# NeuroPose: 3D Hand Pose Tracking using EMG Wearables

Yilin Liu
The Pennsylvania State University
University Park, PA, USA

Shijia Zhang
The Pennsylvania State University
University Park, PA, USA

Mahanth Gowda
The Pennsylvania State University
University Park, PA, USA

## ABSTRACT

Ubiquitous finger motion tracking enables a number of exciting applications in augmented reality, sports analytics, rehabilitation-healthcare, haptics etc. This paper presents *NeuroPose*, a system that shows the feasibility of 3D finger motion tracking using a platform of wearable ElectroMyoGraphy (EMG) sensors. EMG sensors can sense electrical potential from muscles due to finger activation, thus offering rich information for fine-grained finger motion sensing. However converting the sensor information to 3D finger poses is non trivial since signals from multiple fingers superimpose at the sensor in complex patterns. Towards solving this problem, *NeuroPose* fuses information from anatomical constraints of finger motion with machine learning architectures on Recurrent Neural Networks (RNN), Encoder-Decoder Networks, and ResNets to extract 3D finger motion from noisy EMG data. The generated motion pattern is temporally smooth as well as anatomically consistent. Furthermore, a transfer learning algorithm is leveraged to adapt a pretrained model on one user to a new user with minimal training overhead. A systematic study with 12 users demonstrates a median error of 6.24° and a 90%-ile error of 18.33° in tracking 3D finger joint angles. The accuracy is robust to natural variation in sensor mounting positions as well as changes in wrist positions of the user. *NeuroPose* is implemented on a smartphone with a processing latency of 0.101s, and a low energy overhead.

## CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile computing design and evaluation methods**; • **Computing methodologies** → *Neural networks.*

## KEYWORDS

IoT; Wearable; EMG; Hand pose tracking

**ACM Reference Format:**
Yilin Liu, Shijia Zhang, and Mahanth Gowda. 2021. NeuroPose: 3D Hand Pose Tracking using EMG Wearables. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021 ,Ljubljana, Slovenia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3442381.3449890

## 1 INTRODUCTION

3D finger pose tracking enables a number of exciting applications in sports analytics [6], healthcare and rehabilitation [80], sign languages [19], augmented reality (AR), virtual reality (VR), haptics [74] etc. Analysis of finger motion of aspiring players can be compared to experts to provide automated coaching support. Finger motion stability patterns are known to be bio-markers for predicting motor neuron diseases [27]. AR/VR gaming as well as precise control of robotic prosthetic devices are some of the other applications that benefit from 3D finger pose tracking [18, 60].

Web-based augmented/virtual reality applications are becoming popular [37, 49] leading to standardizations of WebXR APIs [85]. Examples include remote surgery, virtual teaching (body-anatomy, sports, cooking etc), multiplayer VR gaming. These applications involve augmenting the context of the user (location, finger-pointing direction etc.) with information from the web (on-screen-viewport, textual-information, haptic stimulation etc.). Finger motion tracking is a common denominator of such applications.

Motivated by the above applications, there is a surge in recent works [23, 59] in computer vision that track 3D finger poses from monocular videos. Given they do not require depth cameras, the range of applications enabled is wide. However, vision based techniques are affected by issues such as occlusions and the need for good lighting conditions to capture intricate finger motions.

In contrast to vision, the main advantage of wearables is in enabling ubiquitous tracking without external infrastructure while being robust to lighting and occlusions. While data gloves [1, 2, 5] with IMU, flex, and capacitative sensors have been popularly used for finger motion tracking, it is shown that gloves hinder with dexterous hand movement [73]. As alternatives to putting sensors on fingers, sensing at wrist with surface acoustic [89], capacitative [81], bioimpedance [91], ultrasonography [57], wrist pressure[29] etc., has been explored, but the sensing is only limited to tens of gestures. Beyond discrete gestures, infrared [44] and thermal cameras [41] mounted on wrist have been explored for continuous 3D pose tracking, but has limitations on hand motion (details in Section. 6). In contrast, we explore using ElectroMyoGraphy (EMG) sensors worn like a band on the forearm (Fig. 5) with the following advantages: (i) Captures information directly from muscles that activate finger motions, thus offering rich opportunities for continuous 3D finger pose sensing (ii) A user does not need to put sensors on fingers and thus she is able to perform activities requiring fine precision (iii) Tracking is independent of ambient conditions of lighting or presence of objects in the background. (iv) EMG sensors can measure emotions (like fear) and muscle strain to make VR tasks on safety (fire, construction etc) and physical-activities (e.g. rock climbing) more realistic [34, 88].

Despite the benefits, EMG sensors are not as popular as smartphones or smartwatches. Thus the user needs to carry a separate EMG band with her. Nevertheless, we believe there are motivating applications (prosthetic devices for amputees, sports coaching, augmented reality) where a user can selectively wear the device when needed instead of constantly wearing it. The prospects of adoption of EMG sensing for AR/VR is on the rise (led by Facebook [3, 11])


*WWW '21, April 19–23, 2021, Ljubljana, Slovenia*

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3449890

because EMG can pick strong/unambiguous signals of minute finger motion. Thus, we believe understanding the limits and bounds of sensing can help develop interesting applications and use-cases encouraging better social adoption.

Prior works on EMG based finger motion tracking are limited to tracking a few hand gestures [28, 32, 71, 72, 76], or tracking hand poses over a set of discrete gesture related motions [71, 76]. They do not provide free form 3D pose tracking for arbitrary hand motion. This paper proposes a system called *NeuroPose* that fills this gap in literature by designing a EMG wearable-based 3D finger pose tracking technology. Towards this end, *NeuroPose* uses an off the shelf armband consisting of 8 EMG channels (Fig. 5) for capturing finger motion and converting it into 3D hand pose as depicted in Fig. 1. Using only two of the eight channels might increase the comfort of wearing the sensor with a modest loss in accuracy.

Briefly, the EMG sensors capture neural signals propagating through the arms due to finger muscle activations. Each finger muscle activation generates a train of neuron impulses, which are the fundamental signals captured by the sensors (more details in Sec. 2). Given such EMG sensor measurements, tracking the 3D pose is non-trivial and introduces a number of challenges: (i) Human hand is highly articulated with upto 21 degrees of freedom from various joints. The complexity of this search space is comparable to tracking joints in the skeletal model of a human body. (ii) Impulses from multiple fingers are mixed in complex non-linear patterns making it harder to decouple the effect of individual fingers from the generated sensor data. (iii) The strength of the captured signals depends on the speed of motion, and finger pose. (iv) The nature of captured data varies across users due to variations in body sizes, anatomy etc. (v) The sensor data is noisy due to hardware imperfections.

