studies on object holding have tackled the issues such as the recognition of grasped object [34] and their gesture design [31], as well as the gestural recognition of holding an imaginary object (e.g. smartphone [59]), this paper is the first effort to consider the text entry when our hands are usually occupied by objects in a mobile situation. By employing EMG and IMU as the key modalities, MyoKey is a multi-modal text entry system that enables users to perform text entry in multiple scenarios including object grasping situations.

3 SYSTEM DESIGN

3.1 Text Entry Interface and Interaction Design

MyoKey employs the standard QWERTY layout containing 26 roman characters, a backspace key, and the white space

1. <https://github.com/Etienne/MyoType/>

key. The familiarity of the QWERTY layout enables faster pick-up rates to novice user [60]. Apart from the 26 characters, the rightmost column of the keyboard consists of the letter ‘p’ and two backspace keys ‘|’, while two white space keys ‘ ’ are located at the bottom row of the 8th and 9th columns. Above the QWERTY layout, three slots are designated for the top-3 recommended words. Figure 1 depicts the QWERTY layout in the proposed text entry system solution.

Employing sEMG as an input modality for text entry tasks is challenging, mainly because it is difficult to map the gestures with all-discrete keys on the keyboard. The prior works on sEMG-based text entry solutions focus on groups of disabilities. Their most common approaches are either the sEMG-based mouse pointing on unambiguous character keys [61] or gestural mapping to discrete character keys [36], resulting in less usable text entry performance. In addition, another evidence [27] shows that the recognition accuracy of EMG-based gestures is inversely proportional to the number of gestures.

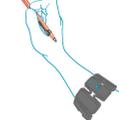
Instead of considering the approach of direct gesture-key mapping, we propose a divide-and-conquer strategy with an additional modality of IMU-driven forearm orientations in MyoKey to keep a minimal number of gestures. The forearm orientation maps to the columns, and each column corresponds to four options of three character keys and one recommended word. Therefore, MyoKey applies an ambiguous keyboard on the QWERTY layout, in which the ten columns are divided horizontally and evenly, and the three slots of word recommendations are separated into the ratio of 3:3:4. The disambiguation of the four options in every column relies on the four discrete EMG-based gestures (more details in Sections 3.2 and 3.3). As shown in Figure 1, a slider locates between the recommended words and the character keys. The slider serves as the visual cue to indicate the forearm orientation with respect to the keyboard columns. In addition, we reserve two keys for the backspace and the white space so the user can reach these key options in alternative columns and rows.

The user selects the column that contains the target input letter by alternating the direction of pointing around the yaw axis within a certain adjustable angle range (from 30 to 90 degrees, see subsection 4). The selected column is highlighted with a light blue color and expanded horizontally. Such an enlargement is not only visual, the yaw angle range that is related to the selected column is also increased to eliminate unwanted switching between columns caused by the change in orientation from walking or performing a character-selection gesture.

MyoKey serves as an alternative solution for the text entry on mobile headsets (freehand interaction) and demonstrates the prominent features in multiple mobile usage scenarios (busy-hand interaction). Thus, we define both freehand and grasp micro-gestures (see Table 1).

Freehand interaction: when the interaction is performed by empty hand mid-air or in a relaxed stance alongside the body. In the case of freehand interaction, we propose to use intuitive gestures of choosing top or bottom row by pointing up or down, respectively. The middle row is being selected by performing a horizontal thumb gesture. An additional factor contributing to the selection of gestures is the fact

TABLE 1: Input with freehand/ micrograsp gestures.

	Freehand	Tripod grasp	Cylindrical grasp
Idle			
Top Row			
Mid Row			
Bottom Row			
Suggestion			

that when performing certain gestures (e.g., wave-in or wave-out) orientational angle of the band might change significantly enough to change the input column.

Busy-hand interaction: when the input is performed with a hand while holding an object in order to deliver an interruption-free experience. Such interaction may include text input while holding a heavy object (an umbrella or a suitcase) or a small object (a pen or a stylus). For both cases, we propose grasp gestures to disambiguate between keys in the same column, as shown in Table 1. For the cylindrical grasp scenario (e.g., umbrella), we propose thumb up for the top, index finger flexion for the middle, and little finger flexion for the bottom row, respectively. In the case of tripod grasp (e.g., continuous input while holding a pen) pointing up for the top, little finger flexion for the middle, and pointing down for the bottom row are used [62]. In both scenarios squeezing the held object corresponds to the selection of a proposed suggestion.

The selection of these gestures is motivated by prior work [31] and preliminary assessment from both, *system* (recognition success and intensity) and *user* (intuitiveness and simplicity of performance) side.

3.2 Gesture recognition

Once a column is selected, a gesture is recognized over a time window (e.g., 100 ms, 200ms). The sensor does not guarantee the exact number of delivered samples, so samples are accumulated till their number reaches a certain

threshold (e.g., 100 samples). In the proposed solution we use a time window of 100 samples (see Figure 5). We employ a convolutional neural network (CNN) consisting of two convolutional layers and one fully-connected layer. The size of the input is $[8 \times 100]$ ($[channels \times samples]$); the first convolutional layer compresses every single channel of the input - it consists of 64 filters of size $[1 \times 25]$. The second convolutional channel (128 $[8 \times 4]$ filters) captures patterns across all channels at once. Finally, there is a dense layer, utilizing 500 units with a dropout rate of 50%, followed by softmax with five label outputs. Output labels represent recognized gestures: *standby state* (palm in case of freehand, or holding an object in busy-hand interaction), three-row selection gestures, and one suggestion selection.

3.3 Voting system

The integrated classifier achieves state-of-the-art [63], [64] gesture recognition static accuracy of 98% (see subsection 4.2). Yet, online performance is affected by multiple factors [28]: gesture intensity, body and arm posture, slight variations of the armband position. Those factors do not alternate significantly during the collection of training (and validation) set but might change afterward. To improve the robustness of the system to the factors affecting online performance, we introduce a voting system for gestural input. That is, a character within a column is chosen only if the classifier has recognized the same gesture multiple times within a fixed amount of time. In other words, each character within a column is associated with the number of votes $v_c = k, k \in 0, 1, 2$; once the classifier outputs the result, the number corresponding to the output is incremented; a character c is then selected if v_c reaches a certain threshold. In our system design, we set the default threshold as $k = 2$. It is clear that in the case of $k > 1$, the time of inputting each character increases. However, the increase of time is mitigated by overlapping the time windows on which the classifier's votes are based. We measure and adjust window overlap in percents (i.e. 0% overlap - windows have no common samples, 50% - second half of the first window is shared with the first half of the second window and so on). We use a 50% overlap by default and fine-tune it based on a user's preference (e.g., selection of characters is perceived to be too fast).

Once a selected character receives its first vote, it is highlighted distinctively from other characters within a column. If the character is finally selected for input, it is highlighted in a darker color to notify a user of successful input. Thus, another advantage of the voting system is that a user has a chance to correct their gesture or even change a column (and subsequently reset all received votes to zero) if they notice that character that has received the first vote is different from the intended one. The usage of such a voting system leads to a trade-off between the robustness of continuous classification and input time. Let us discuss it in detail below.

3.3.1 Robustness of continuous classification

Single Voting System: Assume that the probability of correctly classifying gestures is p . We put p as average classification accuracy of the developed gesture classifier,

$p = 0.943$, for a window of 25 samples, as reported in subsection 4.2). Next, the probability of inputting ten letters in a row correctly by using a single gesture is as follows²:

$$P(C|k = 1) = p^{10} = 0.556 \quad (1)$$

where $C = \{c_1, \dots, c_n\}$ represents a sequence of inputted characters with length n with and k is the number of votes used in the system.

Multiple Voting System: On the other hand, given a voting system with the number of votes $k = 2$, the probability of correctly inputting 10 intended symbols is increased significantly. First, we compute the probability of inputting a single correct character. In the voting system with $k \geq 2$, votes can be assigned to two other characters that are not intended, e.g. for an intended character 'u', two characters ('j' and 'm') in the same column can receive votes from the classifier and can be chosen for input if the number of votes for any of those two reaches the threshold. Note that we denote a correct (right) character as R and two wrong characters as W_1 and W_2 .

There exist three different cases when characters are votes for in a sequence : (1) vote twice for the correct character (1 outcome: RR), (2) vote once for a wrong character and twice for the right character (4 outcomes: $RW_1R, RW_2R, W_1RR, W_2RR$), and (3) vote twice for wrong characters and twice for the right one (6 outcomes: $RW_1W_2R, RW_2W_1R, W_1RW_2R, W_2RW_1R, W_1W_2RR, W_2W_1RR$). Hence, the probability of choosing one right character, c , within the voting system where $k = 2$ is:

$$P(c|k = 2) = p^2 + 4 \times p^2q + 6 \times p^2q^2 = 0.995, \quad (2)$$

where p and q denote the probability of choosing the right character, R , and wrong characters, W_1 or W_2 , respectively. Then, the probability of inputting 10 right characters in a row is as follows:

$$P(C|k = 2) = P(c|k = 2)^{10} = 0.950, \quad (3)$$

The multiple voting system with the smallest threshold $k = 2$ significantly increases the accuracy of MyoKey when typing sequences of characters. Simultaneously with the increase in input accuracy, input time increases as well. We discuss this trade-off in the following subsection.

