WR-Hand: Wearable Armband Can Track User's Hand

YANG LIU, City University of Hong Kong, China
CHENGDONG LIN, City University of Hong Kong, China
ZHENJIANG LI[*], City University of Hong Kong, China

This paper presents WR-Hand, a wearable-based system tracking 3D hand pose of 14 hand skeleton points over time using Electromyography (EMG) and gyroscope sensor data from commercial armband. This system provides a significant leap in wearable sensing and enables new application potentials in medical care, human-computer interaction, etc. A challenge is the armband EMG sensors inevitably collect mixed EMG signals from multiple forearm muscles because of the fixed sensor positions on the device, while prior bio-medical models for hand pose tracking are built on isolated EMG signal inputs from isolated forearm spots for different muscles. In this paper, we leverage the recent success of neural networks to enhance the existing bio-medical model using the armband's EMG data and visualize our design to understand why our solution is effective. Moreover, we propose solutions to place the constructed hand pose reliably in a global coordinate system, and address two practical issues by providing a general plug-and-play version for new users without training and compensating for the position difference in how users wear their armbands. We implement a prototype using different commercial armbands, which is lightweight to execute on user's phone in real-time. Extensive evaluation shows the efficacy of the WR-Hand design.

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; **Interaction techniques**.

Additional Key Words and Phrases: Human Hand Pose Construction, Mobile Sensing, Deep Learning

ACM Reference Format:
Yang Liu, Chengdong Lin, and Zhenjiang Li. 2021. WR-Hand: Wearable Armband Can Track User's Hand. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 3, Article 118 (September 2021), 27 pages. https://doi.org/10.1145/3478112

1 INTRODUCTION

Tracking the geometric motion of a user's hand pose can enable many useful application designs, *e.g.*, disordered hand-motion rehabilitation, neural-aided human-computer interaction, virtual reality, etc. Existing hand-pose tracking solutions rely mainly on either deploying external devices, *e.g.*, cameras and depth sensors [9, 26, 41, 53] (suffering the inherent constraints on the light condition, line-of-sight and high computation cost, as detailed in Section 6), or pasting many sensors on various spots of our hand and/or forearm [15, 55], which is not user-friendly and not portable usually due to the cable connection from these sensors to a computer. In this paper, we instead aim to reconstruct and track the 3D hand pose (skeleton) continuously over time using only one wearable armband device. Our solution is also intended to be *lightweight* and *portable* to execute on the user's mobile device directly to support upper-layer applications.

[*] Corresponding author.

Authors' addresses: Yang Liu, Department of Computer Science, City University of Hong Kong, Hong Kong, China, yliu562-c@my.cityu.edu.hk; Chengdong Lin, Department of Computer Science, City University of Hong Kong, Hong Kong, China, chengdlin2-c@my.cityu.edu.hk; Zhenjiang Li, Department of Computer Science, City University of Hong Kong, Hong Kong, China, zhenjiang.li@cityu.edu.hk.

2474-9567/2021/9-ART118 $15.00
https://doi.org/10.1145/3478112

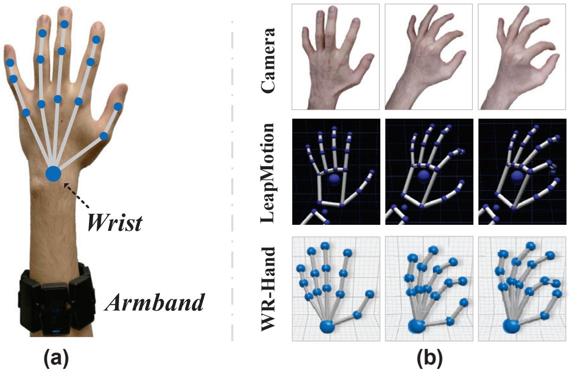

Fig. 1. (a) WR-Hand is built on the commercial armband to track 14 skeletal points on the hand. (b) When a user is performing an "OK" sign, three snapshots of the hand poses captured from camera, Leap Motion and WR-Hand, respectively.

We present WR-Hand to achieve such a design objective, which can be built using commercial armbands [1, 8]. Figure 1(a) illustrates our high-level design. Commercial armband is equipped with eight electromyography (EMG) sensors usually, which can measure the muscles' electrical activities. The muscle cells are activated at different levels when different hand poses are performed, and thus prior works [11, 21, 45], including the original designs of the commercial armbands [1, 8], have utilized the measured EMG signals to recognize a set of pre-defined hand gestures. Recent studies, *e.g.*, CapBand [68] and HandSense [56], could sense other related human bio- or capacitive-signatures for a similar purpose. However, these designs are mainly for gesture recognition[1], which cannot be adopted as a generic platform to track the fine-grained 3D geometric motion of hand pose over time.

In this paper, WR-Hand takes one step further to reconstruct and track the 3D hand pose of 14 skeleton points using the same set of the EMG data from the commercial armband. Figure 1(b) depicts that when a user's hand is performing an "OK" sign, the first two rows show three hand pose snapshots captured by camera and Leap Motion [9] separately. The third row shows that WR-Hand is able to achieve similar outputs as the second row, while it avoids Leap Motion's limitation to be connected to a computer by cable all the time. Thus, WR-Hand can offer a fully *portable* and *ubiquitous* solution that allows the user to bring along the hand pose tracking ability. However, designing WR-Hand encounters the following two challenges.

1) *Tracking methodology*. Our hand skeletons are connected to eight major skeletal muscles wrapped around forearm's radius and ulna bones. Bio-medical studies have established mature theoretical models to infer hand pose using the EMG measurements from forearm skeletal muscles [49, 55]. As the muscles are twisted partially (Section 2), these bio-medical studies have also identified a set of positions where the sensors should be placed, so that *isolated* and *strong* EMG signals from each muscle can be measured. However, an armband contacts only one cross-section of our forearm (as Figure 1(a) depicts). Thus, relatively weak and even mixed EMG signals from multiple muscles are naturally obtained due to the *sub-optimal sensor placement*. Moreover, existing models estimate mainly hand poses with respect to (*w.r.t.*) the user's wrist by assuming no user's arm movements. In this paper, we strive to further track the complete hand poses by superimposing forearm's orientation in a global coordinate system as expected by many applications [3, 23, 74], but we find unsatisfied results if we feed the armband's data directly to existing hand-pose models.

The main reason that prior hand-pose models do not work for armband is that the models' core recursive functions become ineffective when the mixed EMG signals are measured and they are difficult to be revised

[1] One recent work [34] makes a remarkable contribution to develop a wristband with thermal cameras to achieve a continuous hand pose tracking, while it suffers some inherent limitations (detailed in Section 2.2 and Section 6), which can be avoided with the WR-Hand design.

explicitly. Fortunately, we observe that the eight forearm muscles can be fully covered by the armband's EMG sensors, thereby inspiring us to leverage the recent success of recursive neural network (RNN), together with our proposed techniques, to enhance the existing hand-pose models, because RNN is intended for capturing the temporal relation of the input data and mining the sophisticated relation from the input to the desired output, which is hardly captured by the recursive functions or the hand-crafted features used before (Section 3.1). An accurate tracking thus becomes viable with this enhanced model.

On the other hand, gyroscope on the armband can measure forearm's orientation, but the orientation derived from gyroscope can easily cumulate errors over time [64] (detailed in Section 3.2). In WR-Hand, we view forearm's orientation as an inherent component of the hand-pose model and include gyroscope data as the tracker's input. This can benefit the orientation inference, because the tracker utilizes every input data batch independently for each estimation without an error accumulation (Section 3.2). However, EMG and gyroscope data may have different roles and importance under different hand poses during the tracking (Section 3.2). So, we further introduce a new design that could enable the tracker to adapt to and focus on the more crucial data type automatically, thereby improving the performance.

2) *Applicability issues*. We further consider two applicability issues to make WR-Hand practical and easy to use. First, we aim to offer a plug-and-play version for more new users *without training*. To this end, we can train a hand-pose tracker using a data set collected from multiple users, and then customize the adversarial-based domain adaptation framework [24, 77] to remove user-specific features brought by the data set to generalize the tracker through a gradient reversal based design by solving a unique convergence issue when this framework is adopted in WR-Hand (Section 3.3). Second, we cannot ensure that the armband is worn in the same position each time, in terms of the distance between the armband and the wrist, and the armband's rotation along the arm. We notice that the distance difference can be compensated by the input normalization as this difference mainly incur the signal strength change, while the device's rotation may degrade the performance significantly. To address this issue, we introduce an easy-to-perform calibration hand gesture and the user only performs it for three seconds when the armband is worn (one-time effort). WR-Hand can then pre-process the EMG inputs to best match a baseline rotation setting for compensation.

Experimental results. We develop a prototype of WR-Hand and experiment with 18 volunteer users. We train WR-Hand using 80% of the data from 10 users, and evaluate on their rest 20% data and the 100% data of other 8 new users. WR-Hand is deployed with two brands of commercial armbands, Myo [1] and gForce [8]. After we train the tracker using the data collected from one armband (*e.g.*, Myo in our experiments), it can be used for other armband (*e.g.*, gForce) directly with a simple one-time input data calibration (the tracker itself keeps unchanged) and achieves a similar tracking performance. On both armbands, WR-Hand can track 14 hand skeleton joints with an average error of 2.57 cm (Myo) and 2.61 cm (gForce), improving the recent bio-medical model [55] by more than 58%. Errors are not accumulated over time to support a continuous tracking and WR-Hand can execute on smart phone in real-time, outputting 33 pose frames / sec, to provide a portable tracking solution.

Contributions. This paper makes the following contributions: 1) We propose a lightweight and portable 3D hand pose tracking system only using the commercial wearable device. 2) We identify both methodology and applicability issues, propose several effective techniques to address them and also visualize our design to understand why our solution works without viewing neural network as a black-box tool. 3) We implement a prototype on different armbands and conduct extensive evaluations to show the efficacy of our design.

2 PRELIMINARY

2.1 Application Scenarios

With the tracked hand poses as shown in Figure 1(a), WR-Hand can benefit many useful applications, including:

Table 1. Comparison among different candidate devices.

Devices Features	Kinect	Leap Motion	Thermal Camera	Armband
Portable	×	×	✓	✓
Robust to the ambient conditions	×	×	×	✓
Independent of field of view	×	×	×	✓
Miniaturized size	×	✓	✓	✓
Running on the mobile device	×	×	×	✓

1) Disordered hand-motion rehabilitation. Some diseases or accidents can cause people disordered hand motions, *e.g.*, stroke, car accident, sport injury, and so on [4, 5]. Rehabilitation is a crucial and necessary step during the therapy of a patient's upper-limb function recovery. In addition to the clinic rehabilitation, the rehabilitation of the patient at home is equally important [6, 13]. Our technique can be used to record the detailed hand motion situation of the patient and assist the doctor to understand the rehabilitation status when the patient is at home (before the next treatment), so that the doctor can adjust the therapy progress in time. On the other hand, patient's hand poses can be visualized, which can provide useful guidance to make the rehabilitation more effective.

2) Neural-aided human-computer interaction interfaces. Our technique can also enable novel human-computer interactions. For example, in manufacturing, the engineer can use our technique to track his/her hand poses. The hand poses are then duplicated as input to control some robotic arms and hands, which are deployed in some (harsh or inaccessible or dangerous) areas, to perform the machine repairing, testing or maintenance. Our technique can be useful to the research and development of the prosthetic control system as well.

3) Virtual reality (VR). Our technique can further advance the VR application designs using the hand poses as one of the system inputs, without the limitation of the field of view of the camera on the VR helmet. For example, the users can operate more sophisticated tools or weapons controlled by their finger motions in VR games, play piano through VR, learn the human anatomy with their hand operations on a virtual human skeleton, etc.

2.2 Tracking Device Used in WR-Hand

Tracking device selection. Before we continue, we first discuss our considerations in the tracking device selection. In Section 4, we will utilize Leap Motion [9] to get the ground truth of the users' hand poses for training and evaluating WR-Hand. Therefore, one natural question is: **why not use Leap motion to fulfill the design directly?** Although the depth-sensor based devices, *e.g.*, Kinect and Leap Motion, can track the user's hand pose directly, they are not adopted in designing WR-Hand because they need to connect to a computer by cable all the time, which are not *portable*.[2] Moreover, these devices may also suffer the inherent limitations of being sensitive to the ambient environments, *e.g.*, background and illumination, having a limited field of view (and working range), requiring a line-of-sight, incurring high computation overhead, etc. One recent work [34] makes a remarkable contribution to develop a wristband with four thermal cameras for a continuous hand pose tracking, while such cameras could be impacted by the surrounding thermal sources, *e.g.*, other people nearby, and the line-of-sight, *e.g.*, wearing gloves. Moreover, the overhead to process the image data is still high, and [34] needs to offload the images to the laptop by a Wi-Fi router for synchronizing and processing the collected images, as summarized in Table 1. In this paper, we adopt the EMG sensors to avoid these limitations. As the EMG sensors need to wrap the user's forearm, we adopt armbands in designing WR-Hand.