In handling the above challenges, *NeuroPose* exploits a number of opportunities. (i) Finger motion patterns are not random but they follow tight anatomical constraints. Fusion of such constraints with the actual sensor data dramatically reduces the search space. (ii) Innovation in machine learning (ML) algorithms that explicitly and implicitly fuse such constraints with sensor data have been exploited. In particular, *NeuroPose* explores architectures in Recurrent Neural Networks (RNN)[58], Encoder-Decoder[21], ResNets[40] in achieving a high accuracy. (iii) A transfer learning framework based on *adaptive batch normalization* is exploited to learn user dependent features with minimal overhead for adapting a pretrained model to a new user for 3D pose tracking.

*NeuroPose* is implemented on a smartphone and runs with a latency of 0.1s, with low power consumption. A systematic study with 12 users achieves an accuracy of 6.24° in median error and 18.33° in the 90%-ile case. The accuracy is robust to natural variation in sensor mounting positions as well as changes in wrist positions of users. Our contributions are summarized below:

*(1) NeuroPose shows the feasibility of fine grained 3D tracking of 21 finger joint angles using EMG devices for arbitrary finger motions.*

*(2) Fusion of anatomical constraints with sensor data into machine learning algorithms for higher accuracy.*

*(3) Implementation on embedded platforms and extensive evaluation over diverse users.*

## 2 BACKGROUND

We begin with a brief overview of: (i) the anatomical model of the human hand (ii) the neuro-muscular interactions during finger muscle activations and how it manifests as EMG sensor data.

### 2.1 Hand Skeletal Model

The human hand consists of four fingers and a thumb which together exhibit a high degree of articulation. Fig.2(a) depicts the skeletal structure of the hand with various joints that are responsible for complex articulation patterns that generate 3D hand poses. Fig. 2(b) shows a simplified kinematic view. The four fingers consist of MCP (metacarpophalangeal), PIP (proximal interphalangeal), and DIP (distal interphalangeal) joints. The joint angles at PIP ($\theta_{pip}$) and DIP ($\theta_{dip}$) joints exhibit a single degree of freedom (DoF) and can flex or extend (Fig.2(c)) the fingers towards or away from the palm. In addition to flexing, the MCP joint can also undergo adduction and abduction (side-way motions depicted in Fig.2(c)), and thus possesses two DoFs, denoted by $\theta_{mcp,f/e}$, and $\theta_{mcp,aa}$ respectively. Thus, each of the four fingers posses four DoF. The thumb on the other hand exhibits a slightly different anatomical structure in comparison to the other four fingers. The IP (interphalangeal) joint can flex or extend with a single DoF ($\theta_{ip}$). The MCP and TM (trapeziometacarpal) joints possesses both flex and abduction/adduction DoF, thus the thumb has five DoF – $\theta_{ip}$, $\theta_{mcp,f/e}$, $\theta_{mcp,aa}$, $\theta_{tm,f/e}$, and $\theta_{tm,aa}$. The other 6 DoF comes from the motion of palm including translation and rotation. We ignore the motion of the palm in this paper and only focus on tracking fingers which together have 21 DoF – modeled as 21 dimensional space ($\mathbb{R}^{21}$). Thus, *NeuroPose*'s goal is to track this $\mathbb{R}^{21}$ dimensional space to capture the 3D finger pose.

The various joint angles responsible for finger articulation exhibit a high degree of correlation and interdependence [25, 52]. Some of the intra-finger constraints are enumerated below:

$$\theta_{dip} = \frac{2}{3}\theta_{pip} \tag{1}$$

$$\theta_{ip} = \frac{1}{2}\theta_{mcp,f/e} \tag{2}$$

$$\theta_{mcp,f/e} = k\theta_{pip}, \quad 0 \leq k \leq \frac{1}{2} \tag{3}$$

Equation 1 suggests that in order to bend the DIP joint, the PIP joint must also bend under normal finger motion (assuming no external force is applied on the fingers). Likewise, Equation 2 is a constraint on the thumb joints. Similarly, the range of motion for PIP is very much limited by the MCP joint (Equation 3). The generic range of motion constraints for other fingers are enumerated below:

$$\begin{aligned} -15^\circ \leq \theta_{mcp,aa} \leq 15^\circ \\ 0^\circ \leq \theta_{dip} \leq 90^\circ \\ 0^\circ \leq \theta_{pip} \leq 110^\circ \end{aligned} \tag{4}$$

Clearly, abduction/adduction angles have a smaller range of motion compared to flex/extensions. In addition to these constraints, there are complex inter-dependencies between finger joint motion patterns which cannot be captured by well formed equations. However, our ML models will be able to automatically learn such constraints from data and exploit them for high accuracy tracking.

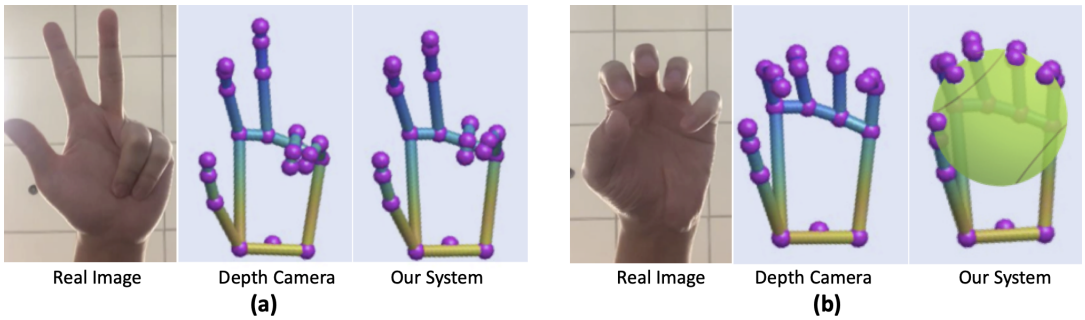


**Figure 1: A comparison between a real image, a depth camera, and *NeuroPose*. Tracking of fine grained hand poses can enable applications like: (a) Word recognition in sign languages (b) Augmented reality by enhancing the tracking output. A short demo is here [15].**