3.3.2 Input time

Single Voting System: We first calculate the expected time of inputting one character within the voting system with $k = 1$ as a baseline. Let us put the window length w_l equal to 125 ms (approximately, 25 samples @ 200 Hz sampling rate), and inference time $\tau_i = 45$ ms (see 4). Thus estimated time equals to:

$$T_{k=1} = w_l + \tau_i = 170 \text{ ms} \quad (4)$$

Multiple Voting System: In the case of $k = 2$, we need 2, 3, or 4 time windows to input a character, as discussed

² We calculate the probability based on two assumptions for simplicity: (1) each trial is independent and (2) a user selects the column correctly.

Algorithm 1: Mistyped character correction algorithm for the word prediction

```

Input: The typed characters  $C = \{c_1, c_2, \dots, c_n\}$ , The learned probabilistic model  $\mathcal{M}$ 
Output: The predicted words up to three candidates
1  $candidates \leftarrow \mathcal{M}.predict(C)$ 
2 if  $candidates \neq \emptyset$  then
3   return  $Top3(candidates)$ 
4 else
5   for  $i \leftarrow n$  to 1 do
6      $neighbor\_character\_list \leftarrow FindNeighborCharacters(c_i)$ 
7     for  $neighbor\_character \in neighbor\_character\_list$  do
8        $C' \leftarrow Swap(neighbor\_character, c_i)$  in  $C$ 
9        $candidates \leftarrow \mathcal{M}.predict(C')$ 
10      if  $candidates \neq \emptyset$  then return  $Top3(candidates)$ 
11   end
12 end
13 end

```

in the previous subsection. Given the window overlap w_o equal to 62 *ms* (50% of the window), time required to have two classification results t_2 is $(2w_l - w_o + \tau_i)$; 3 results, $(t_3 = 3w_l - 2w_o + \tau_i)$ and four, $t_4 = (4w_l - 3w_o + \tau_i)$:³

$$T_{k=2} = p^2 * t_2 + (4 \times p^2 q) t_3 + (6 \times p^2 q^2) t_4 = 238.757 \text{ ms} \quad (5)$$

Assuming that the delay less than 300 *ms* is considered acceptable for continuous classification in real-life applications [65], our voting system with $k = 2$ enables continuous classification with high precision accurately for sequences of characters (95% for 10 characters) within acceptable time window (≈ 240 ms). Furthermore, MyoKey can further reduce the input time by adopting bigger window overlap sizes and smaller time windows with an expense of precision (see 4.2).

3.4 Further Optimizations

We now present further optimization techniques to improve typing speed and decrease error rates. First, we employ the probabilistic approach to predict the word that a user is currently typing. For example, when a user wants to input "weather" and has input 'w' and 'e' characters, the probabilistic model of MyoKey tries to predict up to the three most probable words such as "we," "were," and "well."

There exist many text entry studies [14], [44], [48] which implemented word prediction or word disambiguation models. However, their models, including the probabilistic model in our first optimization, suffer severely when a user inputs wrong characters while typing the current word. For instance, if a user types "weqt" instead of "weat" (i.e. mistypes 'q' instead of 'a'), the probabilistic model fails to predict the "weather". Thus, as our second optimization, we develop an error correction algorithm for incorrectly typed characters while a user is inputting the text in real-time. The details of both of our further optimizations are as follows:

Probabilistic model for word prediction: We leverage the probabilistic word prediction model for user input, which relies on the unigram language model. The basic idea is

3. The assumption is that classification is done in parallel and in a time shorter than the length of the time window, thus for multiple classifications in a row, inference time is counted only once.

to calculate the posterior probability of all words in a pre-trained language model given user inputs (i.e. few characters of the current word being typed by the user). After that, based on the Bayes' theorem, we compute the probability of a complete word (e.g. "weather") from an incomplete word of a few characters (e.g. "wea") and recommend the top-3 most probable words. Given the typed characters $C = \{c_1, c_2, \dots, c_n\}$, we search for the best and complete word with various length ($W_{best} = \{c_1, c_2, \dots, c_n, \dots, c_N\}$) within the lexicon, L , based on a corpus [66].

$$W_{best} = \underset{W \in L}{\arg \max} P(W|C) \quad (6)$$

According to the Bayes' rule, we have:

$$W_{best} = \underset{W \in L}{\arg \max} \frac{P(C|W)P(W)}{P(C)} \quad (7)$$

where $P(C)$ is constant for all candidates. $P(W)$ is the word frequency derived from the language model. Instead of using the spatial model as used in [60], we approximate $P(C|W)$ by checking the candidate words from the highest frequency and including the words that start with the typed characters.

Mistyped character correction: As our second optimization approach to improve our system's accuracy, we develop a lightweight error correction algorithm. For example, a user can mistype a character 'q' instead of 'a' in the target word 'weather'. By utilizing the error correction algorithm, MyoKey can still recommend the right candidate word ("weather") to the user. Algorithm 1 describes the overall process of the mistyped character correction algorithm for word prediction in detail. Given the typed characters $C = \{c_1, c_2, \dots, c_n\}$ which may have a mistyped character in any position from 1 to n and a learned probabilistic model \mathcal{M} , the goal is to find the most probable word candidates (we predict up to three candidates based on our interface design). For the first step, we search potential candidate words using the learned probabilistic model \mathcal{M} given the user input C (see line 1). If candidate words exist from the model, the algorithm returns the top three candidate words (see lines 2-3). In the case when the user input C may contain a mistyped character, the probabilistic model fails and predicts no candidate words (see line 4). Then, while we search through the characters from reverse order (from c_n to c_1), we predict the candidate words again after replacing the character c_i with its neighboring characters (see lines 5-12). The algorithm returns the found candidates up to three words if they exist. Note that we consider neighboring characters located in the same column in the QWERTY keyboard since the neighboring characters in the same column are most likely mistyped (e.g. neighboring characters of 'a' are 'q' and 'z').

Overall, integrating data from sEMG and IMU sensors, our system consolidates positional data (column selection) and recognized gestures (character within a column).

3.5 Recalibration Mechanism

We have established various heuristics and optimizations to ensure the online gesture recognition performance of MyoKey for a slight variation of sEMG signals of each participant. However, it is well-known that gesture recognition

TABLE 2: Order of scenarios for the participants

ID	L/R handed	Scenario 1	Scenario 2	Scenario 3
1	Right	Freehand	Tripod	Cylindrical
2	Right	Freehand	Cylindrical	Tripod
3	Right	Tripod	Freehand	Cylindrical
4	Right	Tripod	Cylindrical	Freehand
5	Right	Cylindrical	Freehand	Tripod
6	Left	Cylindrical	Tripod	Freehand
7	Left	Freehand	Tripod	Cylindrical
8	Right	Freehand	Cylindrical	Tripod
9	Right	Tripod	Freehand	Cylindrical
10	Right	Tripod	Cylindrical	Freehand
11	Left	Cylindrical	Freehand	Tripod
12	Left	Cylindrical	Tripod	Freehand

models using sEMG suffer from an additional variation of data derived from different recording sessions. For example, a user takes MyoKey off and wears it again later, the trained gesture recognition model’s performance may drop significantly [30], [67]. One way to address this issue is to collect new batches of labeled data before every session; the downside of this approach is the burden for users.

To overcome the session dependency, we introduce a recalibration mechanism to ensure reasonably high performance of the gesture recognition model. Our mechanism employs transfer learning [68] and captures user feedback (i.e., user’s selection of recommended words), which avoids cumbersome data collection procedures before a new session. First of all, we adopt the idea of transfer learning that fine-tunes the pre-trained model with new input data because it fits well with our scenario of a new recording session. In detail, we sample the small number of sEMG signals of a user when the user puts MyoKey on again and use them to fine-tune the pre-trained gesture recognition model for the participant. Then, to determine which sEMG signals to sample and identify the signals’ labels, we utilize the user feedback on recommended words from our system. For instance, a user intends to input “develop”. Then, after the user input the characters ‘d’, ‘e’, and ‘v’, MyoKey would correctly recommend the word “develop” so that the user can click on this word for the next input word. Using this user feedback, we obtain new input data of sEMG signals attached with ground-truth labels to fine-tune the pre-trained gesture recognition model in new recording sessions. Note that when a user wears MyoKey again, the user does not need to go through another data collection procedure to update the gesture recognition model since MyoKey uses the user’s usage patterns as new inputs and labels.

4 SYSTEM EVALUATION

We recruited 12 participants for the study (Age 23 – 33, all males, four left-handed). For right-handed, the band was placed on the right forearm; for left-handed - on the left. The surface of the band, including electrodes, was disinfected before every experiment for each participant. The user interface, keyboard, and experiment control elements were displayed on a 15-inch screen, placed on table 1-1.5m away from a participant [42], [69].