[2] Although the idea of Mobile Leap Motion has been proposed [2, 10], it is intentionally designed to be embedded in the AR helmet, which is not suitable for the daily usage and thus cannot replace the armbands in the design of WR-Hand.

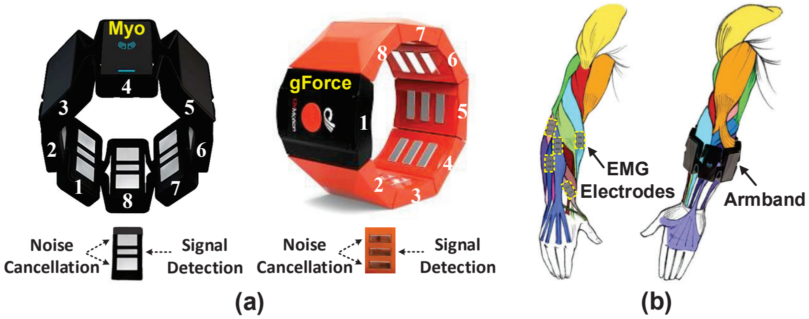

Fig. 2. (a) Myo/gForce armband with eight EMG sensors and motion sensors. (b) Each color represents one skeletal muscle. The left-hand side shows a part of sensor positions in [55] and the right-hand side shows the deployment using the armband.

Armband. Armband is a type of moderately-priced wearable devices, *e.g.*, ~200 USD [1] which are less expensive than many smart watches usually (while we agree that it may indeed bring such an extra device cost to the users if they need the WR-Hand service). In this paper, we develop WR-Hand using Myo and gForce two different commercial armbands as Figure 2(a) shows. Both armbands have EMG and motion two kinds of sensors, which have been used to recognize a specific set of pre-defined hand gestures [1, 8, 21]. WR-Hand takes one step further to reconstruct and track the user's detailed 3D hand poses over time by using the same set of sensors.

1) EMG sensors. The human nervous system utilizes electric signals to control the actions of different body parts. Electromyography (EMG) is a technique to measure the electrical activity levels of muscles. Each EMG sensor on Myo/gForce consists of two noise cancellation areas and one signal collection area (sampled at 200 Hz and 1000 Hz, respectively). As shown in Figure 2(a), both armbands could provide eight-channel EMG signals.

2) Gyroscope. Both armbands are also equipped with the three-axis gyroscope to estimate the orientation of the object it is attached to [78]. WR-Hand leverages its sensory data to determine the orientation of the hand pose.

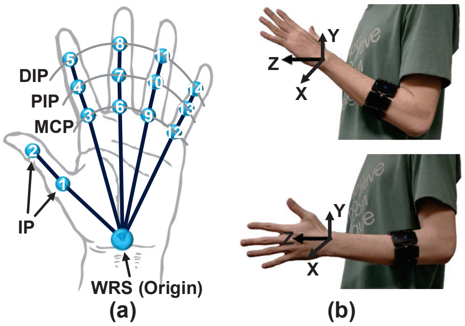

Fig. 3. (a) 14 hand skeleton joints to be tracked by WR-Hand. (b) Illustration of the coordinate system in WR-Hand, wherein the hand poses with the same shape but under different orientations can be distinguished.

2.3 Hand Pose Definition

Skeleton points. In WR-Hand, the hand pose refers to the geometric shape of the user's hand in a 3D space, which can be represented by 15 skeleton points as shown in Figure 3(a). These skeleton points can be classified into five groups: 1) **DIP**: four distal interphalangeal joints; 2) **PIP**: four proximal interphalangael joints; 3) **MCP**:

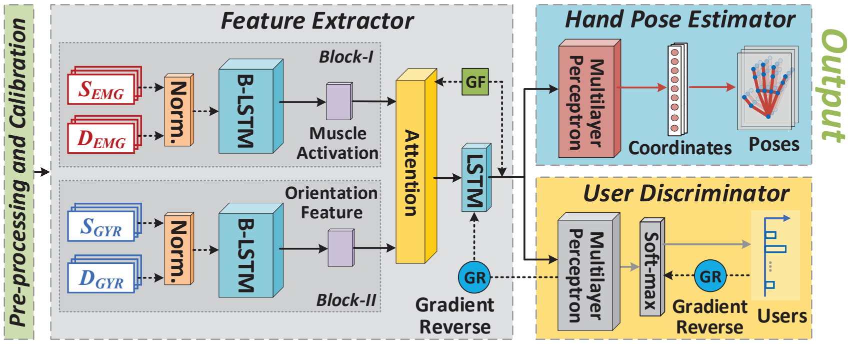

Fig. 4. System architecture of WR-Hand.

four metacarpophalangeal joints; 4) **IP**: two thumb joints; and 5) **WRS**: wrist. The origin of WR-Hand's coordinate system is placed on the wrist (stated below) and thus, we focus on tracking other **14** skeleton points in the first four groups in WR-Hand.

3D coordinate system. We adopt the following 3D coordinate system to quantify the locations of each skeleton point. Its X-Z plane and Y-axis is parallel and perpendicular to the horizontal plane of the earth coordinate system (*i.e.*, the ground)[3], respectively, and its origin is placed on the wrist. In this coordinate system, the hand poses, with the same shape but different arm orientations as shown in Figure 3(b), can be distinguished, which is required by many applications in practice. Meanwhile, the origin of the WR-Hand's coordinate system will change concurrently with the arm's moving and can avoid the extra error introduced by the estimation of the hand's absolution location in the earth coordinate system [44, 64, 65].

2.4 From EMG Signals to Hand Poses

Prior bio-medical works [22, 55] have studied the relation between the EMG signals of forearm muscles to fingers' motion. In Section 3, we will adopt a recent bio-medical model [55] as a concrete instance to instrument our design. It provides four well-estimated steps to track hand poses, and we also follow these steps. Since prior bio-medical studies (including [55]) mainly assume no user's arm movements, the model in [55] can only track hand pose *w.r.t.* the wrist (even after our enhancement). Hence, we will enable this primary hand pose tracking (*w.r.t.* wrist) using armband first (Section 3.1), and further propose new designs to achieve complete hand poses (superimposing forearm's orientation) (Section 3.2).

3 DESIGN

WR-Hand architecture is shown in Figure 4. It has four components to fulfill four main functions: a) Primary hand pose estimation *w.r.t.* the wrist in Section 3.1; b) Complete hand pose estimation with the forearm's orientation in a global 3D coordinate system in Section 3.2; and c-d) User-dependence removal to provide a more general plug-and-play version and calibrating the wearing position of armband in Section 3.3. We now elaborate on each function in the rest of this section.

[3] The X- or Z-axis of this coordinate system may have a fixed offset *w.r.t.* the geographic north, but it does not impact the hands-pose tracking.

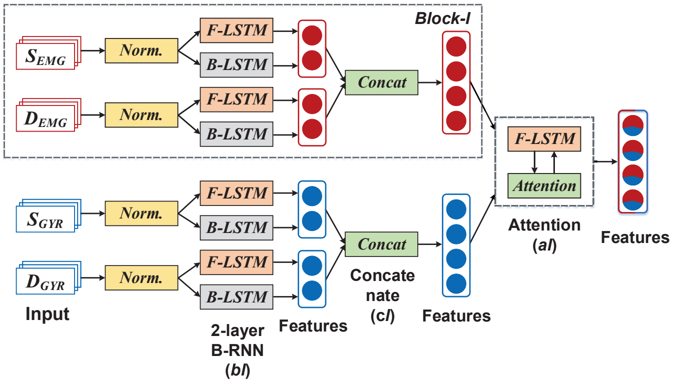

Fig. 5. Internal view of feature extractor in Figure 4.

3.1 Primary Hand Pose Estimation

This function involves "Block-I" in *Feature Extractor* and *Hand Pose Estimator* two components in Figure 4.

Primary tracker design. We leverage the recent success of neural networks to enhance the bio-medical model as stated in Section 2.4, following its four well-established steps:

1) Pre-processing EMG data. We first apply a high-pass filter (with a cut-off frequency at 30 Hz [17]) to exclude the low-frequency noises in the collected EMG signals [69], and then consider explicitly both the EMG signal strength related to the spatial shape of the hand pose and the EMG signal difference related to the hand pose movement as the model input. In particular, we denote S_t as the t-th EMG data frame, and then calculate D_t as the difference between two consecutive frames by $D_t = S_t - S_{t-1}$, where D_0 is defined as 0. We can generate two temporal input sequences: 1) the raw EMG data and 2) the EMG difference, denoted as S_{EMG} and D_{EMG}, respectively. We then pass them to the hand pose feature extractor, as depicted in its Block-I in Figure 4.

2) Extracting muscle's activation level. Muscle's activation is computed for obtaining muscle's physical contraction level in Step 3). Because EMG always appears before the actual activation of the corresponding muscle due to the electromechanical delay [66], recursive function is widely used in the prior bio-medical model to capture the relation of EMG s_m and activation level $u_m(t)$ of each forearm muscle m, *e.g.*,

$$u_m(t) = \alpha \times s_m(t - d_m) - \beta \times u_m(t-1) - \gamma \times u_m(t-2), \tag{1}$$

where d_m represents the electromechanical delay (in a range from 10 ms to about 150 ms [55]), and α, β and γ are three coefficients to be determined in the training.

However, the armband EMG sensors are not optimally placed and may further measure mixed EMG signals from multiple muscles as depicted in Figure 2(b). Traditional recursive function cannot quantify directly muscle's activation using the armband's data. By further considering the electromechanical delay, we propose to adopt RNN to analyze the EMG data, because RNN is capable of analyzing temporal sequences. To avoid the vanishing gradient problem, we use LSTM (Long Short-Term Memory) as each basic RNN unit. To enhance the tracking ability, we use the bidirectional RNN (B-RNN), with both Forward (F-LSTM) and Backward (B-LSTM) hidden layers. In the dashed rectangle (Block-I) of Figure 5, the first B-RNN layer bl extracts the primary features from S_{EMG} and D_{EMG} after Normalization (Norm.) individually, which are then fused by Concatenate (Concat) layer cl. The merged features will be used to estimate the muscle's contraction level in the next step.

3) Calculating compound muscle contraction level. The extracted features in Step 2) could represent the mixture of the neural activation from different muscles. The non-linear relation $v_m(t) = \frac{e^{A_m \times u_m(t)}-1}{e^{A_m}-1}$ to estimate the muscle contraction $v_m(t)$ from its isolated activation level $u_m(t)$ in the prior bio-model thus cannot be applicable for the armband's EMG data any more, where A_m is a parameter. We hence apply a F-LSTM layer in *al* in Figure 5 to generate a compound representation of the muscle contraction level.

4) EMG-based hand pose estimation. As Table 2 shows, the contraction of each forearm muscle m will lead to the motion of certain finger(s). Hence, by training a multilayer perceptron (in Figure 4) to minimize the distance between the constructed hand pose from the nerual network and the ground truth, the muscle contraction level can be converted to the degree of freedom of the corresponding hand joint [55].

Table 2. Relation from forearm muscles to finger joints.

Forearm muscles	Finger joint motions
Abductor pollicis longus	Thumb abduction
Flexor carpi radialis	Wrist abduction
Flexor digitorum superficialis	2-5th finger PIP flexion
Flexor digitorum profundus	2-5th finger DIP flexion
Extensor digitorium	2-5th finger extension
Extensor indices	Index finger
Extensor carpi ulnaris	Wrist extension/abduction
Extensor carpi radialis	Wrist and thumb

Understanding the RNN-based model. Before continuing to introduce other components in WR-Hand, we understand why this RNN-based model can be effective as Section 4 shows first through a visualization.