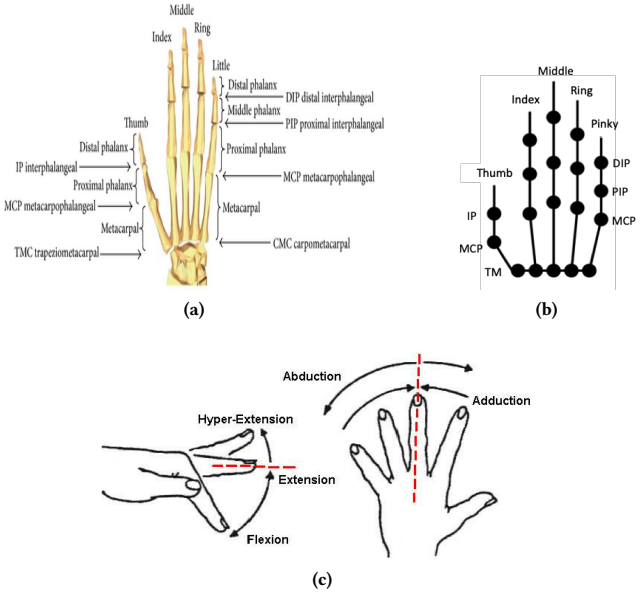


**Figure 2: (a) Anatomical details of the hand skeleton [25] (b) Kinematic structure and joint notations [52] (c) Finger motions include flex/extensions and abduction/adductions [68]**

## 2.2 Electromyography Sensor Model

Electromyography sensors can detect electrical potential generated by skeletal muscles due to neurological activation. Such signals can provide information regarding temporal patterns and morphological behaviour of motor units that are active during muscular motion [78]. Not only are the signals useful for detecting and predicting body motion induced by the muscles but also useful for diagnosis of various neuromuscular disorders and understanding of healthy, aging, or fatiguing neuromuscular systems.

**Muscles of Interest:** We now provide a brief overview of muscular involvement during finger motions. Several muscles are involved in performing finger motions. Fig. 3(a) and (b) depict the anatomical structure of the human arm. *Extensor Pollicis Longus* extends the thumb joints whereas *Abductor Pollicis Longus and Brevis* performs thumb abductions. *Extensor Indicis Proprius* extends the index finger. *Extensor Digitorum* extends the four medial fingers and *Extensor Digiti Minimi* extends the little finger. *Volar interossei* and *Dorsal interossei* group of muscles are responsible for adduction and abduction respectively of index, ring, and little fingers towards/away from the middle finger. They are connected to *proximal phalanx* and the *Extensor digitorum*. *NeuroPose* mainly focuses on such muscles that perform finger actions. Other muscles that are involved in large scale motion and supporting strength include *Supinator* for forearm motion, *Anconeus and Brachioradiali* for elbow joint, *Extensor Carpi Ulnaris*, *Extensor Carpi Radialis Longus and Brevis* for wrist joint etc.

**Feasibility of Tracking the Muscles of Interest:** Among the targeted muscles of interest, although some of them appear close to the skin surface, some of them are deep (such as Extensor Indicis). Therefore, a natural question to ask is: *Is surface EMG alone sufficient to capture all such muscles of interest?* To verify this, we conduct a simple experiment where we flex and extend each of the five fingers, and observe the activity on the EMG channels. Depicted in Fig. 4, all fingers show noticeable activity on the EMG channels for flex/extensions (the activity on channel number 1 is shown per conventions in Fig. 5.) For sake of brevity, we provide one example for abduction/adduction in Fig. 4(f) for abducting/adducting all fingers together however, we note that each finger individually generates a noticeable pattern for abduction/adduction motions. An important observation from the figures is that the muscle group responsible for motion of index finger – *Extensor Indicis*, a non-surface muscle group relative to sensor placement in Fig. 5 – also generates a noticeable spike in the EMG channel data (Fig. 4(b)). This is also validated by prior research related to deep muscle activity [46]. These signals must be carefully analyzed further to capture the precise magnitude of finger joint angles, particularly when multiple fingers are simultaneously in motion. Towards the end, we begin by describing the interference pattern on the EMG sensors by signals from different muscle groups. Separating out the individual finger motions from such EMG sensor data will be discussed in Section 4.

**Biological Model:** We now provide a brief description of the biological model of EMG signals generated due to muscle activations (illustration in Fig. 3(c)). Muscles consist of fundamental units called muscle fibres (MF) which are the primary components responsible for contraction. Activation of an MF by the brain results in propagation of an electrical potential called action potential (AP) along the MF. This is called motor fibre activation potential (MFAP). The MFs are not excited individually but are activated together in groups called motor units. Groups of motor units coordinate

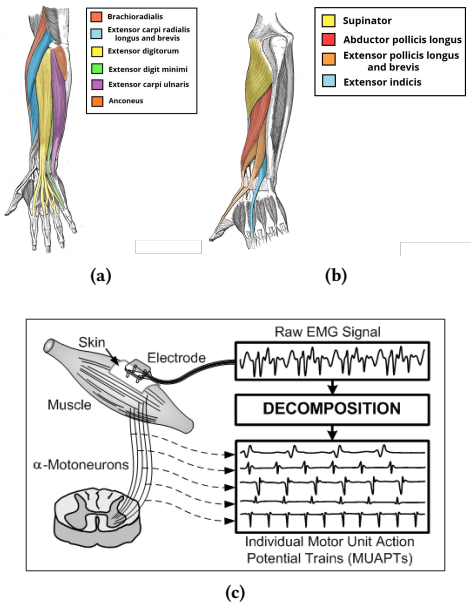


**Figure 3: (a) and (b) Anatomical details of forearm muscles [4] (c) EMG signals from an electrode can be decomposed into constituent motor unit action potential trains (MUAPT) [9]**