In our experimental setup, the testing interface, as presented in Figure 4, displays the target text, which is shown in light gray color, and the typed characters (i.e. inputs by user) overlays above the already input text in black

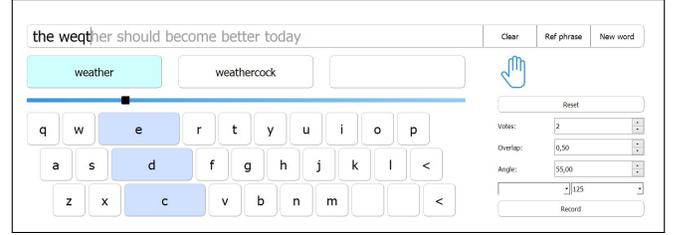


Fig. 4: The text entry layout of MyoBoard

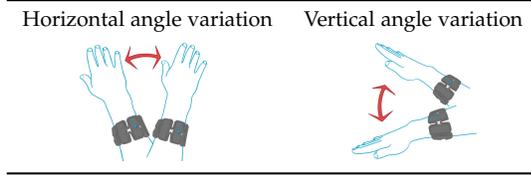
color. The word phrases are extracted from Mackenzie’s phrase set [70] in all experimental sessions. As depicted in Figure 1, the right part of MyoKey contains settings and control elements to run evaluative experiments. The top region of the button trios allows the experiment conductor to clear the user’s input, and pick a new target phrase or word from the phrase set. Below, the currently recognized gesture is depicted for the reference of an experiment participant, among with tick-boxes that enable/disable suggestion prediction. Next, the control elements allow the experiment conductor to reset the position of the band that corresponds to selection of the most right (last) column: utilized hardware and software stack does not implement measures against IMU drift [71] and as long as we expect text input to take significant time in some scenarios, the drift can cause inconvenience and irritation from shifting input angle. Downward, control elements are used to set the number of votes, overlap in decimal format, and selection angle range. The lowest two drop-down lists allow choosing the employed deep learning model and interval, offering additional flexibility on recognition time and accuracy.

Participants were seated in a chair with armrests and were required to rest the elbow of the active hand on the corresponding armrest. For the cylindrical grasp scenario, we used a tennis racket that was grasped in an umbrella fashion (see Table 1) in order to report exact weight (260g) and not obstruct the vision. The table surface, as well as the tennis racket’s handle, was disinfected in between experiments for different participants.

Software components were implemented using Python 3.7.5, Tensorflow 2.0.0 GPU (CUDA 9.2.148); inference time for time window of 100 samples that is used in estimating time in Section 3.3.2 (45.629 ± 3.606 ms) was measured on a laptop with 16 GB RAM, 4 cores @ 2.8 GHz and GeForce(R) GTX 1050 with 2 GB VRAM. Additionally, we convert our deep learning model for deployment on a mobile device to show the feasibility of a ubiquitous application [72], [73]. We use TensorFlow-lite-GPU application with GPUDelegate⁴ enabled. The developed application was further deployed on Samsung Galaxy A51 (SM-A5115F), running on Octa-Core 64bit Exynos 9 Octa 9611 (10nm) processor. We report inference time on the mobile device as following: for the window time of 100 samples - 8.560 ± 0.92 ms; for 50 samples - 4.609 ± 1.11 ms; for 25 - 3.487 ± 1.14 ms; for 10 samples - 2.684 ± 1.30 ms. Thus, the proposed solution is capable of running on up-to-date mobile devices with insignificant inference time. The CNN classifier does not require any preprocessing and Myo armband filters sEMG

4. <https://www.tensorflow.org/lite/performance/gpu>

TABLE 3: Spatial variations during the recording phase



on-board (notch filter 50/60 Hz, low-pass filter 20 Hz, high pass at 500 Hz).

4.1 Procedures

Each participant ran over a total of three sessions for all three scenarios (freehand, tripod grasp, and cylindrical grasp). Each scenario includes three major phases: data recording, training of the model, and text entry evaluation. We employ a full counterbalancing strategy [74], in order to reduce the carryover effects threatening to undermine our findings. That is, without full counterbalancing, the latter sessions may own advantages from the learning effects acquired in earlier sessions. We have two groups of six people, and within each group the order of the online experiments is different. Table 2 shows the full balancing of sessions employed in our experiment with 12 participants.

During the recording phase, sEMG signal for different gestures is collected: each participant was asked to perform and hold each gesture for the current scenario (see Table 1) for 30 seconds. Within each recording, the participants were asked to move their arm with a fixed gesture along the horizontal trajectory and to vary vertical angle as shown in Table 3. This is done in order to capture multiple variations of sEMG noise that can be generated by other muscles controlling the position of an arm and not related to the gesture itself [28].

After the training set is collected, the model is trained. The procedure takes less than a minute, in a meantime, a short demonstration of text input within a scenario is done. Next, the first trial of text input is attempted. Participants were given around 2-3 minutes to get used to the system. Based on the initial feedback, the yaw angle is adjusted (from 45 to 90 degrees) according to the participant's preference on an amplitude of movements and precision of column selection; additionally, the window overlap is tuned (from 50% up to 90%) if the selection of a character after performing a disambiguation gesture was too fast from the participant's point of view.

When a participant reports that they are used to the input protocol, the first text input session starts. Similarly to [75], within each session 5 random target 5-letter words from Mackenzie's phrase set are presented in the input field consecutively [14]. The time required to input a single word is calculated as a difference between the input of the first and the last character. After each session, a short summary of achieved wpm and error rate is presented to the experiment conductor (not to the participant to avoid bias in the further NASA Task Load Index questionnaire). Initially, we run 5 sessions. If the observed wpm during the first 5 sessions kept growing, we run an additional 6th

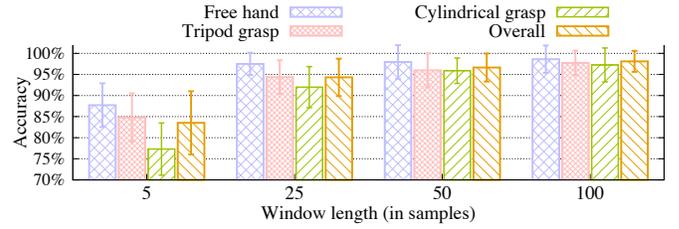


Fig. 5: Accuracy for static gesture recognition given the window size (in samples).

TABLE 4: Confusion matrices for offline gesture recognition, window size = 50 samples

	Freehand (96.6%)					Tripod (95.9%)					Cylindrical (95.8%)				
s	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
t	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
m	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
b	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
r	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
	s	t	m	b	r	s	t	m	b	r	s	t	m	b	r

Legend

■ [95%, 100%]	■ [90%, 95%]	■ (3%, 7%)	■ [1%, 3%]	■ (0%, 1%)
---------------	--------------	------------	------------	------------

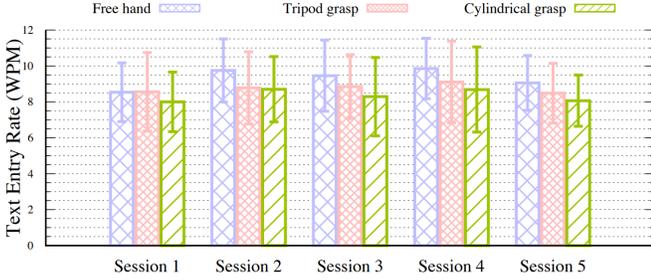
session to determine the knee point in the learning curve. In other words, we use the 6th session as an indicator that the peak performance was reached within 5 sessions.

4.2 Offline gesture recognition

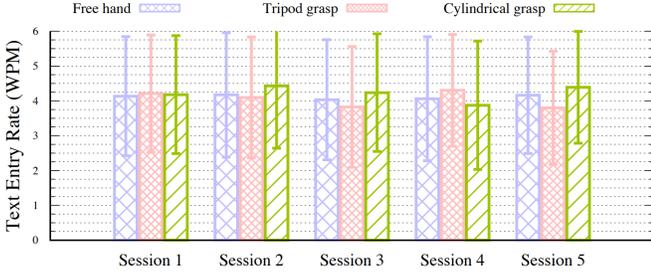
Here, we present the static recognition accuracy of the developed CNN classifier. We use 5-fold cross-validation to analyse model's performance given different lengths of time window for (a) overall performance and performance on (b) freehand, (c) tripod grasp, and (d) cylindrical grasp gesture sets separately. Based on the results presented in Figure 5, it is clear that the smaller the time window we employ, the less accurate the recognition is. We also show confusion matrices for offline gesture classification in Figure 4.

4.3 Text Entry Rate

Figure 6a shows the word-level text entry rate with MyoKey for freehand, tripod and cylindrical grasp scenarios. The standard deviation is represented by error bars. Two-way Repeated Measures (RM)-ANOVA demonstrates a significant effect of the conditions ($F_{2,165} = 4.128$, $p = 0.0178$) but an absence of statistical significance in the Session ($F_{4,165} = 1.301$, $p = 0.2717$), which indicates no obvious learning effect on the conditions throughout the five sessions. The participants in the freehand scenario achieved overall mean text entry rate of 9.33 WPM ($\sigma_F = 1.73$), which is 6.57% and 11.78% faster than the sessions of holding a pen ($\bar{M}_P = 8.76$ WPM, $\sigma_P = 1.94$) and in cylindrical grasp scenarios ($\bar{M}_U = 8.35$ WPM, $\sigma_U = 1.87$). We observe slight improvements in the text entry rates in all the scenarios between the 1st and 4th sessions, as follows. In the 1st session, the participants with freehand, tripod and cylindrical grasps result in 8.54 WPM ($\sigma_{f,1} = 1.63$), 8.56 WPM ($\sigma_{p,1} = 2.19$) and 8.00 WPM ($\sigma_{u,1} = 1.66$), respectively. During the 4th session, all scenarios coincidentally reach the peak performances: freehand ($\bar{M}_{f,4} = 9.86$ WPM, $\sigma_{f,4} = 1.68$), tripod



(a) WPM using suggestions.