1) Analysis method. In the AI domain [40, 47], researchers introduce an effective way recently to analyze RNN-based neural networks by calculating the saliency score (denoted as a_score) of each input channel c_i *w.r.t.* the network's loss L. In particular, for each input channel c_i, its a_score_i is calculated by adding the magnitude of the gradient between the input x^m and output y^m of each intermediate neural network layer m, from the neural network output (*e.g.*, output of *Hand Pose Estimator* in Figure 4) all the way to the input channel c_i: $a\_score_i = \sum_m \left|\frac{\partial y_i^m}{\partial x_i^m}\right|$, where each (x_i^m, y_i^m) layer is on the path from output to c_i. As there are four paths connecting each c_i to the output, where $i = 1, 2, \ldots, 8$ in both S_{EMG} and D_{EMG} (Figure 5), its a_score_i value is the summed gradients from all these four paths. Intuitively, the a_score values describe how much *efforts* or *focuses* have been made by the neural network to extract features from each input channel to infer the current output [40, 47]. Based on this, we investigate how our RNN-based model works for the hand-pose tracking next.

2) Understanding. We use one concrete example to facilitate our discussion. Figure 6(a) plots the eight-channel EMG signals from armband when a "Yeah" sign is performed repeatedly. In the past, when people extract hand-crafted features from signals for various sensing designs [63], we tend to focus on the input channels with relatively strong amplitudes, obvious changes and clear patterns (*e.g.*, periodical), such as c_4 to c_7 in Figure 6(a), because these features are directly visible (through our manual observation), physically meaningful and can be quantified easily. However, when we compare a_score values (we plot them in S_{EMG} for a clear illustration) with one popular hand-crafted feature, *e.g.*, EMG's signal magnitude, in Figure 6(b) after normalization for both, we can see that the neural network does not spend too much efforts (*e.g.*, small a_score values) to these strong-magnitude channels. This is understandable because RNN is powerful to analyze complicated input-output relation, thereby such apparent features can be captured easily.

From Figure 6(b), a more important observation is that the neural network is able to spend substantial efforts to extract useful features from the channels people may focus on less before, *e.g.*, c_1 to c_3, with the "unimpressive" features (*e.g.*, less obvious and regular) from our manual observation's perspective. To visualize

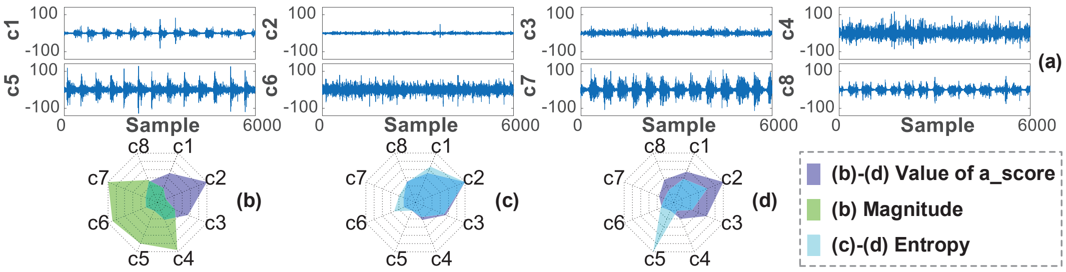

Fig. 6. Understanding the RNN-based hand-pose model. (a) EMG signals collected from armband when a "Yeah" sign is performed repeatedly. Normalized *a_score* values for the eight input channels vs. each channel's normalized (b) signal magnitude, (c) Entropy and (d) Entropy with the intentionally added noise on channel 5.

neural network's efforts made cross different input channels with the features of various "unimpressive" levels, we employ *Entropy* [61] to quantify[4] such feature differences for each input channel and compare them with their corresponding *a_score* values. Figure 6(c) shows these two "views" are consistent to each other. On the other hand, to ensure the unimpressive features are indeed from the useful EMG information, instead of meaningless noises, we intentionally add additional noises to one channel (c_5) in Figure 6(d). We can see the neural network is not distracted by such a high-Entropy yet meaningless input channel. In summary, the neural network can fully extract features from the input signals, even for the channels with features that are non-trivial to be extracted manually before. This is a possible reason we believe why RNN-enhanced model could be viable and effective (Section 4) even with the relatively weak and mixed EMG signals from the armband.

3.2 Upgraded Hand Pose Estimation

With the primary hand pose obtained from the armband's EMG data, next, we will further obtain the complete hand poses by superimposing forearm's orientation using the gyroscope data. For the gyroscope data, we also generate two temporal sequences: 1) the gyroscope data itself (S_{GYR}) and 2) the differences between adjacent samples (D_{GYR}). We then apply a similar B-RNN structure to extract the features that contain the orientation information (Figures 4 and 5). Especially, because neural network only takes the current input data batch to conduct the estimation and each input batch size is small (*e.g.*, $T = 50$ that is about 0.75-second data in our current design), the tracking errors are not accumulated over time, which can support a continuous tracking (Section 4).

Data fusion issue. Now we have two types of input sensor data (EMG and gyroscope), which have different roles in the hand pose tracking. For example, Figure 7(a-b) show the EMG and Gyroscope signals under a continuous finger movement (only) in the first row and a continuous wrist rotation movement (only) in the second row, respectively. We can see that the EMG data exhibits stronger and more diversified responses to the finger movements, while the gyroscope data show an opposite trend. We find that if we treat and fuse these two types of sensor data equally all the time in the hand-pose tracker, it will degrade the tracking performance.

Proposed solution. We introduce an *attention*-based adapter to differentiate the importance of EMG and gyroscope data. Attention [44] is a technique to leverage dynamic weights for the extracted features from the neural network input without modifying the network structure, which can identify the more crucial parts from the input data automatically.

For the attention layer (*al*) in Figure 5, we introduce a weight w_t^s for each feature output y_t^{s,fl_2} from layer fl_2, where $s = 1$ *or* 2 to indicate the sensor data type. The features are fused by a weighted average $\sum_{s=1}^{2} w_t^s \cdot y_t^{s,fl_2}$,

[4] Entropy may not be the only way for this quantification, which merely serves as one possible method to facilitate our understanding.

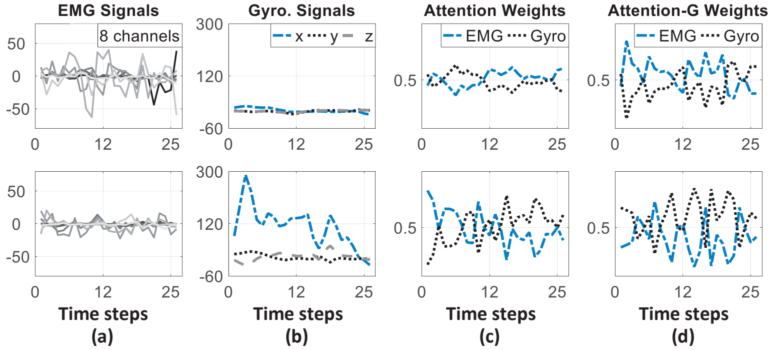

Fig. 7. (a) EMG and (b) Gyroscope signals, and weight updates in (c) basic attention design and (d) improved design with global feature under a continuous finger movement (the first row) and wrist rotation movement (the second row), respectively.

which will be analyzed further by another LSTM in layer rl (to be compatible with attention, we adopt one LSTM here). Figure 8 shows the internal structure of the attention layer al, and the weights $w_t = \{w_t^1, w_t^2\}$ will be adapted by

$$w_t = soft\text{-}max(W_{aa} \times a_t), \tag{2}$$

$$a_t = g(y_t^{fl_2} + W_{ah} \times h_{t-1}^{rl} + b_a), \tag{3}$$

where a^t is an intermediate variable; W_{aa}, W_{ah} and b_a are the parameters to be trained; $soft\text{-}max(\cdot)$ scales each weight in w_t within $[0, 1]$ and $g(\cdot)$ is a nonlinear activation function.

The attention layer employs $g(\cdot)$ to quantify the likelihood between the current feature extracted from each type of input data (at time t) and the internal state h_{t-1}^{rl} in the LSTM (rl), leading to the hand pose estimated until time t-1. After $g(\cdot)$'s parameters are determined in the training, the weight w_t^s for the sensor data type s, which contributes more to the current hand pose estimation, will be increased gradually by $g(\cdot)$ (as t increases), so that the network can focus on this more important input data type automatically. In other words, $g(\cdot)$ behaves like a lightweight neural network to enable weight adaptation. Studies [14, 71] find a simple one-layer perceptron, *e.g.*, $ReLu(\cdot)$ and $tanh(\cdot)$, is sufficient for attention, as it preserves the basic principle of a neural network, *i.e.*, inputs are combined linearly and then undergo the nonlinear function [44], while incurring a low computation overhead. We adopt $ReLu(x) = \max(0, x)$ in the design.

Figure 7(c) shows weight updates for Figure 7(a-b) using attention. The weights for these two types of sensor data have their initial values in the attention layer (that are different cross different input data batches). Then, the weight of the EMG data increases gradually and the weight of the gyroscope data reduces correspondingly

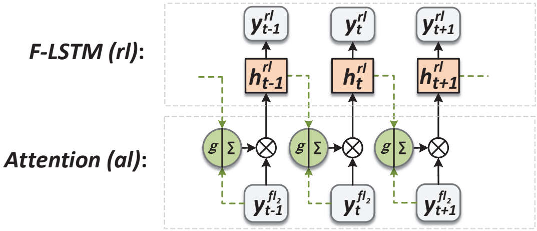

Fig. 8. Internal view of attention layer al and LSTM rl.

for the continuous finger movement in the first row. On the contrary, the case of a continuous wrist rotation movement shows an opposite trend in the second row. These results indicate the weights of two types of inputs can be automatically adjusted based on the network inputs, so that the hand pose estimation can focus more on the more effective and meaningful inputs.

Improving weight adaptation. Weights are updated in an iterative manner and thus, the system outputs from the beginning part of each batch are less reliable because the weights are not fully adjusted. Different from the classification problem [44], where the system can adopt the latter part of each batch to determine the result, each network output serves as an instant system result (hand pose) for a tracking problem in WR-Hand.

To address this issue, our key idea in WR-Hand is to provide a better *basis* for improving weight adaptation. Given an input data set $X = \{X_t\}$ and the desired neural network outputs for each input data sample, *i.e.*, labels $L = \{L_t\}$, we can apply the canonical correlation analysis (CCA) technique [35] to obtain a conversion $C(\cdot)$, which takes each pair of X_t and L_t as inputs to generate a pair of U_t and V_t: $C(X_t, L_t) = \{U_t, V_t\}$, so that the correlation between $X \cdot U$ (= $\{X_t \cdot U_t\}$) and $L \cdot V$ (= $\{L_t \cdot V_t\}$) these two sets can be maximized. By intuition, U and V essentially map X and L to certain domains respectively, wherein their correlation can be maximized. Hence, $X \cdot U$ is also named as the *global* feature of this network [35]. We integrate the global feature (GF in Figure 4) into Eqn. (3) to provide a better basis for the weight adaptation by:

$$a_t \quad = \quad g(y_t^{fl_2} + W_{ah} \times h_{t-1}^{rl} + W_{ag} \times X_t \times U_t + b_a), \tag{4}$$

where W_{ag} is a parameter. The conversion $C(\cdot)$ can be determined in the system training. However, during the system execution, we lack a real label for each X_t. In our current implementation, we adopt the hand pose outputted at the last time t-1 as an approximated label (the interval between two adjacent time steps is only 30 ms) to generate U_t for launching the weight adaptation in Eqn. (4). Figure 7(d) shows the weight updates with the global feature. We can see that using the global feature not only accelerates the initial weight updates, but also distinguishes the inputs' importance more clearly. In Section 4, we find that the global feature can indeed improve the tracking accuracy. Thus far, with the gyroscope's assistance, WR-Hand can estimate complete hand poses with the forearm's orientation.

3.3 More Practical Considerations

In this subsection, we address two practical issues in WR-Hand for 1) generalizing the system to a plug-and-play version and 2) compensating for the armband's wearing position difference.

3.3.1 User-Specifics Discriminator. EMG signals collected from different users (for training the system) inevitably consist of certain user-specific physiological features. So, we further design a more general plug-and-play version to make WR-Hand easier to use and applied further to more new users without training. Our key idea is to remove the user-specific features in WR-Hand, and we introduce a user-specifics discriminator to fulfill the design.