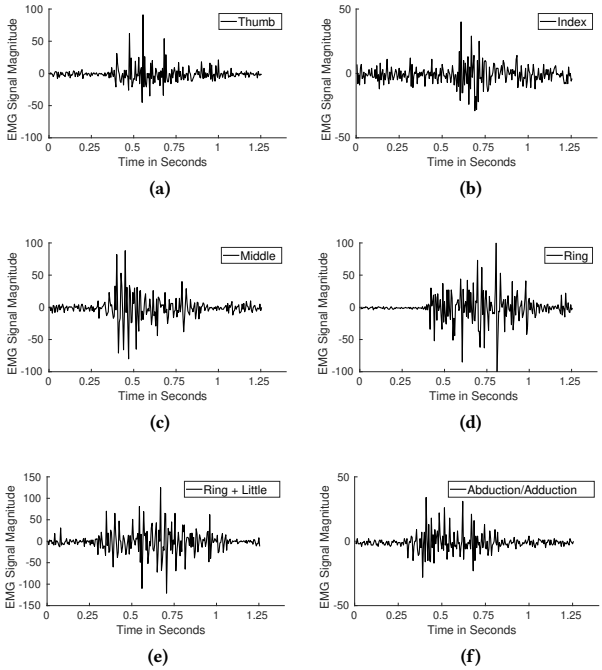


**Figure 4: Flex/extension motion of all fingers generate noticeable "spike" in the EMG data for (a) Thumb (b) Index (c) Middle (d) Ring (e) Little + Ring (the little finger cannot be flexed without jointly moving the ring finger ) (f) Abduction/Adduction of all fingers**

together to contract a single muscle. Individual MFAPs cannot be detected separately, instead summation of all MFAPs within the motor unit generates a signal called as motor unit action potential (MUAP) as shown in the below equation

$$MUAP_j(t) = \sum_{i=1}^{N_j} MFAP_i(t - \tau_i)s_i, \tag{5}$$

where $\tau_i$ is the temporal offset, $N_j$ is the number of fibres in motor unit $j$, and $s_i$ is a binary variable indicating whether or not the muscle fibre is active. The temporal offset depends on the location of the muscle fibre. The number of observed MFAPs within a MUAP also depends on location of EMG electrode because the potential generated by far away fibres are typically detected in attenuated form at the electrode. A similar muscle action can result in different shape of the generated MUAP signal depending on the previous state of the muscle as well as the temporal offset $\tau_i$ which can vary.

The above equation represents a single instance of firing, but the motor units must fire repeatedly to maintain the state of muscle activation. Continuous muscle activations can generate a train of MUAP impulses separated by inter discharge intervals (IDI), as depicted in the below equation

$$MUAPT_j(t) = \sum_{k=1}^{M_j} MUAP_{jk}(t - \delta_{jk}), \tag{6}$$

where $M_j$ is the number of times the $j$th motor unit fires, $\delta_{jk}$ is the $k$th firing time of the $j$th motor unit.

Finally, the electric potential detected at an EMG electrode is the superimposition of signals by spatially separated motor units and their temporal firings patterns dependent on their respective IDIs. This spatio-temporal superimposition is depicted in the below equation where $n(t)$ is the noise term, and $N_m$ is the number of active motor units.

$$EMG(t) = \sum_{k=1}^{N_m} MUAPT_j(t) + n(t). \tag{7}$$

While in theory, the EMG signal is composed of activation from every single muscle fibre, in practice the electrode can only detect the signals from fibres closer to the electrode because the signals attenuate below noise level with distance. Our EMG sensor platform described next exploits multiple electrodes to capture activations of all fibres involved in finger motion. Once the EMG data is captured, the core technical challenge is in decomposing the signals into activations responsible for individual joint movements. Towards this, we introduce ML algorithms in Section 4 for signal decomposition.

## 3 PLATFORM DESCRIPTION

Our platform includes a MYO armband depicted in Fig. 5 worn on the arm. It consists of 8 EMG channels, as well as Inertial Measurement Unit (IMU) sensors of accelerometers, gyroscopes, and magnetometers. The data is streamed wirelessly over bluetooth to a desktop/smartphone device. *NeuroPose* is implemenetd on a sony xperia z3 dual smartphone that captures the EMG data and provides finger motion tracking results. The MYO sensor is low-cost, and appears to be solidly built. Although the MYO armband fits perfectly aesthetically on the arm it might seem intrusive for some users. Towards minimizing the intrusiveness of the platform, *NeuroPose*'s implementation with only a 2-channel EMG data offers a low-intrusive option with a modest loss in accuracy (Section 5).

**Skin Temperature Calibration:** The EMG amplitude may be slightly affected by skin temperature variations [86]. The surface

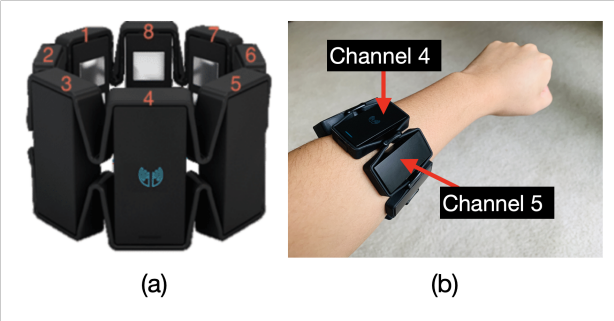


**Figure 5: (a) 8 channel Myo armband (b) Myo armband in action**

Myo platform warm ups the contacted muscle [10] slightly. This helps the sensor to form a stronger electrical connection with the muscles to minimize the effects of temperature.

**Other Platforms:** We note that unlike smartwatches or smartphones, there is no globally acceptable platform for EMG sensing yet. Facebook has recently acquired patents related to MYO armband [3, 11] for developing finger tracking technology for its thrust towards AR/VR applications. Other form factors ranging from arm-bands, tattoos, and arm-gloves have been proposed by both academia and industry with no consensus on what is best [16, 28, 69, 72, 77]. Therefore, the ML models developed in this paper may not apply directly to a hardware of different form factor than what is used here. While there are uncertainities about what platforms will gain wide spread adoption, our goal is to show that enough information exists in surface EMG data for continuous tracking of arbitrary finger motions. Furthermore, by showing the right applications and use-cases, we believe we can influence the process of convergence of hardware platforms.

## 4 CORE TECHNICAL MODULES

We explore multiple ML models for 3D finger motion as elaborated in this section.