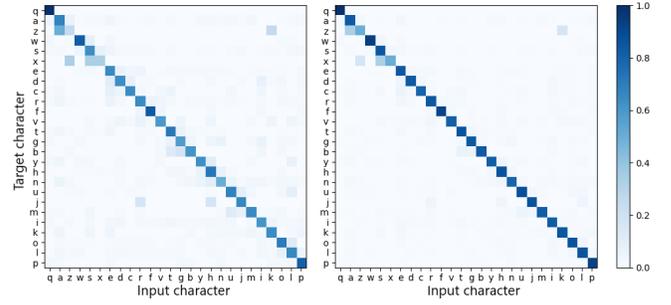


(b) WPM without suggestions.

Fig. 6: Text entry rate of MyoKey in three setups.

grasp ($\bar{M}_{p,4}=9.11$ WPM, $\sigma_{p,4} = 2.28$), and cylindrical grasp ($\bar{M}_{u,4}= 8.69$ WPM, $\sigma_{u,4} = 2.37$). However, we recorded consistently lower performances in the fifth session among all conditions, as the participants reported the tiredness of arms after four consecutive sessions (breaks included). Figure 6b reports the character-level text entry rate in the three scenarios. Throughout the five sessions, the participants maintain consistent throughput in three conditions, where the average text entry rates for freehand, tripod, and cylindrical grasp are 4.11 WPM ($\sigma_F = 1.72$), 4.05 WPM ($\sigma_P = 1.68$), 4.22 WPM ($\sigma_U = 1.73$). We also recorded the actual number of characters being typed before the participants had chosen the suggestion words by performing the Suggestion gesture (Table 1). All scenarios of freehand, tripod, and cylindrical grasps consistently demonstrate an average character number of 2.85 ($\sigma = 0.15$), 2.97 ($\sigma = 0.16$), and 2.92 ($\sigma = 0.04$), respectively.

Instead of typing the word phrases character by character, the participants usually type no more than three characters and seek the desired suggested word. This indicates that the participants rely on the suggested word list to accomplish the word phrases quickly. Also, we notice that an obvious performance gap between word-level and character-level text entry rates. With the optimized dictionary model (Section 3.4), the text entry rates in all three conditions increase by 227% (freehand), 216% (tripod grasp) and 198% (cylindrical grasp). The drastic speed improvements imply that the dictionary model proposed in MyoKey is a necessity for sEMG-based text inputs to reach usable text entry rates. As shown in Figure 6 input rate with suggestions reaches its maximum at sessions 3 and 4. After that, because of accumulated fatigue (reported as physical load in NASA TLX, see Section 4.7), the input rate decreases on average at session 5.



(a) Raw character input, ac- (b) Character accuracy after curacy = 0.666. suggestions = 0.851.

Fig. 7: Confusion matrices for input characters.

TABLE 5: Observed errors.

Entries	2956	Types of Errors			
Errors	985	I	326 (33.0%)	II	107 (10.8%)
Rate	33.4%	III	159 (16.1%)	IV	393 (39.8%)

4.4 Character level error rate

Before we present the word-level error rate in Section 4.5, we depict the **root causes** of erroneous inputs by examining the character-level errors and their types. It is important to note that the overall error rate of 33.4% (Table 5) is significantly alleviated by our voting system and corresponding optimization, resulting in the aforementioned stated entry rate (Section 4.3) at a usable level of input accuracy (Section 4.5). Also, it is worth pinpointing that text input with alternative input modals, i.e., wearable input techniques but not touchscreen-based inputs, are evaluated by word phrases. Meanwhile, the effectiveness of such input techniques is primarily judged by word-level performance [42], [76], [77]. The section explains the root causes for the sake of the transparency of our results. Figure 7 shows the confusion matrices between the intended character and produced character for (a) raw character input and (b) after the word suggestion is applied. Accordingly, we identify four types of character-level errors within our system:

Type I: A user selects the right column (i.e. the forearm locates at the target column), yet performs unrecognized gesture (due to limited familiarity with the system’s protocol) or the classifier misclassifies the gesture – *wrong* gesture within the *right* column. Such an error can be observed in clear line-shaped misclassification pattern in $\{e, d, c\}$ column as it is shown in Figure 6a. Another example of that type of error is the confusion of characters that are located in the two leftmost columns ($\{a, z, s, x\}$). When a user performs gestures to input those characters, they need to position their forearm closer to the chest, which causes the forearm muscles to contract more. Thus, the recorded sEMG signal is intensified and becomes less distinguishable to the developed classifier. Although the participants were asked to move their arm while collecting the training dataset as described in §4.1, the sEMG signals in that particular case still may not be captured properly for the training phase.

Type II: A user mistakenly chooses a character in an

adjacent column by initially selecting neighboring column or accidentally shifting column selection while performing a gesture for character disambiguation: *wrong*, yet adjacent column, *right* gesture e.g. characters 'j' and 'h'.

Type III: A user performs right gesture to input a desired character, but starts too early, in advance, before changing the column: *wrong*, and distant column, *right* gesture e.g. characters 'h' and 'a'.

Type IV: Wrong inputs performed in random columns while occasionally pausing to look for a target character. It can be a case when a user inputs a wrong character in a previously selected column and then moves on to the next target character. This type of errors can be easily observed in underrepresented characters, like 'z': "pizza" is being input as "pikza" due to phantom input of 'k' as it is in the same column with the previous character 'i'.

We present numbers for each type of the identified errors in Table 5. Individual physiological traits and the uniqueness of the bio-signal in each participant combined with the non-linearity of the deep learning model (and subsequent non-linearity of classification errors) lead to the observed complexity of the text error rate distribution. We calculate character accuracy presented in the confusion matrix based on raw input and target characters. Nevertheless, we only consider character-level predictions as a reference performance. That is, once the very first part of an intended word is incorrectly entered, our model of word suggestion corrects the word.

4.5 Word-level Error Rate

Figure 8 shows the error rate of the three MyoKey usage scenarios, and the error bars representing the standard deviation. Throughout the five sessions, the mean error rates of the three conditions (Freehand, Tripod and Cylindrical grasps) are 6.88% ($\sigma_F = 0.092$), 7.54% ($\sigma_P = 0.096$) and 9.81% ($\sigma_U = 0.121$) respectively. We ran a two-way RM-ANOVA indicating that Freehand, Tripod and Cylindrical grasps do not result in being statistically different in terms of the error rate ($F_{2,165} = 1.281$, $p = 0.2804$) and the learning effect among sessions ($F_{4,165} = 0.871$, $p = 0.482$). However, we spot reduced error rate between the initial and the peak performance. Initially, the scenarios of Freehand, Tripod and Cylindrical grasps generate error rates of 9.67% ($\sigma_{f,1} = 0.104$), 8.67% ($\sigma_{p,1} = 0.137$) and 9.71% ($\sigma_{u,1} = 0.133$), while their error rate reached the minimum values in either the 4th and 5th sessions – Freehand ($\bar{M}_{f,5} = 3.67\%$, $\sigma_{f,5} = 0.077$), Tripod grasp ($\bar{M}_{p,4} = 5.00\%$, $\sigma_{p,4} = 0.093$), and Cylindrical grasp ($\bar{M}_{u,4} = 7.71\%$, $\sigma_{u,4} = 0.099$).

4.6 Recalibration Performance

Additional Data Collection. To experimentally demonstrate the performance drop of the gesture recognition model on a new session data and its recovery based on our recalibration mechanism in our system, we first collected additional user data. Then, we asked the six participants among the 12 participants to perform another recording session. We performed the new recording session (at least) 12 after the initial data recording date, allowing us to investigate the effectiveness of our recalibration mechanism on user

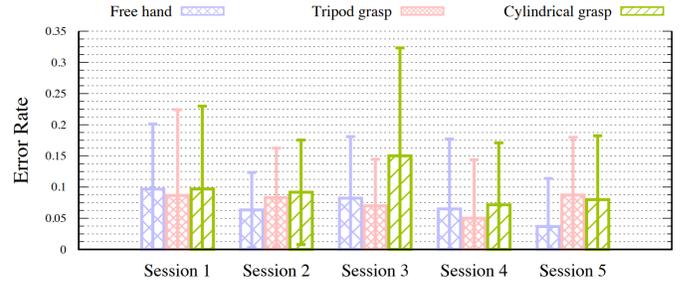


Fig. 8: Error Rate

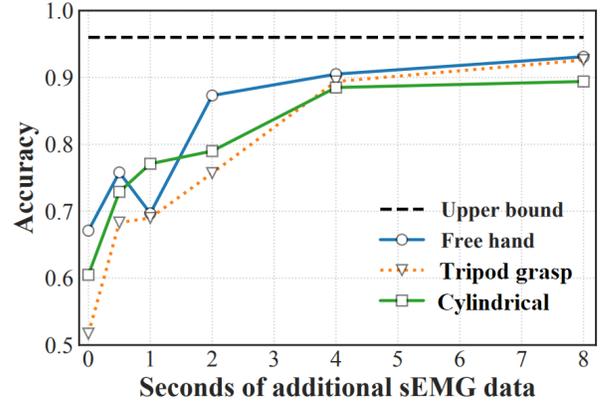


Fig. 9: Gesture recognition performance after applying the recalibration mechanism with different amounts of additional sEMG data. Upper bound represents the performance of a model trained from scratch using the entire new data.

behavior changes over a long period. The data collection process is the same as described in Sections 4.1 and 4.2.