Discriminator design. As the architecture in Figure 4 shows, WR-Hand has three key neural network components: *feature extractor*, *hand pose estimator* and *user-specifics discriminator*. We denote their parameters as θ_f, θ_e and θ_d, respectively. For the hand pose tracking design stated in Section 3.2, we only need to optimize the parameters in feature extractor and hand pose estimator by minimizing the loss function $L_{tra}(\theta_f, \theta_e)$ to achieve a good tracking. The user-specifics discriminator is the third neural network. This network itself aims to use the same set of feature output (from feature extractor) to distinguish different users in the training data set, by minimizing its own loss $L_{dis}(\theta_f, \theta_d)$. In general, the feature extractor plays a min-max game against the user-specific discriminator to prevent the discriminator from distinguishing different users from the feature output (from the feature extractor). Therefore, to remove the user-specific features, the overall loss function for the entire system can be constructed as follows:

$$L_{loss}(\theta_f, \theta_e, \theta_d) \quad = \quad L_{tra}(\theta_f, \theta_e) - \lambda \times L_{dis}(\theta_f, \theta_d), \tag{5}$$

where λ is a factor. By minimizing Eqn. (5), we can first ensure tracking performance, since tracker's loss $L_{tra}(\theta_f, \theta_e)$ is still adjusted towards being minimized. Meanwhile, as $L_{dis}(\theta_f, \theta_d)$ is connected by "−", minimizing Eqn. (5) tends to maximize the discriminator's loss, *i.e.*, the selected features guide the discriminator not to distinguish different users. In other words, the finally extracted features are still effective for the tracking but they become more user-independent.

To train such a network, existing studies [31] suggest to iterate two steps. We first train the tracker by $(\hat{\theta}_f, \hat{\theta}_e) = \arg\min_{\theta_f, \theta_e} L_{loss}(\theta_f, \theta_e, \hat{\theta}_d)$. We can then train the discriminator by $\hat{\theta}_d = \arg\min_{\theta_d} L_{loss}(\hat{\theta}_f, \hat{\theta}_e, \theta_d)$ when fixing $\hat{\theta}_f$ and $\hat{\theta}_e$, where $\hat{\theta}_f$, $\hat{\theta}_e$ and $\hat{\theta}_d$ are the optimized parameters obtained so far. However, we find with RNNs and attentions that handle more complicated temporal relations, it is difficult to converge following this iterative training paradigm and the resulting system always achieves unsatisfactory performance.

Gradient reversal. In WR-Hand, we propose to add gradient reversal (GR) layers [24] to overcome this problem. In the training, GR does not change the computation stream in the forward propagation, *i.e.*, $GR(x) = x$, and it reverses the sign of the computed gradient in the back propagation, *i.e.*, $\frac{\partial GR(x)}{\partial x} = -\mathbf{I}$, where $\mathbf{I}$ is an identity matrix[5]. Hence, we can add one GR layer between the $soft\text{-}max(\cdot)$ function and discriminator's output (to distinguish different users in the training data set), as shown in Figure 4. By doing this, we can apply the standard stochastic gradient descendent (SGD) algorithm to train the entire network directly.

In particular, before we add the GR layer in the discriminator, the update of its parameters θ_d by SGD is incorrect: $\theta_d \longleftarrow \theta_d - \mu \times \frac{\partial L_{loss}}{\partial \theta_d} = \theta_d + \mu \times \frac{\partial L_{dis}}{\partial \theta_d}$, where μ is the learning rate and we omit the parameters in L_{loss} and L_{dis} two loss functions. The gradient update above clearly adds the gradient instead of the subtraction. However, with GR, the gradient update becomes correct as follows:

$$\theta_d \quad \longleftarrow \quad \theta_d - \mu \times \frac{\partial(-\mathbf{I} \times L_{loss})}{\partial \theta_d} = \theta_d - \mu \times \frac{\partial L_{dis}}{\partial \theta_d}.$$

Although the GR layer can correct the parameter update in the discriminator, the reserved gradient value could cause the incorrect update for θ_f in the feature extractor: $\theta_f \longleftarrow \theta_f - \mu \times (\frac{\partial L_{tra}}{\partial \theta_f} + \lambda \times \frac{\partial L_{dis}}{\partial \theta_f})$. We thus propose to add one more GR layer to the feature extractor before the gradient propagates into feature extractor (Figure 4), to reverse the addition to the subtraction before λ.

After the user-specific features are removed, we obtain a more generic plug-and-play version of WR-Hand, which can be applied directly for more new users without training.

3.3.2 Calibrating Wearing Position of Armband.

The position to wear the armband on the user's arm may vary each time, which could degrade WR-Hand's performance.

Distance difference. Possible distance difference between the armband and the wrist is usually no more than 4 cm due to armband's comfort and fit (from our measurements on 18 volunteers), and such slight difference could incur the EMG signal strength difference. Thus, we can compensate it by normalizing the input data (Section 4).

Rotation difference. Figure 9(a) shows the cross-section of our forearm, which can be divided into eight areas [68] from "1" to "8". We also number all EMG sensors on armband from "1" to "8" as shown in Figure 2(a). As a *baseline setting*, we make each sensor contact one of the eight forearm areas of the same ID. However, to minimize user's calibration effort, *e.g.*, no need to memorize these ID sequences, we propose to reorder all the signal inputs to follow the baseline setting as long as any EMG sensor block points to user's index finger direction when this finger is straightened.

[5] To implement GR, we only need to intercept the gradient value at the GR's location in the back propagation and invert its sign.

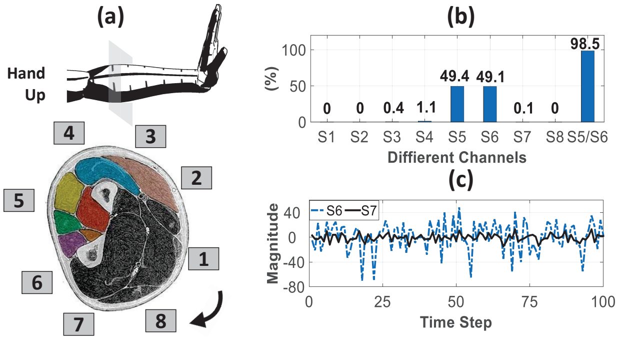

Fig. 9. (a) Baseline rotation setting. (b) Likelihood that the strongest EMG signal is collected from each sensor. (c) A showcase of the EMG signals collected from sensors 6 and 7.

To this end, we introduce an easy-to-perform calibration hand gesture "hand up"" as depicted in Figure 9(a). In Figure 9(b), we plot the likelihood that the strongest EMG signal can be collected from each sensor in the baseline setting — the sensor ID is the same as the forearm area ID. If we denote the strongest EMG signal is collected from sensor s, we can see that s is either 5 or 6 in 98.5% cases, but we cannot reliably distinguish them further. Fortunately, we find the signals from other sensor pairs could provide additional hints to determine s. For instance, sensor $s + 1$ could represent sensor 6 or 7, and Figure 9(c) shows that the signals from these two sensors exhibit certain differences. In addition, the signals from sensors 3 and 4 (for $s - 2$), as well as sensors 2 and 3 (for $s - 3$), also have differences. Thus, we can train three corresponding classifiers and conduct a majority vote (to achieve a better reliability) on their classification results to determine s through following three steps.

- **Step 1**: Rotate armband to make any sensor block point to the index finger and then perform "hand up" for 1 s after the armband is worn. We denote the sensor ID as $\hat{s}$, from which the strongest EMG signal is collected. Sensor $\hat{s}$ should correspond to sensor 5 or 6 in the baseline setting.
- **Step 2**: Feed the signals from other three sensors "$(\hat{s} + 1) \bmod 8$", "$(\hat{s} - 2) \bmod 8$" and "$(\hat{s} - 3) \bmod 8$" to the classifiers. Majority vote then can determine $\hat{s}$, which should correspond to 5 or 6.
- **Step 3**: Calculate the rotation offset $\Delta = s - \hat{s}$. Based on Δ, WR-Hand can reorder the collected EMG signals concurrently (rotating in a clockwise order when Δ is positive; Otherwise, it is a reversed order), so that the strongest signal virtually from sensor s (actually from sensor $\hat{s}$).

The reordered EMG signals then can be used for the hand pose tracking, and one point is worth noting — *Tiny rotation mismatches*. There could exist a tiny rotation mismatch (*e.g.*, $1 \sim 2°$) every time even the user tries to point one sensor block to the index finger, which may impact the performance. Fortunately, in the system training, such tiny mismatches are included already from the training data because the wearing of armband cannot be perfectly consistent cross different users, which thus improve the system robustness and remain the tracking performance (Section 4).

4 EVALUATION

4.1 Evaluation Methodology

We built a WR-Hand prototype using both Myo and gForce armbands. The neural network is developed using TensorFlow and it is trained on a desktop with Intel Core i7-600CPU and Nvidia GTX 1080Ti GPU. After training, the entire system can be installed and executed on a smart phone, *e.g.*, SAMSUNG Galaxy S7.

Methodology. We have obtained the university's ethical approval for this study and recruited 18 volunteers (9 females) to participate into the experiment with ages from 18 to 27. The circumference of their forearms, where they feel comfortable to wear the armband, vary from 18 cm to 30 cm. We anonymize the data to ensure the user's privacy. Before the data collection, a tutorial on the device usage and the data collection procedure are given to each user. The user then wears the armband with one sensor block pointing to the index finger to record eight channels of the EMG and three-axis gyroscope data for the following 15 motion trials:

a) Meaningful hand gestures. In trials 1 to 12, we collect data first for the calibration hand gesture "hand up" and then other 11 gestures, including 1) yeah, 2) pistol, 3) rock, 4) stretch, 5) thumb up, 6) thumb down, 7) zero, 8) OK, 9) fist, 10) call me, and 11) rest, with the following considerations. On one hand, these gestures are widely performed in our daily life. On the other hand, they cover the gestures with the movements from only several isolated fingers and also all the fingers, which could enable a more comprehensive examination of WR-Hand.

b) Free motions with 3 different motion trails: 1) users keep their arms and wrists static and only move their fingers freely (finger motion only); 2) users keep their fingers static and move their wrists (together with arms) freely (wrist motion only); and 3) users freely move fingers and wrists concurrently.

When we collect the sensor data from each user, we deploy a Leap Motion to get the ground truth for training and evaluation purposes of WR-Hand, which is placed on table in front of the user's left shoulder. For each data collection trial, the user performs one gesture (repeatedly) or free motion in five rounds. Each round lasts one minute and there is a one-minute rest between two rounds. We first finish above data collection using one armband, *e.g.*, Myo in our experiment, which contains 2700 pieces of sensory data (one piece lasts for near one minute). Because WR-Hand requires users to perform the calibration hand gesture "hand up" initially and all the users did it already after wearing the armband, we did not ask them to detach the device and wears it again intentionally in this data collection. On the other hand, as stated below, after we train the WR-Hand tracker using the data from one armband, it can be used for another armband (*e.g.*, gForce in our experiment) directly with a simple one-time input data calibration (the tracker itself keeps unchanged). Hence, for gForce, we only need to collect the testing data for the users. The detailed architecture of WR-Hand is tabulated in Appendix.

Training. We use 80% of the Myo data from 10 users to form the training data set. In the rest of this section, we evaluate WR-Hand's performance using: a) other 20% of the Myo data from these 10 users, b) 100% Myo data from another 8 new users (whose data are not used in the training at all), and c) all the gForce data. We have summarized the hyper-parameters adopted to train our system in Table 3.

Armband Calibration. To reuse the tracker trained already (instead of training different trackers for different armbands), we conduct an one-time calibration for gForce. In particular, when all a user's fingers are naturally straightened, we measured the average EMG value of each channel (within a 5-second window) for both devices, and compute $r_i = EMG_i^{Myo}/EMG_i^{gForce}$, where i is the channel index. Afterwards, we simply multiply r_i to the EMG data collected from gForce's channel i before they are fed to the tracker. We collect the eight r_i values using the data from one randomly selected user and then apply them to all other users' data in the following evaluation.

Evaluation metric. We measure the pose tracking error for each of 14 skeleton joints by calculating the Euclidean distance between WR-Hand's estimated location and the ground truth from Leap Motion.

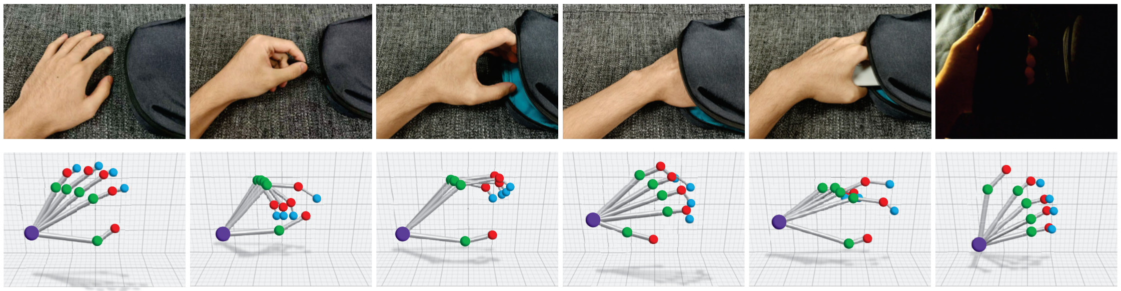

Fig. 10. Illustration of the 3D hand pose output of WR-Hand, when a user takes a phone out of the bag. The first row shows the images for six snapshots captured by a camera. The second row shows WR-Hand's results. WR-Hand can recover the skeleton joints even they are occluded by other fingers and objects (*e.g.*, bag), or in a poor lighting condition.