### 4.1 Encoder Decoder Architecture

In order to generate plausible finger pose sequences with spatial constraints across fingers, as well as temporally smooth variations over time, we design an encoder-decoder network as illustrated in Fig. 6. Specifically, the network captures a holistic view of a large interval of time-series sensor data instead of a single sensor sample. This enables the network to enforce and learn the key spatio-temporal constraints as well as consider historical EMG data while making hand pose inferences. The network accepts 5$s$ of sensor data and outputs the corresponding 3D hand pose sequence. The various components of the architecture are elaborated next.

**Encoder:** The encoder-decoder model maps a sequence of input EMG data to a sequence of 3D finger poses. Unlike discrete classes, the output space of the model is a continuous domain $\mathbb{R}^{21}$. Among these 21 dimensions, 5 of the dimensions ($\theta_{dip}$ for four fingers and $\theta_{ip}$ for thumb) can be directly computed using Equations 1, 2. Thus, the actual output of the network is only 16 dimensions – $\mathbb{R}^{16}$.

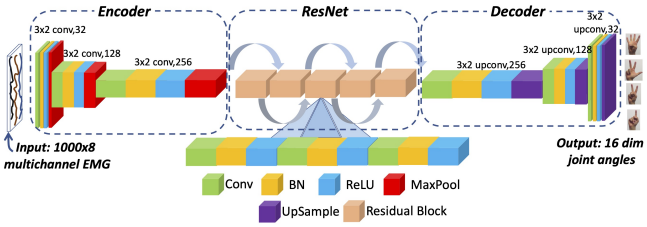


**Figure 6: Encoder Decoder Architecture used in *NeuroPose*:**

While one possibility is to build a network with a series of convolutional layers, this will increase the number of parameters in the network, thus causing issues not only in compute complexity and memory but also in convergence. Thus, the encoder uses a series of downsampled convolutional filters. This captures a compact representation of the input which will later be used by the decoder in generating 3D hand poses.

The input $x$ to the encoder is a multi-channel EMG data of dimensions $T \times 8$, where we choose $T = 1000$, which at a sampling rate of 200Hz translates to a duration of 5s. The encoder consists of a series of CONV-BN-RELU-MAXPOOL layers, which are elaborated below: (i) The *CONV* sub-layer includes 2D convolutional filters that perform a basic convolution operation[48]. The *CONV* sub-layer extracts spatio-temporal patterns within EMG data to learn features representative of finger motions. (ii) This is followed by a batch normalization (*BN*) sub-layer whose role is to accelerate convergence of the model by controlling huge variations in the distribution of input that passes from one layer to the next [42]. (iii) The *BN* module is followed by an *activation* sub-layer, which applies an activation function to the output of the *BN* layer. We chose a Rectified Linear Unit *(ReLU)* activation function [20]. While non-linearities are critical in training a deep neural network, among possible alternatives ReLU is popular because of its strong biological motivation, practicality of implementation, scale in-variance, better gradient propagation etc. We also add dropouts [83] following RELU activations. They serve as an adaptive form of regularization which knocks off some of the parameters of the network with a random probability of 0.05. (iv) Finally, max-pooling is applied to the output so as to downsample the feature size toward reaching a compact feature representation of the EMG data. Max pooling is done by applying a max filter to non-overlapping subregions of the initial representation. For example, a max-pool filter of size $2 \times 2$ applied to an input of size $100 \times 100$, will slide a non-overlapping window of size $2 \times 2$ and extracts the maximum element from the input at each overlap resulting in an output of size $50 \times 50$.

The first of the CONV-BN-RELU-MAXPOOL layers applies 32 2D-*CONV* filters of size $3 \times 2$, and down samples the feature sizes by 5 and 2 over temporal and spatial (EMG channels) domains. Similarly, the filter sizes and number of filters of the other layers is depicted in Fig.6. The second and third layers down-sample by $(4 \times 2)$, and $(2 \times 2)$ over time and space. Thus, the final output of the encoded representation is of dimensions $25 \times 1 \times 256$. The decoder processes this encoded data to obtain finger joint angles.

**Residual Blocks:** A natural question to ask is: *Why not increase the depth of the network to extract stronger feature representations?* Unfortunately, deeper networks are harder to optimize and they

also pose challenges in convergence. ResNets[40] proposed a revolutionary idea of introducing skip connections between layers so as to balance this tradeoffs between stronger feature representations and convergence. The skip connections, also called as residual connections provide shortcut connections between layers as shown in the middle of the network in Fig. 6. Suppose, $y$, and $x$, denote the intermediate representations at different layers in the network, with $y$ being deeper than $x$ with a few layers in between. Then, the skip connections are denoted by the below equation.

$$y = f(x, W_l) + x \tag{8}$$

$f(x, W_l)$ denotes the intermediate layers between $x$, and $y$. Because of the existence of a shortcut path between $y$ and $x$, the representation at $x$ is directly added to $f(x, W_l)$. Therefore, the network can choose to ignore $f(x, W_l)$, and exploit the shortcut connection $y = x$ to first learn a basic model. As the network continues to evolve, it will exploit the deeper layers ($f(x, W_l)$) in between shortcut connections to learn stronger features than the basic model. As shown in Fig. 6, we incorporate ResNets in between the encoder and decoder part of the network. As evaluated in Sec. 5, this design choice plays a critical role in achieving a high accuracy.

**Decoder:** The decoder maps the encoded representations into 3D hand poses. The decoder uses upconvolutional layers to upsample and increase the size of the encoded representation to match the shape of the output. The decoder network consists of a series of CONV-BN-RELU-UPSAMPLE layers. Each such layer consists of following sub-layers. (i) The CONV layer tries to begin making progress towards mapping the encoder representations into joint angles. The job of (ii) BN sub-layer, and (iii) RELU activation sub-layer is similar to their roles in the encoder. (iv) The upsampling sub-layer's job is to increase the sampling rate of the feature representations. Upsampling (with nearest neighbor interpolation method [39]) across multiple layers will gradually increase the size of the compact encoder features to match the size of the output.

The size and number of conv filters in the decoder at each layer is shown in Fig. 6. The three layers of the decoder upsample by factors of $(5 \times 4)$,$(4 \times 2)$,$(2 \times 2)$ respectively on temporal and spatial domains thus matching the output shape of $1000 \times 16$ at the last layer. Finally, the decoder output is subject to a Mean Square Error (MSE) loss function as elaborated next to facilitate training.