Results. Figure 9 presents the recognition performance of MyoKey when [0.5, 1, 2, 4, 8] seconds of each gesture from a user is added for updating the model. To begin with, as we mentioned in Section 3.5, we have observed that the performance of the gesture recognition model decreases sharply from 96.6%, 95.9%, 95.8% to 67.1%, 51.7%, 60.5% for all three conditions (Freehand, Tripod, and Cylindrical grasps) respectively. The results indicate that the more user data is added to re-train the model, the higher accuracy can be achieved. In particular, using two seconds of additional data for each gesture, our recalibration mechanism demonstrates that it can achieve high accuracy of 79-87.3%. Furthermore, as shown in Figure 9, after obtaining the four-second additional sEMG signals for each gesture from user feedback, the performance converges to 88.5-90.5%, which is close to the upper bound model that is trained from scratch using the entire new data. Note that our recalibration mechanism happens once when a user wears MyoKey again and starts to use it. This result indicates that our recalibration mechanism effectively maintains the high accuracy of the gesture recognition model. Also, since we can obtain approximately one second of the user’s sEMG signals when the user clicks on a character and a set of corresponding labels when the user selects a recommended word, the performance of MyoKey can be recovered promptly within a few selections of recommended words.

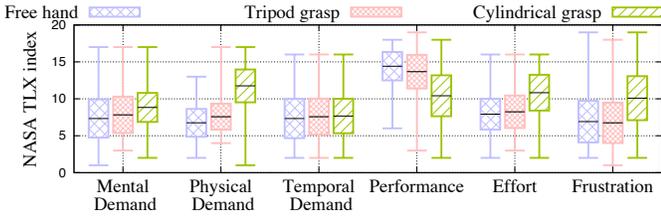


Fig. 10: User Workload measured via the NASA Task Load Index questionnaire: the line within box plot – mean value, and the circular dot outside the box – outlier.

4.7 NASA Task Load Index

Figure 10 depicts the user workloads in the three scenarios, which is quantified on a scale from 1 to 20. For *Mental*, *Physical*, *Temporal*, *Frustration* and *Effort*, the lower the score the higher the user acceptance; for *Performance* metric it is opposite: higher score reflects bigger satisfaction with the achieved result.

Among the three scenarios, one-way ANOVA shows that no statistical significance in *Mental* ($F(2, 33) = 0.3155$, $p = 0.732$), *Temporal* ($F(2, 33) = 0.015$, $p = 0.985$), *Performance* ($F(2, 33) = 2.457$, $p = 0.101$), *Effort* ($F(2, 33) = 1.518$, $p = 0.234$), and *Frustration* ($F(2, 33) = 1.303$, $p = 0.285$), except *Physical* ($F(2, 33) = 5.556$, $p = 0.008$), or to put it simple, the participants did not notice the difference in the conditions of all three scenarios in terms of the above metrics, only physical demand was clearly higher for the Cylindrical grasp (heavier object) scenario. Furthermore, we ran the Tukey HSD Post-hoc Test for the Physical metric that demonstrates a significant difference in the group of Freehand and Cylindrical grasp ($p=0.0104$), as well as the group of Tripod and Cylindrical grasp ($p=0.0366$), but no difference among Freehand and Tripod grasp ($p = 0.8629$). In comparison to performing gestural inputs in the Freehand scenario ($\bar{M}= 6.75$), the lightweight pen does not make a different burden ($\bar{M}= 7.58$). Therefore, the participants report that holding the tennis racket (260 grams) for forearm movements in mid-air leads to a significantly higher workload ($\bar{M}= 11.75$).

5 DISCUSSION

One of the grand challenges for researchers on HCI and more specifically on text inputs is to move beyond the text entry on desktop computers and smartphones and to start exploring novel and emerging sensory technologies for alternative contextual uses [32], [78]. When we consider the text entry problems on the head-worn computers such as mobile augmented reality headsets, we usually encounter limited physical form factor [42], and consequently, the diminishing input interfaces (i.e. the missing of touch-screens) [14]. Gaze-based interaction requires no hand in the text entry process but it requires obtrusive and non-mobile sensors. For instance, EyeSwipe (11.7 WPM) [79] is supported by the Tobii EyeX eye tracker that violates the form factor restriction on mobile augmented reality headsets. Most recent works demonstrate the sEMG sensing capabilities in wearable computers, for instance, gestural controls (e.g. select and swipe) on Google Glass [80], notification

management [81], enriching the meaning of user touch on touch-sensitive surfaces on smart tangible objects [82], as well as recognizing the physical objects on a hand [34].

MyoKey serves as a groundwork leveraging the EMG sensors for text entry tasks on wearable computers achieving high usability and user acceptance. The prior works propose the non-standard keyboards [36] such as LURD-Writer [61], and users need to learn the new text entry layouts, in addition to the burden from practicing new gestures driven by sEMG sensors. Instead, MyoKey leverages the user’s ingrained memory of the standard QWERTY keyboard layouts [14], in which the familiar layout can significantly reduce the learning efforts and hence improve usability [60].

The goal behind the design of MyoKey is to implement a text entry system on headset computers while holding objects. In general, MyoKey can serve all headset users with occupied/busy hands. More specifically, we see that the most immediate demand for MyoKey belongs to workers in the industrial sector. For instance, a worker in a factory or a warehouse is holding a tool, and simultaneously intends to perform text entry. Another example is that office personnel may grab a document and at the same time perform text entry. Although the state-of-the-art two-handed text entry solutions leveraging within-finger gestures achieve significantly higher text entry rates (Mid-air: 12.5 WPM [49] and Touch-based: 16 WPM [83] – 23.4 WPM [84]) than the one-handed solutions, we have the concerns about the case when both hands are occupied in outdoor environments. To this end, we compare MyoKey with other most relevant Same and Single-Handed (SSH) text entry systems. SSH is highly characterized by the one-handed operations with subtle gestures *within the finger space* [23]. Additionally, the latest works on SSH own high mobility, as another empty hand is reserved for other potential tasks in outdoor environments, for instance, holding a handrail [42]. Certain SSH input systems, leveraging two-handed operations, outperform MyoKey in terms of text entry speed. For instance, a two-handed SSH text entry solution, named DigiTouch [83], achieves a mean text entry rate of 13.0 WPM. However, DigiTouch involves two simultaneous input channels driven by two thumb operations on their respective finger space. The two-handed SSH solutions cannot serve as a fair comparison to MyoKey with only a single finger space, not to mention the drawbacks of two occupied hands.

FingerT9 [23] is a one-handed glove for text entry tasks. The user’s thumb performs multi-taps on a touch-sensitive ambiguous T9 layout located on the index, middle, and ring fingers (finger space). The thumb-to-finger interaction with FingerT9 achieves a mean text entry rate of 3.43 WPM. The primary reasons for the low text entry rate are as follows. First, the users spent three days memorizing the alphabetical layout on FingerT9. Second, the thumb movements are less dexterous than other fingers, especially the index finger [85], and thus the multi-tap operations on the T9 layout become time-consuming and less efficient. Another glove-based text entry system [42] addresses the thumb dexterity issue by substituting the multi-tap input by force-assisted interaction, which results in a mean text entry rate of 5.12 WPM. The above two solutions need freeing up the finger space of the operating hand, while MyoKey

can accomplish the text entry with a busy hand. We notice that an SSH solution named RotoSwipe [86] achieved 14 WPM after 5-day training. It involves repetitive clicks on a touch interface on the ring-form addendum, in addition to frequent wrist rotations that translate to the x-y movement of a pointer on the swipe-based keyboard of RotoSwipe. However, the gesture design (i.e., rotational movements) may not be applicable to our scenario of holding an object. Additionally, a recent work leveraging the sEMG sensors for one-handed text entry achieves only 2.56 WPM [36] in mobile scenarios. To the best of our knowledge, MyoKey (8.35 – 9.33 WPM) uniquely covers both free-hand and busy hand text entry leveraging the property of SSH text entry solutions, that is, subtle interactions and reserving at least one hand for physical tasks.