4.2 Overall Performance

In this subsection, we first evaluate the overall hand pose tracking accuracy by comparing the following methods:

1) WR-Basic: The tracked hand pose consists of both the hand shape and forearm orientation.

2) Bio-Model method [55]: The state-of-the-art EMG-to-muscle activation model stated in Section 3.1.

We first compare above two methods using the testing data from the 10 users, whose (Myo) data are used in the system training.We will further examine the more generic version of WR-Hand on the new users in Section 4.4.

Illustration of WR-Hand's output. Before quantifying the accuracy of WR-Hand, we first illustrate our system output through one concrete example. Figure 10 shows six snapshots when a WR-Hand user takes a phone out of the bag. We can see that WR-Hand can reconstruct a fine-grained result for 14 skeleton joints, which are quite similar as the user's actual hand poses depicted in the first row. Moreover, WR-Hand can also recover the skeleton joints even they are occluded by other fingers and objects (*e.g.*, bag) or the lighting condition is poor, which is a fundamental challenge for the vision-based methods. Next, we quantify WR-Hand's tracking accuracy and compare it with the Bio-Model method.

Tracking accuracy. In Figure 11, we examine the WR-Hand's tracking accuracy for each of the 14 skeleton joints. From Figure 11, we can see that tracking errors from these two armbands are comparable. In particular, the average error is 25.7 mm for Myo, which is lower than gForce since the tracker is trained using the Myo's data only. Since our armband calibration is effective, the average error is only slightly increased to 26.1 mm for gForce. The comparable accuracy from both devices also indicates the good generalization of the WR-Hand design.

In Figure 12(a), we further compare the hand-pose tracking accuracy of WR-Hand using two armbands respectively with the Bio-Model method [55]. Because these skeleton joints form MCP, PIP, DIP and IP four groups as depicted in Figure 3(a), for a clear presentation, we report the tracking error of each group (average error of all skeleton joints in one group) in Figure 12(a). The results show that the average error cross all the four groups of Bio-Model is relatively high, *e.g.*, 62.3 mm, since the armband cannot ensure the isolated and strong EMG signals as desired by Bio-Model due to its fixed sensor positions on the device. With the WR-Hand design, the average error can be reduced to 25.7 mm with a 58.8% error reduction for Myo and 26.1 mm with a 58.1% error reduction for gForce. From Figure 12(a), we also observe accuracy of the joints far away from the wrist is slightly worse for all three methods, *e.g.*, 32.5 mm, 34.1 mm and 72.9 mm on DIP group respectively, because the EMG signals are weaker reaching these finger joints.

Table 3. The hyper-parameters adopted to train WR-Hand.

Hyper-parameters	Value
Learning rate	0.002
Decay rate of learning rate	0.96
Decay step of learning rate	200
Batch size	1200
L2 regularization	0.0001
Dropout rate	0.16
Cell dropout rate for LSTM	0.16

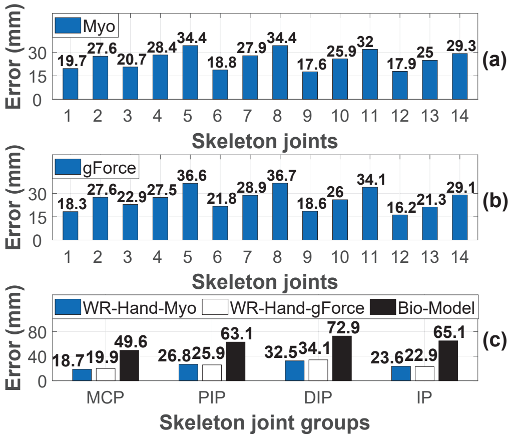

Fig. 11. Tracking error for each of 14 skeleton joints in WR-Hand as shown in Figure 3(a) using (a) Myo and (b) gForce. (c) WR-Hand is compared with the Bio-Model [55].

4.3 Ablation Study

Next, we conduct an ablation study to understand the design efficacy of each system module in WR-Hand.

Efficacy of network modules. We configure WR-Hand to the following versions by adding network modules gradually atop a benchmark and then examine the tracking performance of each version as shown in Figure 12:

(a) WR-Basic: this is a baseline version of WR-Hand, which is the *primary hand pose estimator* design introduced in Section 3.1, composed of four vanilla *bi-directional* LSTMs. Compared with the average error of ~60 mm ("Bio-Model") in Figure 11(c), "WR-Basic" can reduce it to ~28 mm, providing a good design baseline.

(b) WR-SL: this is an even simplified version than "WR-Basic", wherein we simplify the four *bi-directional* LSTMs in "WR-Basic" to four *single-directional* LSTMs. Figure 12(a) suggests that "WR-Basic" improves the tracking performance by 10.79% for Myo (error is reduced from 31.5 mm to 28.1 mm) and 5% for gForce (error is reduced from 30 mm to 28.5 mm) compared with "WR-SL".

(c) WR-wo-GF: this version includes our primary attention-based adapter design atop "WR-Basic", yet without adding the enhanced scheme that utilizes the global feature (GF) of the training data. We find this version can further reduce the average error to 26.5 mm for Myo and 27 mm for gForce, respectively.

(d) WR-Hand: this version uses the global feature atop "WR-wo-GR", which is the full-version design for users whose data are used during training. The result in Figure 12(a) shows that it can further improve the tracking performance by 3.02% and 3.33% for Myo and gForce, respectively. The versions of (c) and (d) together indicate the efficacy of our attention-based update, as it can focus more on the more effective types of the input data.

(e) WR-Hand': this is still "WR-Hand", but it is directly applied to the **new** users, whose data are not used in the system training at all (the detailed performance of each new user is shown in next subsection). Because the user-specific features are not removed yet, the error will naturally increase when "WR-Hand" is applied for such new users directly, *e.g.*, 36.9 mm and 39.2 mm on these two devices, respectively, as shown in Figure 12(b).

(f) WR-Hand'-w-Dis-wo-GR: this version includes the discriminator module, but without using the gradient reversal (GR) layer yet. This version can reduce the error for the new users to 34.7 mm (for Myo) and 32.8 mm (for gForce) on average, leading to 16.33% and 5.96% improvements respectively, compared with WR-Hand'.

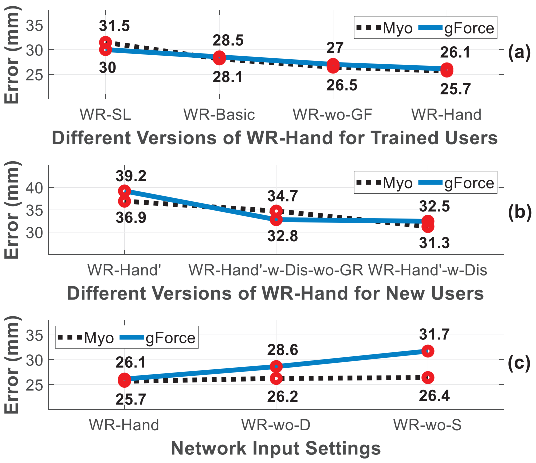

Fig. 12. Ablation study on different versions of WR-Hand over (a) the trained users and (b) the new users, as well as (c) the impact of different input settings.

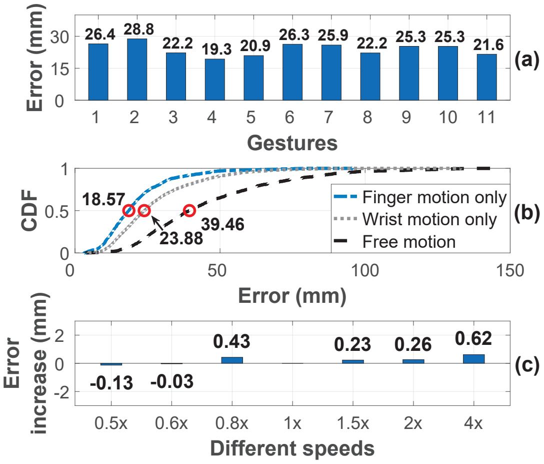

Fig. 13. Tracking errors for different (a) hand gestures, (b) motion types, and (c) hand movement speeds.

(g) WR-Hand'-w-Dis: this version further includes the gradient reversal (GR) layer atop "WR-Hand'-w-Dis-wo-GR" to make the discriminator more effective, which is the best version for the new users. The tracking error in Figure 12(b) are reduced to 31.3 mm and 32.5 mm on average for Myo and gForce, respectively.

Efficacy of different input settings. We further examine the effectiveness of different network input settings on the tracking performance in Figure 12(c). We consider the following two different input settings: i) **WR-wo-D**: we keep the original sensor data as input, and exclude the sensor data differences (as well as their associated LSTMs), and ii) **WR-wo-S**: we keep the sensor data differences as input, and exclude the original sensor data (as well as their associated LSTMs). The result in Figure 12(c) shows that excluding D_{EMG} and D_{GYR} from the input, the tracking performance of "WR-wo-D" can be slightly degraded, *e.g.*, the average error increases from 25.7 mm to 26.2 mm (for Myo) and 26.1 mm to 28.6 mm (for gForce), leading to 1.95% and 9.58% performance drop, respectively. On the other hand, Figure 12(c) also suggests that excluding S_{EMG} and S_{GYR} could lead to a similar performance drop for Myo compared with "WR-wo-D", while it deteriorates more for gForce, *e.g.*, the error is increased to 31.7 mm.[6] In summary, Figure 12(c) suggests that the better tracking performance can be obtained when these two types of inputs are both used in the system.

4.4 Micro-benchmarks

In this subsection, we further evaluate the impacts of various factors on WR-Hand's performance. Because WR-Hand achieves comparable performance with Myo and gForce, we mainly focus on one armband (*e.g.*, Myo) in this subsection for a clear illustration, while we also present the results for both devices when it is necessary.

Different types of motions. In Figure 13, we examine the tracking accuracy for different types of motions, including 11 pre-defined hand gestures and 3 forms of free motions stated in Section 4.1. For the hand gestures, the tracking error is from 19.3 mm to 28.8 mm on average in Figure 13(a). For the three types of free motions,

[6] This is likely because the network is trained by using the Myo data only and it is transplanted to gForce directly after a simple calibration. The imperfect sensor data alignment cross these two devices has been amplified when only the difference sequences are adopted, which impairs the calibration's effectiveness slightly.

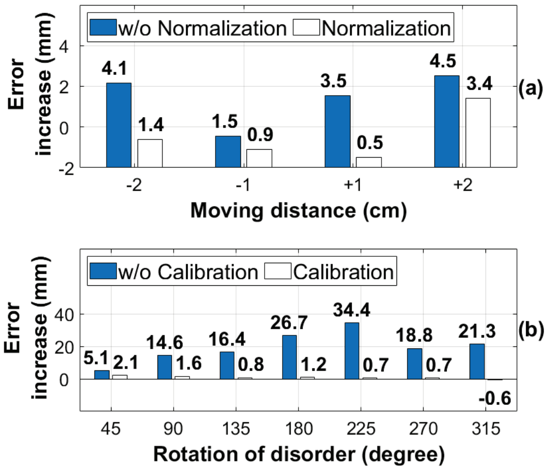

Fig. 14. (a) Impact of the armband's distance to the wrist. (b) Impact of the armband rotation with and without all the inputs being virtually reordered.

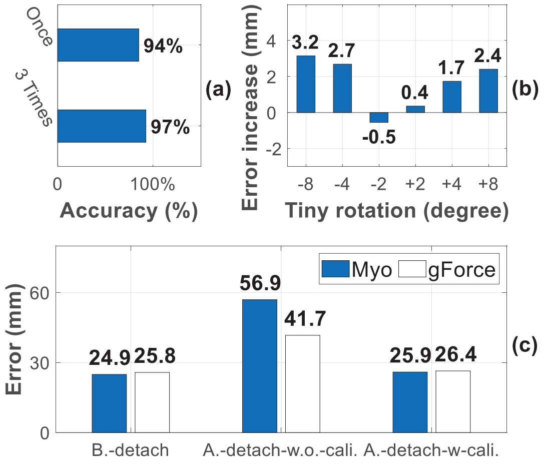

Fig. 15. (a) Accuracy of our rotation calibration design. (b) Impact of armband's tiny rotations. (c) Impact of the detaching operation.

their median hand pose tracking errors in Figure 13(b) are 18.57 mm, 23.88 mm and 39.46 mm, respectively. The slightly increased error for "concurrent" is because the EMG signals may also be caused by the arm motions. We anticipate more training data covering various free motions can further improve the performance and plan to investigate this opportunity in the future.