**Loss Functions and Optimization:** In all equations below, $\hat{\theta}$ denotes the prediction by the ML model, whereas $\theta$ denotes the training labels from a depth camera (leap sensor [7]).

$$loss_{mcp,f/e} = \sum_{i=1}^{i=4} (\hat{\theta}_{i,mcp,f/e} - \theta_{i,mcp,f/e})^2 \tag{9}$$

$$loss_{pip} = \sum_{i=1}^{i=4} (\hat{\theta}_{i,pip} - \theta_{i,pip})^2 \tag{10}$$

$$loss_{mcp,a/a} = \sum_{i=1}^{i=4} (\hat{\theta}_{i,mcp,aa} - \theta_{i,mcp,aa})^2 \tag{11}$$

The above equations capture the MSE loss in prediction of joint angles of MCP (flex/extensions and adduction/abduction), and PIP joints of the four fingers.

$$\begin{aligned} loss_{thumb} = (\hat{\theta}_{th,mcp,aa} - \theta_{th,mcp,aa})^2 + \\ (\hat{\theta}_{th,mcp,f/e} - \theta_{th,mcp,f/e})^2 + (\hat{\theta}_{th,tm,aa} - \theta_{th,tm,aa})^2 + \\ (\hat{\theta}_{th,tm,f/e} - \theta_{th,mcp,f/e})^2 \end{aligned} \tag{12}$$

The above equations capture the MSE loss in the MCP and TM joints of the thumb.

$$loss_{smoothness} = ||(\nabla\hat{\theta}_t - \nabla\hat{\theta_{t-1}})||_2^2 \tag{13}$$

The above equation enforces constant velocity smoothness constraint in the predicted joint angles where $\theta_t$ above is a representative vector of all joint angles across all fingers at a time step $t$.

The overall loss function is given by the below equation.

$$\begin{aligned} loss = loss_{mcp,f/e} + loss_{mcp,aa} + \\ loss_{pip} + loss_{thumb} + loss_{smoothness} \end{aligned} \tag{14}$$

Note that the loss function does not include $\theta_{dip}$ or $\theta_{ip}$ because we compute them directly from anatomical constraints: Equations 1, 2.

**Finger motion range constraints:** As described in Section 2, each finger joint has a certain range of motion for both flex/extensions and abduction/adductions. In order to apply these constraints, we first normalize the predicted output of a joint angle by dividing it by the range constraint (for example, by 90° for $\theta_{dip}$). We then apply the bounded ReLU activation (bReLU) function [50] to the last activation layer in our network. The bReLU adds an upper bound to constrain its final output. The bReLU outputs are multiplied again with their range constraints such that the unit of the output is in degrees. The bReLU, in conjunction with other loss functions based on temporal constraints (Equation 13) facilitates predicting anatomically feasible as well as temporally smooth tracking results.

## 4.2 Transfer Learning with Semi Supervised Domain Adaptation

For the encoder-decoder model proposed above, training separate models for each user will be burdensome. Therefore, we explore domain adaptation strategies to *pretrain* a model with one (*source*) user and *fine-tune* it to adapt to new users with low training overhead.

Transfer-learning based domain adaptation is popular in vision and speech processing. For example, AlexNet model [47] pretrained on ImageNet database [30] was fine-tuned for classifying images in medical domain[94], remote-sensing [38] and breast-cancer [62]. Similarly, a pre-trained BERT language model [31] was fine-tuned for tasks such as text-summarizing [90], question answering [70] etc. This significantly reduces the burden of training for a new task. In a similar spirit, we use pretrained model from one user and fine-tune it for a different user to significantly decrease the training overhead (Fig. 14(a)) without losing much of accuracy.

At a high level, we exploit domain adaptation at the Batch Normalization (BN) layers. Given the sufficient success of BN layers in accelerating convergence by minimizing *covariate shift* [42] with a relatively fewer number of parameters, we exploit them towards domain adaptation as well. The success of this approach has already been shown in other domains such as computer vision [24, 56].

Our domain adaptation process is performed as enumerated below: (i) We generate a model for one user by extensively training the model with labelled data from that user – known as the *pretrain*ed model. (ii) We collect small training data with only few labels from the new (*target*) user. Instead of developing the model for the *target* user from scratch, we initialize the model weights to be same as the *pretrain*ed model. (iii) We make all layers in the model untrainable

except the Batch Normalization (*BN*) layers. Using the few labels from the *target* user, we update the BN layers to minimize the loss function. This is called *fine tuning*. The model thus generated will be used for making inferences on the *target* user.

*Finetuning* the BN layers help with domain adaptation because of their ability to contain wide oscillations in the distributions of input fed from one layer to the next. Given the sufficient success in BN layers (with only a few parameters) for accelerating convergence by minimizing *covariate shift* [42], we exploit them towards domain adaptation as well. The BN layers will learn to sufficiently transform the distribution from *target* user to a distribution of the *source* user on which the model is *pretrained* on. If successful, the *pre-trained* model from the *source* user can be used for performing inferences on the target user with the *finetuning* steps discussed here. As discussed in Section 5, this results in reduction of training overhead on the *target* user by an order of magnitude.

### 4.3 RNN Architecture

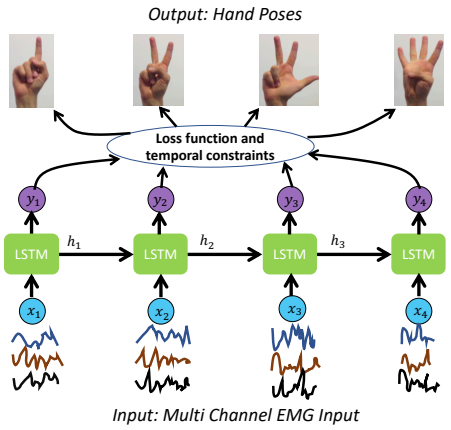


**Figure 7: RNN alternative explored in this paper.**

The encoder-decoder model proposed above has a holistic view of a relatively long interval (5s) of sensor data, and thus can exploit complex spatio-temporal relationships. However, in order to ensure real-time performance with this model, we need to constantly process previous 5s of data at any given instant. Although this can ensure real-time performance, the power consumption can be higher due to redundant computations. Therefore, we explore an alternative model with Recurrent Neural Networks (RNN) to obtain real-time performance without redundant computation.