Moreover, we acknowledge other touch-based interactions designated for one-handed text entry on various touch-sensitive surfaces such as the spectacle frame of Google Glass (One-Dimensional Handwriting (4.67 WPM), SwipeZone [6] (8.73 WPM), a finger-worn addendum (8 WPM) [87], and fingertip addendum (11.9 WPM) [76]. The text entry rate of MyoKey is comparable to the above state-of-the-art solutions but only slightly inferior to the fingertip addendum, with the below reasons. In MyoKey, subtle electric currents in each muscle-contraction gesture involve an unavoidable dwell time of ≈ 300 ms, which is less responsive than the touch-based interaction [15] on the above devices. Also, the horizontal arm movements in the MyoKey layout lead to higher overheads in character selections than the subtle movements within an area of one fingertip on the fingertip addendum. MyoKey allows the user not only to perform freehand operations but also to maintain a firm grasp on an object. It is worthwhile to mention that MyoKey and the sets of micro-gestures [31], enable a user's hand to hold an object and another hand can be free during the text entry tasks. For instance, the users holding an umbrella can complete the text entry task through a series of subtle arm movements and micro-gestures during bad weather. Similarly, other everyday objects in a cylindrical shape (e.g., handrails in public transport) can be analogized by the third condition in our evaluation (cylindrical grasp), as long as secondary modality is different (e.g., gaze tracking). In contrast, both the operations on the spectacle frame of the smartglasses and the addendum devices require a lifting hand on the head position and rigorous finger movements on the addendum devices.

We have shown the feasibility of deployment on the conventional mobile phone, yet the deployment of our system on a smaller ubiquitous device, such as Google Glass, might be challenging since it has limited computing capabilities given the constraints on computational power, memory, and energy. It is important to investigate the practicality of deploying the learned deep CNN model for gesture recognition and our MyoKey system on a standalone XR device first hand. Yet, there exist various approaches to tackle this challenge based on previous works. Offloading [20], when sEMG signal is captured and pre-processed on a mobile device, but the actual deep learning recognition is performed on an external computing device, is one of them. Another one - is the utilization of a Digital Signal Processor (DSP) on commercial smartphones or wearables that enables

fast and efficient signal processing and consumes an order of magnitude less energy than an active WiFi chip and CPU [88], [89].

Grasp micro-gestures can be classified into six types including Cylindrical (e.g., umbrella), Palmar (e.g., book), Hook (e.g., Bag), Lateral (e.g., Paper), Tripod (e.g. Pen), and Spherical (e.g. ball) [31]. Therefore, wider coverage of grasp micro-gestures to the remaining four grasp types can be addressed. As a natural continuation of our work, detection of held objects can be implemented [34] in addition to the existing functionality. With this capability, MyoKey can intelligently swap the sets of grasp micro-gestures from one type to another. Also, the ability to incorporate new user gestures using continual learning [90], [91] would make MyoKey provide better system usability.

We acknowledge the limitation of our experimentation protocol, as participants were required to sit in a chair with the fixed elbow, thus one of the important future directions is to evaluate Myokey in walking and running postures. In addition, external aspects can affect the practicality of our system, such as muscle fatigue and sweats on the skin of a user's arm. Considering these aspects into the system design can be promising future work.

6 CONCLUSION

In this work, we introduced MyoKey, a text entry system for extended realities based on myoelectric signals and inertial measurements. MyoKey leverages from users' ingrained memory of the standard QWERTY keyboard. The deep-learning classifier and error correction techniques let Myokey achieve high usability and user acceptance. We have recruited twelve participants to experience MyoKey in three different scenarios of five or six sessions each. In one scenario, the hands of the participants were not occupied, while in the other two they were holding two types of everyday objects, one small (pen) and a bigger one (tennis racket in the conducted experiments). We established the applicability of three different micro-grasping gesture sets in different contexts to text inputs. MyoKey is evaluated in terms of words per minute, word error rate, and using the NASA task load index. Throughout the five sessions of text entry tasks, MyoKey achieves an averaged text entry rate of 9.33 words per minute in the scenario of the freehand. The participants with grasp gestures for a pen reached an averaged text entry rate of 8.76 words per minute. In the case of cylindrical grasp, they reached a comparable averaged text entry rate of 8.35 words per minute.

REFERENCES

- [1] "Google glass," <https://www.google.com/glass/start/>, (Accessed on 05/14/2020).
- [2] "Focals by north," <https://www.bynorth.com/>, (Accessed on 05/14/2020).
- [3] V. Reports, "Augmented reality and virtual reality (ar & vr) market size is expected to reach usd 571.42 billion by 2025," <https://www.prnewswire.com/news-releases/augmented-reality-and-virtual-reality-ar-vr-market-size-is-expected-to-reach-usd-571-42-billion-by-2025-valuation-reports-301004582.html>, (Accessed on 05/14/2020).
- [4] A. Kapur, S. Kapur, and P. Maes, "Alterego: A personalized wearable silent speech interface," in *23rd International Conference on Intelligent User Interfaces*, 2018, pp. 43–53.