Different hand movement speeds. Figure 13(c) examines the performance under different hand movement speeds. In particular, we adopt a normal speed — about 1.5 seconds to complete one hand gesture, and then examine the system performance by speeding up (and slowing down) the speed from 1.5x to 4x (and from 0.8x to 0.5x). Figure 13(c) shows compared to the normal speed (1x), the tracking error differences under different speeds are less than 1 mm, indicating the hand movement speed has a limited impact on the tracking performance. The slightly decreased error at 0.6x and 0.5x is probably due to the relatively stronger muscle control for slowing down the movement to such speed levels.

Wearing distance differences. In Figure 14(a), we first investigate the impact of the distance between the wearing position of the armband and the wrist. In particular, we adopt the users' default wearing positions (where they feel comfortable to wear the armband) as each origin and then move device closer ("-") to and farther away from ("+") the wrist. We note that this difference will not be very large in practice since otherwise the armband wraps the user's arm loosely or tightly, and the user will feel very uncomfortable. Figure 14(a) shows this difference has a limited impact on the performance, as it mainly incurs signal strength changes, *e.g.*, the error is increased by no more than 3.4 mm only. Even the input normalization is disabled, the increased error is no more than 4.5 mm.

Wearing rotation differences. We next examine the impact of the rotation difference, in Figure 14(b), we intentionally rotate the armband (from the baseline setting) and examine how rotation impacts the performance. When we rotate the armband counterclockwise by one sensor block width (45°), the error increases 5.1 mm compared with the baseline performance. As we keep rotating, the increased error can be up to 34.4 mm. Hence, it is crucial to calibrate device's rotation initially and then virtually reorder all the signal inputs back to follow the baseline setting (Section 3.3.2). To this end, the first step is to obtain the number of sensor blocks to be rotated

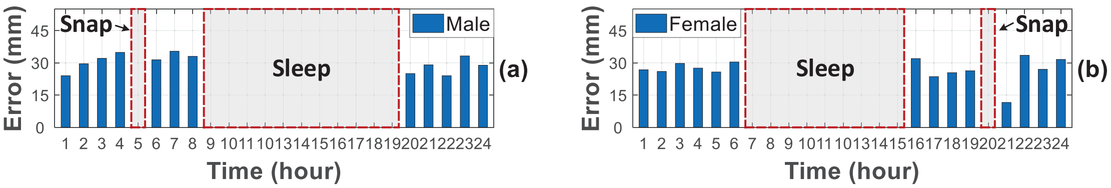

Fig. 16. Tracking errors from (a) one male user and (b) one female user with a 24-hour continuous wearing of the armband.

for calibrating the rotation disorder, *i.e.*, the rotation offset. Figure 15(a) shows that our method (Section 3.3.2) can achieve a high accuracy to obtain the correct offset. To further improve the calibration reliability, we utilize three individual calibration results to determine the final offset (*e.g.*, users performs "hand up" three times initially and each time lasts one second) and the accuracy is improved to 97%. We note that this accuracy is computed by using all the collected users' calibration gesture data for the evaluation purpose. When we actually calibrate using only the first three-second data for each user, we do not observe calibration errors in the experiment. After the offset is known, Figure 14(b) further shows that the system performance with the virtually reordered input signals, which is comparable with that when the armband's wearing directly follows the baseline setting, *e.g.*, the increased error is no more than 2.1 mm.

To achieve a more comprehensive understanding of the calibration design, we further examine the impact when the armband is detached and worn again explicitly. In Figure 15(b), we can see that before users detach the device, the average tracking error ("B.-detach") is 24.9 mm (for Myo) and 25.8 mm (for gForce). After the armband is detached from the arm, users wait for a while and then wear the armband again with one EMG sensor block pointing to the index finger direction as required in WR-Hand. From the result, we can see that if the calibration gesture is not performed after the armband is worn again, the detaching operation can impact the system performance significantly and the tracking error ("A.-detach-w.o.-cali.") is very large, *e.g.*, 56.9 mm (for Myo) and 41.7 mm (for gForce). The actual error increase is related to the rotation offset of the sensor blocks (with respect to the initial setting adopted in the training) caused by the detaching. Nevertheless, after the calibration is performed, all the signal inputs will be virtually reordered to follow the initial setting. WR-Hand then can work properly, and the tracking error ("A.-detach-w-cali.") becomes similar to that before the detaching, which indicates the necessity and efficacy of the wearing position calibration design.

Tiny rotation mismatches. On the other hand, even a user tries to point one sensor to her index finger or after a long-term wearing, there could always exist a tiny rotation mismatch *w.r.t.* the ideal baseline case. Fortunately, such tiny rotation mismatches are included already from training data in system training phase, as the wearing of armband cannot be perfectly consistent cross users. Figure 15(c) shows the system performance under different tiny rotations (within one EMG sensor block width) intentionally introduced. We can see that when the rotation is slight, *e.g.*, less than 2°, the increased tracking error is small, *e.g.*, less than 0.5 mm. When the tiny rotation continues further, the increased error is still not substantial.

Long-term tracking in the wild. We examine the tracking performance of two users (one male and one female) with a 24-hour continuously wearing of armband without taking off device from their arms (even in sleep), which covers two daytimes and one night. The daytime events contain all their daily activities such as commuting, eating, working, jogging, exercise, etc., and we only measure the tracking errors in the daytime. Since Leap Motion (to get ground truth) needs connecting to a computer, we cannot measure WR-Hand's tracking errors all the time. Therefore, the two users perform hand-pose motions in front of a deployed Leap Motion once per hour for the evaluation purpose, including the free motions and the gestures of yeah, pistol, rock, zero, OK, fist, call me, rest, wrist translation, etc. Figure 16 shows the results. The armband has a good ergonomic design, which

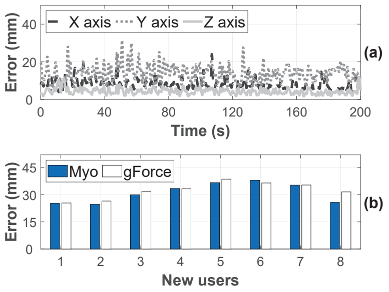

Fig. 17. (a) Tracking errors do not accumulate over time. (b) Tracking accuracy comparison among eight new users without training using Myo and gForce.

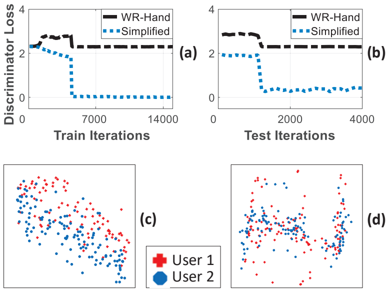

Fig. 18. Discriminator's loss values on (a) training and (b) testing data sets. Visualizing the extracted features from (c) the simplified version and (d) WR-Hand.

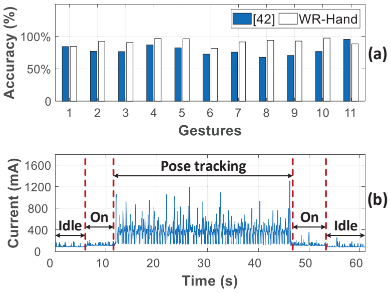

Fig. 19. (a) Accuracy to recognize hand gestures by WR-Hand's output and [42] using the raw EMG data. (b) Energy consumption of WR-Hand on the phone.

can firmly and comfortably wrap user's arm. Hence, even WR-Hand is calibrated only once after armband is worn, we can see that WR-Hand can perform well in the wild, *e.g.*, the average errors of these two users vary from 11.6 mm to 35.4 mm over a 24-hour time span in the experiment.

Tracking errors over time. Figure 16(c) further plots the average tracking errors along three axes for a 200-second hand pose tracking. The result shows that WR-Hand can provide a continuous tracking without error accumulation over time, since neural network only takes the current input data batch to conduct the estimation independently and each input batch size is small, *e.g.*, the batch size is about 0.75-sec data in the current WR-Hand.

New users and the discriminator design. We examine the performance among eight new users with the discriminator design on WR-Hand using two armbands. As shown in Figure 12(b) before, the tracking error is 36.9 mm and 39.2 mm on average for Myo and gForce, respectively, when WR-Hand is applied to the new users directly without the discriminator design. Figure 17(a) depicts that with the discriminator design, the tracking error on new users (whose data are not used in the training at all) can be reduced to 31.3 mm and 32.5 mm on average for these two devices respectively, which indicate that the discriminator can effectively remove some user-dependent features and perform better on new users.

We also look at the losses of the user discriminator in both the training and testing, as shown in Figure 18(a) and (b), respectively. We first configure WR-Hand to a simplified version that does not have the gradient reversal layer and stops the gradient propagation between the feature extractor and the user discriminator. The higher loss suggests the removal of the user-specific features from the training data. We can see that the loss of the user discriminator is bounded by 2.3 on both the training and testing data sets for WR-Hand. Meanwhile, the loss of the discriminator for the simplified version decreases rapidly, *e.g.*, 0.01 and 0.29 on the training and testing data sets, respectively. The results indicate the extracted features with our discriminator design are invariant across users. Figure 18(c) and (d) further illustrates the extracted features from the feature extractor in WR-Hand and the simplified version (using the stochastic neighbor embedding algorithm [46] to show the features on a 2D plane). In particular, two users perform a same hand gesture individually and their collected sensor data are provided to these two versions. Figure 18(c) shows the simplified version can extract some user-specific features due to the inconsistency between the distribution of the learned features from two users. On the contrary, WR-Hand can extract features from different users with almost similar distribution (Figure 18(d)), which further indicates the effectiveness of the user discriminator design.

Hand gesture recognition. As recent designs using the commercial armband [11, 37, 42] mainly focus on recognizing a set of pre-define hand gestures, we thus also compare the hand gesture recognition performance on the collected gestures in our data set using WR-Hand's output as the input with the recent method [42], which uses the EMG data from the armband directly. Figure 19(a) shows that the recognition accuracy of WR-Hand is from 81.72% to 97.54%. It outperforms the recent design [42] for nearly all the hand gestures in Figure 19(a). Because the reconstructed hand poses also implicitly embrace the bio-medical domain knowledge, Figure 19(a) unveils the recovered fine-grained hand poses are more indicative and meaningful to recognize hand gestures than adopting the raw EMG data directly.

System overhead. We examine the memory usage and execution time on both desktop and smart phone, *i.e.*, the desktop with Intel Core i7-600CPU and SAMSUNG S7. The average memory usage values are 450 MB on both platforms, indicating it is comfortable to deploy WR-Hand on different devices. Moreover, the execution time to construct one-frame hand pose is approximately 0.03 ms and 7.5 ms on the desktop and the smart phone, respectively. It shows that WR-Hand can reconstruct hand poses efficiently.

Table 4. Energy comparison with some built-in APPs on phone.

Apps	Music	Type	Photo	WR-Hand
Energy	150 mA	300 mA	650 mA	390 mA

We further measure the energy consumption of WR-Hand on SAMSUNG S7 using the Monsoon power monitor. In principle, WR-Hand can output 133 hand pose frames per second. However, to be more energy efficient, we estimate hand pose every 30 milliseconds, leading to about 33 frames per second in our current implementation, which is a common frame rate level on mobile devices, *e.g.*, camera. As a benchmark, the device's working current (mA) in the idle state with screen on is about 90 mA in Figure 19(b). When the application is open ("On"), the working current is 100 mA. After we execute WR-Hand, the average working current is 390 mA with the peak value of ~1000 mA, which leads to about 8-hour battery life with a continuous execution of WR-Hand on SAMSUNG S7. To further understand this energy profile, we have also measured the energy consumption of some in-built applications on the phone as illustrated in Table 4. From Table 4, we can see that the average working current of WR-Hand is slightly higher than the energy consumption when a user continuously types on the phone, indicating WR-Hand's energy consumption is acceptable for running on the mobile platform.