Our model is presented in Fig.7. The generated EMG sensor data is not only dependent on muscle contractions to maintain the current finger pose but also dependent on the force exerted in the muscles to move the fingers to a new position. Such temporal dependencies can be systematically modeled with a recurrent neural network (RNN). Each RNN unit accepts as inputs one sample of an eight channel EMG data as well as previous hidden state. In particular, we use the Long Short Term Memory (LSTM) variant of RNN because of its ability to handle vanishing/expanding gradients [67] and selectively forgetting/remembering features from past. It outputs an $\mathbb{R}^{16}$ dimension finger joint angles and a new hidden state to be used as input in the next iteration of the RNN unit.

During training, the outputs are subjected to MSE loss functions, as well as temporal constraints identical to ones used in encoder-decoder architecture. We use truncated backpropagation through time (TBPPT [43]) in training with a truncation of 64 time units.

## 5 PERFORMANCE EVALUATION

Our experiments are designed to comprehensively test the robustness to sensor positions, usability, and accuracy of *NeuroPose* over users, joint angles etc. We also compare various ML models, overall training cost as well as perform system level measurements for efficiency of implementation on smartphones.

### 5.1 User Study

We conduct a study with 12 users (8 males, 4 females). The users are aged between 20-30, and weigh between 47-96kgs.

**Data Collection Methodology:** The users wear the Myo armband as shown in Fig.5 on the left hand in a position where it fits naturally, with channel number 4 on top. The users were then instructed to perform random finger motions that include flexing or extending of fingers as well as abduction or adduction thus incorporating all range of possible hand poses. Under the guidance of a study team member, we let the users practice finger motions before the study to ensure that the user moves all fingers over the entire range of motion. This ensures good convergence of the ML models as well as generalizability to arbitrary finger motions. There are no discrete classes of gestures. The motion patterns are entirely arbitrary thus making the data collection easier.

**Labels for Training and Testing:** The collected data includes 8 EMG channels from the Myo sensor as well as the fingers' 3D co-ordinates and joint angles captured by leap motion sensor [7]. While the Myo sensor provides EMG data for 3D pose tracking, the leap sensor data serves as the ground truth for validation as well as provides labels for training *NeuroPose*'s ML models. These labels include joint angles for each finger. The EMG and leap data were synchronized using Coordinated Universal Time (UTC) timestamps. Since *NeuroPose* performs continuous finger tracking instead of identifying discrete gestures, we use MSE (instead of cross-entropy) between predicted joint angles (from Myo) and leap (ground truth) for training and testing.

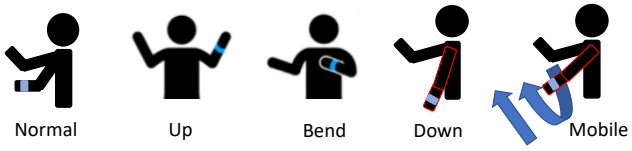


**Figure 8: Wrist Configuration Map**

**Training Data Collection :** Each user participates in 12 separate sessions with each session lasting for 3 minutes, with sufficient rest between sessions. For the first 5 sessions, both the sensor position and the wrist position are not changed (wrist maintained at the "normal" position depicted in Fig. 8). For the each of the last 6 sessions, we remove and remount the sensor. For the $6^{th}$ session, we let the user place the wrist still in the normal position. However, for the last 6 sessions, we let the user place the wrist in 4 different configurations (*up*, *down*, *bend*, *mobile*) as indicated in Fig. 8. In the *mobile* configuration, the wrist was moved up and down

including rotations of the wrist within the tracking range of the leap sensor. Users perform *up*, *down*, *bend*, *down*, *up* for sessions 7-11 respectively. For the last session, the users perform the *mobile* configuration. The position of the leap sensor was adjusted using a tripod so that it can capture the ground truth. This data is used for developing two kinds of models. (i) **User-dependent model:** A model for each user that requires 900 seconds of training data from the first 5 sessions of that user. (ii) **Model with domain adaptation:** A model for each user where a pre-trained model from a different user is taken and fine-tuned using techniques in Section 4.2 such that only a small fraction (90 seconds) of user-specific training data is used for developing a model for the user. (iii) **Model without domain adaptation or user-independent model:** Here, we use the trained model from one user directly to perform inferences on a new user without any training data from the new user. (iv) **Multi-user model:** This is also a user-independent model. Here, we train a model based on training data from multiple users. The trained model is directly used for inferences on a new user without any training data from the new user.

**Test Data:** Using the models developed above, we evaluate the joint angle prediction accuracy over test cases that include the last 6 sessions where (i) The sensor has been removed and remounted on the user's arm (ii) The wrist position is completely different from the one used to train the models.

## 5.2 Implementation

*NeuroPose* is implemented on a combination of desktop and smartphone devices. The ML model is implemented with TensorFlow [17] packages and the training is performed on a desktop with Intel i7-8700K CPU, 16GB RAM memory, and Nvidia GTX 1080 GPU. We use the Adam optimizer[45] with a learning rate of 1e-3, $\beta_1$ of 0.9 and $\beta_2$ of 0.999. To avoid over-fitting issues that may happen in the training process, we apply the L2 regularization[22] on each CONV layer with a parameter of 0.01 and also add dropouts[83] with a parameter of 0.05 following each RELU activations. Once a model is generated from training, the inference is done entirely on a smartphone device using TensorFlowLite [36] on a sony xperia z3 with a Quad-core 2.5 GHz Krait 400 CPU.