- [5] J. Kim, G. Bennett, S. Fu, and J. Kruse, "Extended reality practice in art & design creative education," in *SIGGRAPH Asia Courses*, 2019, pp. 1–99.
- [6] F. Rabbi, T. Park, B. Fang, M. Zhang, and Y. Lee, "When virtual reality meets internet of things in the gym: Enabling immersive interactive machine exercises," *Proc. ACM IMMUT.*, vol. 2, no. 2, Jul. 2018.
- [7] A. Guo, I. Canberk, H. Murphy, A. Monroy-Hernández, and R. Vaish, "Blocks: Collaborative and persistent augmented reality experiences," *Proc. ACM IMMUT.*, vol. 3, no. 3, Sep. 2019.
- [8] D. Chatzopoulos, C. Bermejo, Z. Huang, A. Butabayeva, R. Zheng, M. Golkarifard, and P. Hui, "Hyperion: A wearable augmented reality system for text extraction and manipulation in the air," in *Proc. of the 8th ACM MMSys*, ser. MMSys'17, 2017, p. 284–295.
- [9] D. Chatzopoulos, C. Bermejo, Z. Huang, and P. Hui, "Mobile augmented reality survey: From where we are to where we go," *IEEE Access*, vol. 5, pp. 6917–6950, 2017.
- [10] C. Jang, K. Bang, S. Moon, J. Kim, S. Lee, and B. Lee, "Retinal 3d: augmented reality near-eye display via pupil-tracked light field projection on retina," *ACM TOG*, vol. 36, no. 6, pp. 1–13, 2017.
- [11] S. Sangu, T. Shimokawa, and S. Tanaka, "Ultracompact eye and pupil tracking device using vcsel arrays and position sensitive detector," in *Optical Architectures for Displays and Sensing in AR, VR, and MR*, vol. 11310, 2020, p. 113101F.
- [12] K. L. Li, "Understanding waveguide: the key technology for augmented reality near-eye display (part i)," <https://virtualrealitypop.com/understanding-waveguide-the-key-technology-for-augmented-reality-near-eye-display-part-i-2b16b61f4bae>, (Accessed on 05/14/2020).
- [13] Y.-H. Lee, K. Yin, and S.-T. Wu, "Reflective polarization volume gratings for high efficiency waveguide-coupling augmented reality displays," *Optics express*, vol. 25, no. 22, pp. 27008–27014, 2017.
- [14] L. H. Lee, K. Yung Lam, Y. P. Yau, T. Braud, and P. Hui, "Hibey: Hide the keyboard in augmented reality," in *IEEE PerCom*, 2019, pp. 1–10.
- [15] L. Lee and P. Hui, "Interaction methods for smart glasses: A survey," *IEEE Access*, vol. 6, pp. 28712–28732, 2018.
- [16] C. Yu, Y. Gu, Z. Yang, X. Yi, H. Luo, and Y. Shi, "Tap, dwell or gesture? exploring head-based text entry techniques for hmds," in *Proceedings of CHI*, 2017, p. 4479–4488.
- [17] Y.-T. Hsieh, A. Jylhä, V. Orso, L. Gamberini, and G. Jacucci, "Designing a willing-to-use-in-public hand gestural interaction technique for smart glasses," in *Proceedings of CHI*, 2016, p. 4203–4215.
- [18] J. Shu, R. Zheng, and P. Hui, "Your privacy is in your hand: Interactive visual privacy control with tags and gestures," in *International Conference on Communication Systems and Networks*, 2017, pp. 24–43.
- [19] X. Zhai, B. Jelfs, R. H. Chan, and C. Tin, "Self-recalibrating surface emg pattern recognition for neuroprosthesis control based on convolutional neural network," *Frontiers in neuroscience*, vol. 11, p. 379, 2017.
- [20] K. A. Shatilov, D. Chatzopoulos, A. W. T. Hang, and P. Hui, "Using Deep Learning and Mobile Offloading to Control a 3D-printed Prosthetic Hand," *Proc. ACM IMMUT.*, vol. 3, no. 3, pp. 102:1–102:19, Sep. 2019.
- [21] J. Ghillebert, S. De Bock, L. Flynn, J. Geeroms, B. Tassignon, B. Roelands, D. Lefeber, B. Vanderborght, R. Meeusen, and K. De Pauw, "Guidelines and recommendations to investigate the efficacy of a lower-limb prosthetic device: A systematic review," *IEEE Trans. on Medical Robotics and Bionics*, vol. 1, no. 4, pp. 279–296, 2019.
- [22] O. J. Quittmann, J. Meskemper, K. Albracht, T. Abel, T. Foitschik, and H. K. Strüder, "Normalising surface emg of ten upper-extremity muscles in handcycling: Manual resistance vs. sport-specific mvics," *Journal of Electromyography and Kinesiology*, vol. 51, p. 102402, 2020.
- [23] P. C. Wong, K. Zhu, and H. Fu, "Fingert9: Leveraging thumb-to-finger interaction for same-side-hand text entry on smartwatches," in *Proc. of CHI*, 2018.
- [24] "Welcome to myo support," <https://support.getmyo.com/hc/en-us>, (Accessed on 05/14/2020).
- [25] "Oymotion," <https://oymotion.github.io/>, (Accessed on 05/14/2020).
- [26] Y. Yang, S. Chae, J. Shim, and T.-D. Han, "Emg sensor-based two-hand smart watch interaction," in *Proceedings of ACM Symposium on User Interface Software & Technology*, 2015, p. 73–74.
- [27] Y. D. Kwon, K. A. Shatilov, L.-H. Lee, S. Kumyol, K.-Y. Lam, Y.-P. Yau, and P. Hui, "Myokey: Surface electromyography and inertial motion sensing-based text entry in AR," in *2020 IEEE PerCom Workshops*, Mar. 2020, pp. 1–4.
- [28] U. Côté-Allard, G. Gagnon-Turcotte, A. Phinyomark, K. Glette, E. Scheme, F. Laviolette, and B. Gosselin, "Virtual reality to study the gap between offline and real-time emg-based gesture recognition," *arXiv preprint arXiv:1912.09380*, 2019.
- [29] S. Pal, D. Juyal, S. Adekhandi, M. Sharma, R. Prakash, N. Sharma, A. Rana, and A. Parihar, "Mobile phones: Reservoirs for the transmission of nosocomial pathogens," *Advanced biomedical research*, vol. 4, 2015.
- [30] I. Ketykó, F. Kovács, and K. Z. Varga, "Domain Adaptation for sEMG-based Gesture Recognition with Recurrent Neural Networks," in *IJCNN*, Jul. 2019, pp. 1–7, ISSN: 2161-4407.
- [31] A. Sharma, J. S. Roo, and J. Steimle, "Grasping microgestures: Eliciting single-hand microgestures for handheld objects," in *Proc. of CHI*, 2019.
- [32] K. A. Shatilov, D. Chatzopoulos, L.-H. Lee, and P. Hui, "Emerging exg-based nui inputs in extended realities: A bottom-up survey," *ACM TiiS*, vol. 11, no. 2, pp. 1–49, 2021.
- [33] M. Cognolato, A. Gijssberts, V. Gregori, G. Saetta, K. Giacomino, A.-G. M. Hager, A. Gigli, D. Faccio, C. Tiengo, F. Bassetto *et al.*, "Gaze, visual, myoelectric, and inertial data of grasps for intelligent prosthetics," *Scientific Data*, vol. 7, no. 1, pp. 1–15, 2020.
- [34] J. Fan, X. Fan, F. Tian, Y. Li, Z. Liu, W. Sun, and H. Wang, "That in your hand?: Recognizing grasped objects via forearm electromyography sensing," *Proc. of ACM IMMUT.*, vol. 2, no. 4, p. 161, 2018.
- [35] B. Fan, X. Liu, X. Su, P. Hui, and J. Niu, "Emgauth: An emg-based smartphone unlocking system using siamese network," in *IEEE PerCom*. IEEE, 2020, pp. 1–10.
- [36] F. Wang, S. Cui, S. Yuan, J. Fan, W. Sun, and F. Tian, "Myotyper: A myo-based texting system for forearm amputees," in *Proceedings of the Sixth International Symposium of Chinese CHI*, 2018, pp. 144–147.
- [37] S. Zhai and P. O. Kristensson, "The word-gesture keyboard: Reimagining keyboard interaction," *Commun. ACM*, vol. 55, no. 9, p. 91–101, Sep. 2012.
- [38] L. H. Lee, T. Braud, F. H. Bijarbooneh, and P. Hui, "Tipoint: Detecting fingertip for mid-air interaction on computational resource constrained smartglasses," in *Proc. of ISWC*, 2019, p. 118–122.
- [39] K. Palin, A. M. Feit, S. Kim, P. O. Kristensson, and A. Oulasvirta, "How do people type on mobile devices? observations from a study with 37,000 volunteers," in *Proc. of MobileHCI*, 2019.
- [40] W. S. Walmsley, W. X. Snelgrove, and K. N. Truong, "Disambiguation of imprecise input with one-dimensional rotational text entry," *ACM Trans. Comput.-Hum. Interact.*, vol. 21, no. 1, Feb. 2014.
- [41] C.-Y. Wang, W.-C. Chu, P.-T. Chiu, M.-C. Hsiu, Y.-H. Chiang, and M. Y. Chen, "Palmtyping: Using palms as keyboards for smart glasses," in *Proc. of International Conference on Human-Computer Interaction with Mobile Devices and Services*, 2015, pp. 153–160.
- [42] L. H. Lee, K. Y. Lam, T. Li, T. Braud, X. Su, and P. Hui, "Quadmetric optimized thumb-to-finger interaction for force assisted one-handed text entry on mobile headsets," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 3, no. 3, Sep. 2019.
- [43] I. Hayet, T. F. Haq, H. Mahmud, and M. K. Hasan, "Designing a hierarchical keyboard layout for brain computer interface based text entry," in *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, Feb 2019, pp. 1–6.
- [44] M. Zhong, C. Yu, Q. Wang, X. Xu, and Y. Shi, "Forceboard: Subtle text entry leveraging pressure," in *Proc. of ACM CHI*, 2018.
- [45] L. H. Lee, Y. Zhu, Y. P. Yau, T. Braud, X. Su, and P. Hui, "One-thumb text acquisition on force-assisted miniature interfaces for mobile headsets," in *IEEE PerCom*, March 2020, pp. 1–10.
- [46] F. Kerber, M. Puhl, and A. Krüger, "User-independent real-time hand gesture recognition based on surface electromyography," in *Proceedings of MobileHCI*, 2017.
- [47] N. Yanagihara and B. Shizuki, "Cubic keyboard for virtual reality," in *Proceedings of the Symposium on Spatial User Interaction*, ser. SUI '18, 2018, p. 170.
- [48] F. C. Y. Li, R. T. Guy, K. Yatani, and K. N. Truong, "The 1line keyboard: A qwerty layout in a single line," in *Proc. of the 24th ACM Symposium on User Interface Software and Technology*, 2011, p. 461–470.
- [49] J. Fashimpaur, K. Kin, and M. Longest, "Pinchtype: Text entry for virtual and augmented reality using comfortable thumb to fingertip pinches," ser. ACM CHI Extended abstracts, 2020, p. 1–7.