5 POINTS OF DISCUSSION

Network inputs. Our system has two types of inputs: 1) the original sensor data, including both the EMG (S_{EMG}) and gyroscope (S_{GYR}) sequences, and 2) the EMG's (D_{EMG}) and gyroscope's (D_{EMG}) differences. From Fig. 12(c), we find that including the sensor data differences as input, the system performance can be slightly improved. A similar observation has been obtained in prior studies as well, *e.g.*, [23]. Our understanding on this phenomenon is as follows. Due to the limited network size allowed for a mobile system, its feature mining ability more or less has a limit and some aspects of the features from the input sensor data might be under-explored yet. When we feed another modality or format of the input sensor data, we also add more LSTM branches to process them and all the extracted features will be used together for the hand-pose tracking finally. Therefore, providing new network branches may increase the chance to learn more comprehensive features, so as to harness them for the output. On the other hand, we find that in WR-Hand, the performance gain brought by D_{EMG} and D_{GYR} is not significant. Therefore, if a user prefers to reduce the system's cost further, such inputs and their associate LSTM branches can be removed to achieve a more lightweight network design.

Tracking accuracy. Because hand poses are subtle motions, the precise hand-pose tracking designs are always desired. In the literature, 3D hand pose is mainly tracked by the vision-based methods using cameras or depth sensors, which can achieve the performance with errors of about 7.5 to 24mm on the datasets with the pre-defined gestures [48, 72, 76]. These methods indeed generally achieve a higher accuracy than WR-Hand. However, our WR-Hand design still owns unique advantages, because these prior methods suffer inherent constraints due to the limited service coverage, light condition, line of sight, user-to-device distance and computation cost [68] and may incur some privacy concerns. When the sensing condition is not good, the performance of the vision-based methods could drop significantly, i.e., with occlusion and clutter, the hand-pose tracking error in [16] could be degraded to nearly 52 mm [16], which is much higher than WR-Hand. Although the WR-Hand design makes meaningful attempt towards avoiding these limitations, we agree that the accuracy obtained by the current WR-Hand design should be further enhanced, and the higher accuracy achieved by the vision-based methods in general suggests the room that the wearable-based hand pose tracking can be further improved at least. Therefore, we anticipate that new novel designs can be proposed to enhance the tracking performance in the future.

6 RELATED WORK

Hand pose estimation. Various hand pose estimation designs using cameras [20, 25, 26, 28, 53, 67, 79] or depth sensors [27, 73] have been proposed. However, they are limited due to the inherent constraints on the limited service coverage, light condition, line of sight, user-to-device distance and heavy computation cost [68]. They may incur privacy concerns as well. To mitigate these issues, some recent works [18, 39, 58, 60] estimate hand pose by sensor fusion technique. In particular, [39, 58] use two Leap Motions simultaneously to solve the problem of occlusion. [60] combines the Leap Motion with a customized non-vision based Flex sensor to track the hand pose. Without the heavy customised Flex sensor, [18] fuses the signals from Leap Motion and Myo for hand pose estimation. However, these works still depend on the Leap Motion, which needs to connect to a computer all the time (not portable) and is limited by the service coverage, line of sight, user-to-device distance, etc.

To avoid above issues, mobile or wearable based solutions then appear. In particular, designs using motion sensors on mobile devices [12, 19, 32, 43] are introduced. Studies [30, 50, 57] use the light reflected off the skin and some methods [33, 37, 51] also measure user's bio-signals. Tomo [75] further captures impedance tomography. CapBand [68] conducts a battery-free sensing and HandSense [56] recognizes micro finger gestures. Some methods based on commercial armbands have been developed as well, including the armbands' own applications [8], prosthetic hand control [42], etc. However, all these designs are for recognizing a specific set of pre-defined hand gestures merely. One recent work [34] develops a wristband with four thermal cameras for a continuous hand pose tracking, while such cameras could be impacted by the surrounding thermal sources, *e.g.*, other people nearby, and the line-of-sight, *e.g.*, wearing gloves. Moreover, the overhead of image processing is still high, which needs an offloading to the laptop by a Wi-Fi router for synchronizing and processing the collected images in [34]. Start-up [7] also aims to track hand skeletons. However, their solution requires dedicated device, and their hardware and software algorithms are not open to the general public yet. Recently, some smart gloves are developed for fine-grained hand pose tracking [29, 59]. To avoid occupying the user's hand by the gloves, the bio-medical studies [54, 55] have established the theories for the hand pose tracking by using the EMG signals from the forearm. However, these prior designs require the deployment of EMG sensors to specialized locations on the forearm with the medical knowledge, and these sensors are connected to a micro-controller or computer through the cable, which are not lightweight and portable.

Other wearable skeleton tracking designs. Recently, sensor data collected from wearable devices have been widely used to track other body parts, *e.g.*, legs [38], arms [44, 64, 65] and the torso [62, 70]. As the motion sensor data alone cannot provide enough information to track the hand pose, these methods cannot be applied to the WR-Hand design directly. On the other hand, for the hand orientation estimation, we do not use recent designs [64, 78] directly neither, as the derived orientation from gyroscope can accumulate errors easily, *e.g.*, A^3 [78] may lack opportunities to calibrate orientation errors and MUSE [64] is prone to drift along the magnetic north and it can be also impacted by the magnetic interference nearby. To address this issue, we integrate the gyroscope data into our neural network for the orientation estimation without the error accumulation, since the network takes the current input data batch to conduct the estimation independently and the batch size is small.

Assorted deep learning techniques. To design the plug-and-play version of the hand-pose tracker, we propose to remove some user-specific features (a type of the *domain* information) brought by the labelled data. The study [77] uses a source discriminator to guide the feature extraction to remove the coupling between the input data and domains [36]. But we find that the network training by existing designs is difficult to converge to achieve a good performance with RNNs used in our proposed solution to handle more complicated temporal relations. Inspired by [24], we propose the addition of two gradient reversal layers and integrate them with our system, so that standard back propagation can be applied to train the entire system directly. On the other hand, the attention technique has been applied previously to natural language processing [14] and computer vision [52]. A recent study [44] also uses attention to address the missing input data issue, wherein the neural network is designed for a classification problem. WR-Hand further uses attention to distinguish the importance of different data types and customizes it to better fit a tracking problem.

7 CONCLUSION

This paper introduces a lightweight and fully portable system WR-Hand for 3D hand pose tracking based on the commercial armband with EMG and gyroscope sensors. The primary design challenge stems from the degraded quality of the EMG data collected from the armband because of their fixed positions on the device as well as the twist muscles on the forearm. EMG data also need to be integrated intelligently with the gyroscope data and contain user-specific features. The armband's wearing position could vary as well. We address all these challenges in WR-Hand and develop a prototype to show the efficacy of the WR-Hand design.

ACKNOWLEDGMENTS

This work is supported by the GRF grant from Research Grants Council of Hong Kong (Project No. CityU 11217420).

REFERENCES

[1] 2015. Myo Price. https://www.businessinsider.com/myo-armband-demo-ces-2015-2015-1.

[2] 2016. Leap Motion Goes Mobile Blog. http://blog.leapmotion.com/mobile-platform/.

[3] 2019. Leap Motion Applications. https://gallery.leapmotion.com/.

[4] 2019. Outpatient Hand Rehabilitation. https://www.stonybrookmedicine.edu/patientcare/physical-occupational-therapy/hand-rehabilitation.

[5] 2020. Everything you need to know about stroke. https://www.medicalnewstoday.com/articles/7624#treatment.

[6] 2020. Hand Rehabilitation: 5 Best Methods for Recovery at Home. https://www.flintrehab.com/hand-rehabilitation/.

[7] 2020. Immersive Control. https://www.ctrl-labs.com/.

[8] 2021. gForce Armband. http://www.oymotion.com/en/product32/149.

[9] 2021. Leap Motion. https://www.leapmotion.com/.

[10] 2021. Leap Motion Goes Mobile. https://developer.leapmotion.com/107.

[11] João Gabriel Abreu, João Marcelo Teixeira, Lucas Silva Figueiredo, and Veronica Teichrieb. 2016. Evaluating sign language recognition using the myo armband. In *Proc. of IEEE SVR*.

[12] Sonu Agarwal, Arindam Mondal, Gurdeepak Joshi, and Gaurav Gupta. 2017. Gestglove: A wearable device with gesture based touchless interaction. In *Proc. of ACM AH*.

[13] Serdar Ates, Claudia JW Haarman, and Arno HA Stienen. 2017. SCRIPT passive orthosis: design of interactive hand and wrist exoskeleton for rehabilitation at home after stroke. *Autonomous Robots* 41, 3 (2017), 711–723.

[14] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. of ICLR*.

[15] Matteo Bianchi, Paolo Salaris, and Antonio Bicchi. 2013. Synergy-based hand pose sensing: Optimal glove design. *The International Journal of Robotics Research* 32, 4 (2013), 407–424.

[16] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 2019. 3d hand shape and pose from images in the wild. In *Proc. of IEEE CVPR*.

[17] Thomas S Buchanan, David G Lloyd, Kurt Manal, and Thor F Besier. 2004. Neuromusculoskeletal modeling: estimation of muscle forces and joint moments and movements from measurements of neural command. *Journal of applied biomechanics* 20, 4 (2004), 367–395.

[18] Jingxiang Chen, Chao Liu, Rongxin Cui, and Chenguang Yang. 2019. Hand Tracking Accuracy Enhancement by Data Fusion Using Leap Motion and Myo Armband. In *Proc. of IEEE ICUSAI*.

[19] Ke-Yu Chen, Shwetak N Patel, and Sean Keller. 2016. Finexus: Tracking precise motions of multiple fingertips using magnetic sensing. In *Proc. of ACM CHI*.

[20] Chiho Choi, Sangpil Kim, and Karthik Ramani. 2017. Learning hand articulations by hallucinating heat distribution. In *Proc. of the ICCV*.

[21] Ulysse Côté-Allard, Cheikh Latyr Fall, Alexandre Drouin, Alexandre Campeau-Lecours, Clément Gosselin, Kyrre Glette, François Laviolette, and Benoit Gosselin. 2019. Deep Learning for Electromyographic Hand Gesture Signal Classification Using Transfer Learning. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 27, 4 (2019), 760–771.

[22] Mia Erickson, Heather F Smith, Carol Waggy, and Neal E Pratt. 2020. Anatomy and Kinesiology of the Hand. *Rehabilitation of the Hand and Upper Extremity, E-Book* (2020), 1.

[23] Biyi Fang, Jillian Co, and Mi Zhang. 2017. DeepASL: Enabling Ubiquitous and Non-Intrusive Word and Sentence-Level Sign Language Translation. In *Proc. of ACM Sensys*.

[24] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research* 17, 1 (2016), 2096–2030.

[25] Ruohan Gao, Bo Xiong, and Kristen Grauman. 2018. Im2flow: Motion hallucination from static images for action recognition. In *Proc. of IEEE CVPR*.

[26] Liuhao Ge, Yujun Cai, Junwu Weng, and Junsong Yuan. 2018. Hand PointNet: 3d hand pose estimation using point sets. In *Proc. of CVPR*.

[27] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. 2017. 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images. In *Proc. of IEEE CVPR*.

[28] Liuhao Ge, Zhou Ren, and Junsong Yuan. 2018. Point-to-point regression pointnet for 3d hand pose estimation. In *Proc. of ECCV*.

[29] Oliver Glauser, Shihao Wu, Daniele Panozzo, Otmar Hilliges, and Olga Sorkine-Hornung. 2019. Interactive hand pose estimation using a stretch-sensing soft glove. *ACM Transactions on Graphics* 38, 4 (2019), 41.

[30] Jun Gong, Xing-Dong Yang, and Pourang Irani. 2016. Wristwhirl: One-handed continuous smartwatch input using wrist gestures. In *Proc. of ACM UIST*.

[31] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Proc. of NIPS*.

[32] Jose L Hernandez-Rebollar, Nicholas Kyriakopoulos, and Robert W Lindeman. 2002. The AcceleGlove: a whole-hand input device for virtual reality. In *Proc. of ACM SIGGRAPH*.

[33] Nalinda Hettiarachchi, Zhaojie Ju, and Honghai Liu. 2015. A new wearable ultrasound muscle activity sensing system for dexterous prosthetic control. In *Proc. of IEEE SMC*.

[34] Fang Hu, Peng He, Songlin Xu, Yin Li, and Cheng Zhang. 2020. FingerTrak: Continuous 3D Hand Pose Tracking by Deep Learning Hand Silhouettes Captured by Miniature Thermal Cameras on Wrist. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 2 (2020), 1–24.

[35] Xu Jia, Efstratios Gavves, Basura Fernando, and Tinne Tuytelaars. 2015. Guiding the long-short term memory model for image caption generation. In *Proc. of IEEE ICCV*.