- [50] J. Clawson, T. Starner, D. Kohlsdorf, D. P. Quigley, and S. Gilliland, "Texting while walking: An evaluation of mini-qwerty text input while on-the-go," in *Proc. of MobileHCI*, 2014, p. 339–348.
- [51] A. Yadav and A. S. Arif, "Effects of keyboard background on mobile text entry," in *Proc. of International Conference on Mobile and Ubiquitous Multimedia*, 2018, p. 109–114.
- [52] S. Mizobuchi, M. Chignell, and D. Newton, "Mobile text entry: Relationship between walking speed and text input task difficulty," in *Proc. of MobileHCI*, 2005, p. 122–128.
- [53] T. Mustonen, M. Olkkonen, and J. Hakkinen, "Examining mobile phone text legibility while walking," in *ACM CHI Extended Abstracts*, 2004, p. 1243–1246.
- [54] J. Orlosky, K. Kiyokawa, and H. Takemura, "Dynamic text management for see-through wearable and heads-up display systems," in *Proc. of International Conference on Intelligent User Interfaces*, 2013, p. 363–370.
- [55] E. M. Klose, N. A. Mack, J. Hegenberg, and L. Schmidt, "Text presentation for augmented reality applications in dual-task situations," in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, March 2019, pp. 636–644.
- [56] Y.-C. Lin, L. Hsu, and M. Y. Chen, "Peritextar: Utilizing peripheral vision for reading text on augmented reality smart glasses," in *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, ser. VRST '18, 2018.
- [57] I. Chaturvedi, F. H. Bijarbooneh, T. Braud, and P. Hui, "Peripheral vision: A new killer app for smart glasses," in *Proc. of the 24th International Conference on Intelligent User Interfaces*, 2019, p. 625–636.
- [58] Y. Matsuura, T. Terada, T. Aoki, S. Sonoda, N. Isoyama, and M. Tsukamoto, "Readability and legibility of fonts considering shakiness of head mounted displays," in *Proceedings of the 23rd International Symposium on Wearable Computers*, ser. ISWC '19, 2019, p. 150–159.
- [59] D. Huang, X. Zhang, T. S. Saponas, J. Fogarty, and S. Gollakota, "Leveraging dual-observable input for fine-grained thumb interaction using forearm emg," in *Proc. of ACM Symposium on User Interface Software & Technology*, 2015, pp. 523–528.
- [60] X. Bi, B. A. Smith, and S. Zhai, "Quasi-qwerty soft keyboard optimization," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10, 2010, p. 283–286.
- [61] T. Felzer and R. Nordmann, "Alternative text entry using different input methods," in *Proceedings of*, 2006, p. 10–17.
- [62] F. Matulic, B. Vogel, N. Kimura, and D. Vogel, "Eliciting pen-holding postures for general input with suitability for emg armband detection," in *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*, 2019, pp. 89–100.
- [63] E. Rahimian, S. Zabihi, S. F. Atashzar, A. Asif, and A. Mohammadi, "Xceptiontime: A novel deep architecture based on depthwise separable convolutions for hand gesture classification," *arXiv preprint arXiv:1911.03803*, 2019.
- [64] U. Côté-Allard, G. Gagnon-Turcotte, F. Laviolette, and B. Gosselin, "A low-cost, wireless, 3-d-printed custom armband for semg hand gesture recognition," *Sensors*, vol. 19, no. 12, p. 2811, 2019.
- [65] K. Englehart and B. Hudgins, "A robust, real-time control scheme for multifunction myoelectric control," *IEEE Transactions on Biomedical Engineering*, vol. 50, no. 7, pp. 848–854, Jul. 2003.
- [66] "Project Gutenberg," library Catalog: www.gutenberg.org/ [Online]. Available: <http://www.gutenberg.org/>
- [67] Y. Du, W. Jin, W. Wei, Y. Hu, and W. Geng, "Surface EMG-Based Inter-Session Gesture Recognition Enhanced by Deep Domain Adaptation," *Sensors*, vol. 17, no. 3, p. 458, Mar. 2017, number: 3 Publisher: Multidisciplinary Digital Publishing Institute.
- [68] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *Advances in Neural Information Processing Systems*, vol. 27, pp. 3320–3328, 2014.
- [69] Z. Yang, C. Yu, X. Yi, and Y. Shi, "Investigating gesture typing for indirect touch," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 3, no. 3, Sep. 2019.
- [70] I. S. MacKenzie and R. W. Soukoreff, "Phrase sets for evaluating text entry techniques," in *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '03, 2003, p. 754–755.
- [71] S. Li, Y. Gao, G. Meng, G. Wang, and L. Guan, "Accelerometer-based gyroscope drift compensation approach in a dual-axial stabilization platform," *Electronics*, vol. 8, no. 5, p. 594, 2019.
- [72] Y. D. Kwon, J. Chauhan, and C. Mascolo, "FastICARL: Fast Incremental Classifier and Representation Learning with Efficient Budget Allocation in Audio Sensing Applications," in *Proc. Interspeech 2021*, 2021, pp. 356–360.
- [73] S. S. Rodríguez, C. Mascolo, and Y. D. Kwon, "Knowing when we do not know: Bayesian continual learning for sensing-based analysis tasks," *arXiv preprint arXiv:2106.05872*, 2021.
- [74] I. S. MacKenzie, *Human-computer interaction: An empirical research perspective*. Newnes, 2012.
- [75] T. Grossman, X. A. Chen, and G. Fitzmaurice, "Typing on glasses: Adapting text entry to smart eyewear," in *Proc. of MobileHCI*, 2015, p. 144–152.
- [76] Z. Xu, P. C. Wong, J. Gong, T.-Y. Wu, A. S. Nittala, X. Bi, J. Steimle, H. Fu, K. Zhu, and X.-D. Yang, "Tiptext: Eyes-free text entry on a fingertip keyboard," in *Proceedings of the 32nd ACM Symposium on User Interface Software and Technology*, 2019, p. 883–899.
- [77] P. C. Wong, K. Zhu, and H. Fu, "Fingert9: Leveraging thumb-to-finger interaction for one-handed text entry on smartwatches," in *SIGGRAPH Asia Mobile Graphics Interactive Applications*, 2017.
- [78] J. Clawson, A. S. Arif, S. Brewster, M. Dunlop, P. O. Kristensson, and A. Oulasvirta, "Text entry on the edge," in *Proc. of ACM Conference Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '15, 2015, p. 2381–2384.
- [79] A. Kurauchi, W. Feng, A. Joshi, C. Morimoto, and M. Betke, "Eyeswipe: Dwell-free text entry using gaze paths," in *Proc. of ACM CHI*, 2016, p. 1952–1956.
- [80] D. Huang, X. Zhang, T. S. Saponas, J. Fogarty, and S. Gollakota, "Leveraging dual-observable input for fine-grained thumb interaction using forearm emg," in *Proc. of ACM Symposium on User Interface Software & Technology*, 2015, p. 523–528.
- [81] T. Duenke, J. Schulte, M. Pfeiffer, and M. Rohs, "Muscleio: Muscle-based input and output for casual notifications," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 2, Jul. 2018.
- [82] V. Becker, P. Oldrati, L. Barrios, and G. Sörös, "Touchsense: Classifying and measuring the force of finger touches with an electromyography armband," in *Proceedings of the 9th Augmented Human International Conference*, ser. AH '18, 2018.
- [83] E. Whitmire, M. Jain, D. Jain, G. Nelson, R. Karkar, S. Patel, and M. Goel, "Digitouch: Reconfigurable thumb-to-finger input and text entry on head-mounted displays," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 3, Sep. 2017.
- [84] Z. Xu, W. Chen, D. Zhao, J. Luo, T.-Y. Wu, J. Gong, S. Yin, J. Zhai, and X.-D. Yang, "Bitiptext: Bimanual eyes-free text entry on a fingertip keyboard," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI '20, 2020, p. 1–13.
- [85] S. Azenkot and S. Zhai, "Touch behavior with different postures on soft smartphone keyboards," in *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services*, ser. MobileHCI '12. ACM, 2012, pp. 251–260.
- [86] A. Gupta, C. Ji, H.-S. Yeo, A. Quigley, and D. Vogel, "Rotoswype: Word-gesture typing using a ring," in *Proc. of ACM CHI Conference on Human Factors in Computing Systems*, 2019, p. 1–12.
- [87] S. Nirjon, J. Gummesson, D. Gelb, and K.-H. Kim, "Typingring: A wearable ring platform for text input," in *Proc. of ACM MobiSys*, 2015, pp. 227–239.
- [88] N. D. Lane, P. Georgiev, and L. Qendro, "DeepEar: Robust Smartphone Audio Sensing in Unconstrained Acoustic Environments Using Deep Learning," in *Proc. of Ubicomp 2015*, pp. 283–294.
- [89] P. Georgiev, S. Bhattacharya, N. D. Lane, and C. Mascolo, "Low-resource Multi-task Audio Sensing for Mobile and Embedded Devices via Shared Deep Neural Network Representations," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, pp. 50:1–50:19, Sep. 2017.
- [90] J. Chauhan, Y. D. Kwon, P. Hui, and C. Mascolo, "Contauth: Continual learning framework for behavioral-based user authentication," *Proc. of ACM IMWUT*, vol. 4, no. 4, 2020.
- [91] Y. D. Kwon, J. Chauhan, A. Kumar, P. Hui, and C. Mascolo, "Exploring system performance of continual learning for mobile and embedded sensing applications," in *ACM/IEEE Symposium on Edge Computing*, 2021.



Kirill A. Shatilov is a PhD student majoring in Computer Science and Engineering at The Hong Kong University of Science and Technology. He is a member of HKUST-DT Systems and Media Laboratory. Kirill received his M.S. as well as B.S. degree of science in computer science in Novosibirsk State University. His main research interests are in the areas of natural human-computer interfaces, extended realities and mobile computing.

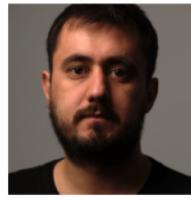


Young D. Kwon is a Ph.D. student majoring in Computer Science at the University of Cambridge, U.K. He received the M.Phil. degree from The Hong Kong University of Science and Technology and the B.S. degree from Sungkyunkwan University. He has won the Best Paper Award at SEC '21 and published in several renowned international conferences and journals including IPSN, IMWUT (UbiComp), INTERSPEECH, PerCom, ASONAM, and IEEE Pervasive Computing. His main research interests are in the areas

of On-Device Machine Learning, Continual Learning, Mobile and IoT Systems, and Ubiquitous Computing.



Lik-Hang Lee received the Ph.D. degree from SyMLab, Hong Kong University of Science and Technology, and the bachelor's and M.Phil. degrees from the University of Hong Kong. He is an assistant professor (tenure-track) with Korea Advanced Institute of Science and Technology (KAIST), South Korea. He is also the Director of the Augmented Reality and Media Laboratory, KAIST. He has built and designed various human-centric computing specialising in augmented and virtual realities (AR/VR).



Dimitris Chatzopoulos is an assistant professor at University College of Dublin. Dimitris received his PhD in Computer Science and Engineering from HKUST and his Diploma and Msc in Computer Engineering and Communications from University of Thessaly, Greece. His main research interests are in the areas of mobile computing and privacy-preserving mobile systems and decentralized blockchain-assisted systems.



Pan Hui received the Ph.D. degree from Computer Laboratory, University of Cambridge, and the bachelor's and M.Phil. degrees from the University of Hong Kong. He is the Nokia Chair Professor of data science and a Professor of computer science with the University of Helsinki. He is also the Director of the HKUST-DT System and Media Laboratory, Hong Kong University of Science and Technology. He has published around 500 research papers and with over 21,000 citations. He has 29 granted and filed

European and U.S. patents in the areas of augmented reality, data science, and mobile computing. He is an ACM Distinguished Scientist, a Member of the Academy of Europe, a fellow of the IEEE and the Royal Academy of Engineering .