[36] Wenjun Jiang, Chenglin Miao, Fenglong Ma, Shuochao Yao, Yaqing Wang, Ye Yuan, Hongfei Xue, Chen Song, Xin Ma, Dimitrios Koutsonikolas, et al. 2018. Towards Environment Independent Device Free Human Activity Recognition. In *Proc. of ACM MobiCom*.

[37] Xianta Jiang, Lukas-Karim Merhi, Zhen Gang Xiao, and Carlo Menon. 2017. Exploration of force myography and surface electromyography in hand gesture classification. *Elsevier Medical engineering & physics* 41 (2017), 63–73.

[38] Yonghang Jiang, Zhenjiang Li, and Jianping Wang. 2019. Ptrack: Enhancing the applicability of pedestrian tracking with wearables. *IEEE Transactions on Mobile Computing* 18, 2 (2019), 431–443.

[39] Haiyang Jin, Qing Chen, Zhixian Chen, Ying Hu, and Jianwei Zhang. 2016. Multi-LeapMotion sensor based demonstration for robotic refine tabletop object manipulation task. *CAAI Transactions on Intelligence Technology* 1, 1 (2016), 104–113.

[40] Andrej Karpathy, Justin Johnson, and Li Fei-Fei. 2015. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078* (2015).

[41] Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. 2013. Real time hand pose estimation using depth sensors. In *Consumer depth cameras for computer vision*. Springer.

[42] Agamemnon Krasoulis, Iris Kyranou, Mustapha Suphi Erden, Kianoush Nazarpour, and Sethu Vijayakumar. 2017. Improved prosthetic hand control with concurrent use of myoelectric and inertial measurements. *Journal of neuroengineering and rehabilitation* 14, 1 (2017), 71.

[43] Gierad Laput, Robert Xiao, and Chris Harrison. 2016. Viband: High-fidelity bio-acoustic sensing using commodity smartwatch accelerometers. In *Proc. of ACM UIST*.

[44] Yang Liu, Zhenjiang Li, Zhidan Liu, and Kaishun Wu. 2019. Real-time Arm Skeleton Tracking and Gesture Inference Tolerant to Missing Wearable Sensors. In *Proc. of ACM MobiSys*.

[45] Zhiyuan Lu, Xiang Chen, Qiang Li, Xu Zhang, and Ping Zhou. 2014. A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices. *IEEE transactions on human-machine systems* 44, 2 (2014), 293–299.

[46] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[47] Andreas Madsen. 2019. Visualizing memorization in RNNs. *Distill* 4, 3 (2019), e16.

[48] Jameel Malik, Ahmed Elhayek, Fabrizio Nunnari, Kiran Varanasi, Kiarash Tamaddon, Alexis Heloir, and Didier Stricker. 2018. Deephps: End-to-end estimation of 3d hand pose and shape by learning from synthetic depth. In *Proc. of IEEE 3DV*.

[49] Kurt Manal, Roger V Gonzalez, David G Lloyd, and Thomas S Buchanan. 2002. A real-time EMG-driven virtual arm. *Computers in biology and medicine* 32, 1 (2002), 25–36.

[50] Jess McIntosh, Asier Marzo, and Mike Fraser. 2017. Sensir: Detecting hand gestures with a wearable bracelet using infrared transmission and reflection. In *Proc. of ACM UIST*.

[51] Jess McIntosh, Charlie McNeill, Mike Fraser, Frederic Kerber, Markus Löchtefeld, and Antonio Krüger. 2016. EMPress: Practical hand gesture classification with wrist-mounted EMG and pressure sensing. In *Proc. of ACM CHI*.

[52] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. 2014. Recurrent models of visual attention. In *Proc. of NIPS*.

[53] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. 2018. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *Proc. of IEEE CVPR*.

[54] Jimson Ngeo, Tomoya Tamei, Kazushi Ikeda, and Tomohiro Shibata. 2015. Modeling dynamic high-DOF finger postures from surface EMG using nonlinear synergies in latent space representation. In *Proc. of IEEE EMBC*.

[55] Jimson G Ngeo, Tomoya Tamei, and Tomohiro Shibata. 2014. Continuous and simultaneous estimation of finger kinematics using inputs from an EMG-to-muscle activation model. *Journal of neuroengineering and rehabilitation* 11, 1 (2014), 122.

[56] Viet Nguyen, Siddharth Rupavatharam, Luyang Liu, Richard Howard, and Marco Gruteser. 2019. HandSense: Capacitive coupling-based Dynamic, Micro Finger Gesture Recognition. In *Proc. of ACM SenSys*.

[57] Masa Ogata and Michita Imai. 2015. SkinWatch: skin gesture interaction for smart watch. In *Proc. of ACM AH*.

[58] Salih Ertug Ovur, Hang Su, Wen Qi, Elena De Momi, and Giancarlo Ferrigno. 2021. Novel Adaptive Sensor Fusion Methodology for Hand Pose Estimation With Multileap Motion. *IEEE Transactions on Instrumentation and Measurement* 70 (2021), 1–8.

[59] Timothy F O'Connor, Matthew E Fach, Rachel Miller, Samuel E Root, Patrick P Mercier, and Darren J Lipomi. 2017. The Language of Glove: Wireless gesture decoder with low-power and stretchable hybrid electronics. *PloS one* 12, 7 (2017), e0179766.

[60] Godwin Ponraj and Hongliang Ren. 2018. Sensor fusion of leap motion controller and flex sensors using Kalman filter for human finger tracking. *IEEE Sensors Journal* 18, 5 (2018), 2042–2049.

[61] GE Powell and IC Percival. 1979. A spectral entropy method for distinguishing regular and irregular motion of Hamiltonian systems. *Journal of Physics A: Mathematical and General* 12, 11 (1979), 2053.

[62] Qaiser Riaz, Guanhong Tao, Björn Krüger, and Andreas Weber. 2015. Motion reconstruction using very few accelerometers and ground contacts. *Graphical Models* 79 (2015), 23–38.

[63] Muhammad Shahzad, Alex X Liu, and Arjmand Samuel. 2013. Secure unlocking of mobile touch screen devices by simple gestures: you can see it but you can not do it. In *Proc. of ACM MobiCom*.

[64] Sheng Shen, Mahanth Gowda, and Romit Roy Choudhury. 2018. Closing the Gaps in Inertial Motion Tracking. In *Proc. of ACM MobiCom*.

[65] Sheng Shen, He Wang, and Romit Roy Choudhury. 2016. I am a smartwatch and i can track my user's arm. In *Proc. of ACM MobiSys*.

[66] Nikhil A Shrirao, Narender P Reddy, and Durga R Kosuri. 2009. Neural network committees for finger joint angle estimation from surface EMG signals. *Biomedical engineering online* 8, 1 (2009), 2.

[67] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. 2014. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics* 33, 5 (2014), 169.

[68] Hoang Truong, Shuo Zhang, Ufuk Muncuk, Phuc Nguyen, Nam Bui, Anh Nguyen, Qin Lv, Kaushik Chowdhury, Thang Dinh, and Tam Vu. 2018. CapBand: Battery-free Successive Capacitance Sensing Wristband for Hand Gesture Recognition. In *Proc. of ACM SenSys*.

[69] Jingpeng Wang, Liqiong Tang, and John E Bronlund. 2013. Surface EMG signal amplification and filtering. *International Journal of Computer Applications* 82, 1 (2013).

[70] Frank Wouda, Matteo Giuberti, Giovanni Bellusci, and Peter Veltink. 2016. Estimation of full-body poses using only five inertial sensors: An eager or lazy learning approach? *Sensors* 16, 12 (2016), 2138.

[71] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proc. of ICML*.

[72] Cheol-hwan Yoo, Seung-wook Kim, Seo-won Ji, Yong-goo Shin, and Sung-jea Ko. 2019. Capturing Hand Articulations using Recurrent Neural Network for 3D Hand Pose Estimation. *Feedback* 15 (2019), 16.

[73] Shanxin Yuan, Guillermo Garcia-Hernando, Björn Stenger, Gyeongsik Moon, Ju Yong Chang, Kyoung Mu Lee, Pavlo Molchanov, Jan Kautz, Sina Honari, Liuhao Ge, et al. 2018. Depth-based 3d hand pose estimation: From current achievements to future goals. In *Proc. of*

IEEE CVPR.

[74] Xu Zhang, Xiang Chen, Wen-hui Wang, Ji-hai Yang, Vuokko Lantz, and Kong-qiao Wang. 2009. Hand gesture recognition and virtual game control based on 3D accelerometer and EMG sensors. In *Proc. of ACM IUI*.

[75] Yang Zhang, Robert Xiao, and Chris Harrison. 2016. Advancing hand gesture recognition with high resolution electrical impedance tomography. In *Proc. of ACM UIST*.

[76] Zhaohui Zhang, Shipeng Xie, Mingxiu Chen, and Haichao Zhu. 2020. HandAugment: A Simple Data Augmentation Method for Depth-Based 3D Hand Pose Estimation. *arXiv preprint arXiv:2001.00702* (2020).

[77] Mingmin Zhao, Shichao Yue, Dina Katabi, Tommi S Jaakkola, and Matt T Bianchi. 2017. Learning sleep stages from radio signals: A conditional adversarial architecture. In *Proc. of ICML*.

[78] Pengfei Zhou, Mo Li, and Guobin Shen. 2014. Use it free: Instantly knowing your phone attitude. In *Proc. of ACM MobiCom*.

[79] Christian Zimmermann and Thomas Brox. 2017. Learning to estimate 3d hand pose from single rgb images. In *Proc. of IEEE ICCV*.

APPENDIX

Table 5. Detailed network layers and its input and output sizes in WR-Hand. For the size in a form of $([a, b, c])$ and $([a, b])$, they refer to the 3-dimension and 2-dimension tensors respectively, *e.g.*, the value of a, b and c specifies the size of each dimension. For the size in a form of $(a, [b, c])$, a is the number of tensors and each tensor here is a $b \times c$ 2-dimension tensor.

Module	Input (size)	Layer	Output (size)
Feature Extractor	S_{EMG} ([1200, 50, 8])	Batch Normalization	S_{EMG}^{Norm} ([1200, 50, 8])
	D_{EMG} ([1200, 50, 8])	Batch Normalization	D_{EMG}^{Norm} ([1200, 50, 8])
	S_{GYR} ([1200, 50, 3])	Batch Normalization	S_{GYR}^{Norm} ([1200, 50, 3])
	D_{GYR} ([1200, 50, 3])	Batch Normalization	D_{GYR}^{Norm} ([1200, 50, 3])
	S_{EMG}^{Norm} ([1200, 50, 8])	B-LSTM + Reshape +Dropout	F_{EMG}^{S} ((50, [1200, 128]))
	D_{EMG}^{Norm} ([1200, 50, 8])	B-LSTM + Reshape +Dropout	F_{EMG}^{D} ((50, [1200, 128]))
	S_{GYR}^{Norm} ([1200, 50, 3])	B-LSTM + Reshape +Dropout	F_{GYR}^{S} ((50, [1200, 128]))
	D_{GYR}^{Norm} ([1200, 50, 3])	B-LSTM + Reshape +Dropout	F_{GYR}^{D} ((50, [1200, 128]))
	F_{EMG}^{S} ((50, [1200, 128])) + F_{EMG}^{D} ((50, [1200, 128]))	Concatenation	F_{EMG} ((50, [1200, 256]))
	F_{GYR}^{S} ((50, [1200, 128])) + F_{GYR}^{D} ((50, [1200, 128]))	Concatenation	F_{GYR} ((50, [1200, 256]))
	F_{Global} ([1200, 64, 15])	Reshape + Perceptron	F_{Global} ([1200, 128])
	F_{EMG} ((2, [1200, 256])) + F_{GYR} ((2, [1200, 256])) +F_{Global} ([1200, 128])	Attention	$Context^{t}$ ([1200, 512])
	$Context^{t}$ ([1200, 512])	LSTM	$F_{Overall}^{t}$ ([1200, 128])
Hand pose Estimator	$F_{Overall}^{t}$ ([1200, 128]) +$Context^{t}$ ([1200, 512]) +F_{Global} ([1200, 128])	Multilayer Perceptron + Dropout	$Skeleton^{t}$ ([14, 1200, 3])
User Discriminator	$F_{Overall}^{t}$ ([1200, 128]) +$Context^{t}$ ([1200, 512]) +F_{Global} ([1200, 128])	Multilayer Perceptron + Dropout	$User_{ID}^{t}$ ([1200, 10])