## 5.3 Performance Results

If not stated otherwise, the reported results are under the following conditions: (i) Averaged across the test cases where the sensor has been removed and remounted, as well as the wrist position is different from one used during training. (ii) Uses the *model with domain adaptation* as described above that requires approximately 90 seconds of training data from each user. The user-independent case is separately evaluated under *model without domain adaptation* (Fig. 11(c)), and *multi-user models* (Fig. 10(a)). The performance of user-dependent models are also shown separately (Fig. 14). (iii) Combines the Encoder-Decoder architecture (including ResNets) in Section 4.1 with semi supervised domain adaptation in Section 4.2 because that is the best performing design with minimal training overhead for different users. The RNN design presented in Section 4.3 is evaluated separately (Fig. 15). (iv) The errors reported are for flex/extension angles as they are prone for more errors with a high range of motion. The errors for abduction/adduction are discussed

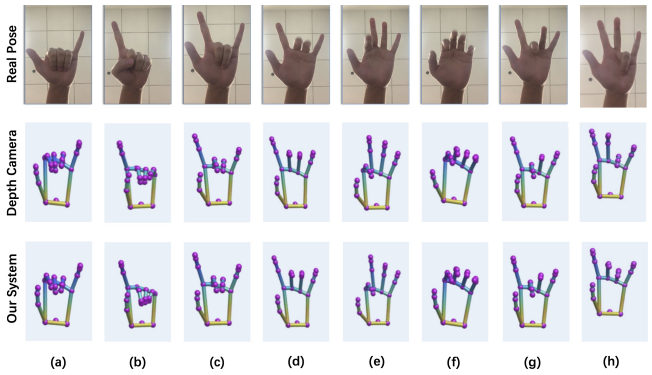


**Figure 9: Comparison of pose tracking results between depth camera (ground truth) and *NeuroPose*.**

separately (Fig. 12(b)). (v) The error bars denote the $10^{th}$ percentile and the $90^{th}$ percentile errors.

**Qualitative Results:** A short demo is provided in this url[15]. Fig. 9 shows qualitative results from *NeuroPose*. The predicted hand pose matches closely with reality for a number of example applications including holding virtual objects, ASL signs, pointing gestures etc. Figs. 9(a) to (c) include static positions, whereas Figs. 9(d) to (g) capture the pose while in motion. Fig. 9(h) is an example of an error case. Our inspection of error cases suggests that in most cases, *NeuroPose* is following the trend in the actual hand pose, albeit with a small delay. This delay introduces errors. Another observation is that the ground truth's (leap depth sensor) detected range of motion for thumb is slightly limited. Extreme thumb motion between Figs 9(a) and (b) causes only a small deviation of the thumb in the leap sensor results. Nevertheless, *NeuroPose*'s prediction of thumb angles match closely with the leap sensor (ground truth).

**Accuracy over Users:** Fig.10(a) shows the breakup of accuracy across users over all joint angles. Although the direct use of a model trained from 11 users (multi-user model) and tested on a new user (without domain adaptation) performs reasonably well with a median error of 9.38° degrees, the $90\% - ile$ errors can be huge.

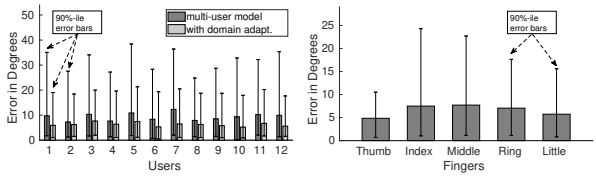


**Figure 10: Performance results (a) Domain adaptation significantly reduces errors over users (b) Accuracy is consistent across fingers.**

On the other hand, semi-supervised domain adaptation techniques not only decreases the median error to 6.24° but also cuts down 90%-ile tail error bars dramatically. The accuracy is robust with diversity in users, their body mass indices, gender etc.

**Robustness to Natural Variations in Sensor Position and Orientation:** We evaluate robustness to natural variations in sensor position by removing the sensor and remounting. Fig. 11(a) shows the accuracy when the sensor position was changed 6 times by removing and remounting (these are the last six sessions of data collection phase). Evidently, the accuracy is consistent across all

positions. In addition, we followed up with all the users over 4 more days to evaluate the robustness over time, temperature, humidity etc. Fig. 11(b) shows the accuracy when the sensor position change happens across multiple days (with a random wrist position). The model that was initially trained continues to provide consistent accuracy over time thus enhancing the usability of *NeuroPose*. We hypothesize that the robustness comes due to three reasons (i) With a snugly fit sensor, its position and orientation changes only by a few mm. The "channel number four" among the 8 EMG channels is clearly marked on the sensor making it easier for the user to maintain the same orientation across multiple sessions of wearing. (ii) Based on the muscle structure map in Fig. 3 which extends from elbow to wrist, the relative positions of the target muscles and the sensor changes only slightly. (iii) The Myo sensor warms up the muscles to ensure good contact with electrodes [12], we believe this helps in robustness for temperature change over days

**Robustness to Wrist Position and Mobility:** Fig. 11(c) shows how the accuracy is consistent despite changes in wrist positions. *NeuroPose* can track finger motions accurately even when the wrist is moving. We hypothesize that regardless of the state of the wrist, ML algorithms always track the muscles responsible for finger motion. The muscles activated for finger motions is independent of the state of the wrist.

**Accuracy over Fingers:** Fig.10(b) provides a breakup of joint angle accuracy over various fingers. For each finger, the accuracy is computed over $\theta_{mcp,f/e}, \theta_{pip}, \theta_{dip}$ angles. For the thumb, the accuracy is computed over $\theta_{mcp,f/e}, \theta_{tm,f/e}, \theta_{ip}$. Overall, the results suggest that *NeuroPose* can track all of the fingers with reasonable accuracy. Although the median error of the index finger is similar to other fingers, one reason why the 90%-ile error is higher could be because the *Extensor Indicis* muscle responsible for index finger motion is a non-surface muscle. Nevertheless, we believe the tracking results of the index finger is promising.

**Accuracy over Flex/Extension Joint Angles:** Fig.12(a) shows the accuracy breakup between the three flex angles – $\theta_{mcp,f/e}, \theta_{pip}$, and $\theta_{dip}$. Evidently, *NeuroPose* maintains similar accuracy for all joint angles. Fig.12(b) depicts that the error in abduction/adduction is smaller than flex/extension angles. This is because the range of motion is very limited in abduction/adduction angles.

**Intrusiveness and Accuracy Trade-offs:** Fig.12(c) illustrates the accuracy over number of EMG channels. As expected, the best results are achieved with all 8 channels. However, the error when only using 4 or even 2 channels (shown in Fig. 13) offers a reasonable trade-off between accuracy/intrusiveness. Evidently, the median accuracy with 4 and 2 channels is comparable to the case with 8 channels, even though the tail errors are higher. This suggests the promise in further decreasing the intrusiveness of the system.

**Training Overhead:** Fig.14(a) shows the accuracy as a function of amount of training data. Evidently, with domain adpatation strategies proposed in *NeuroPose*, even a small fraction (1% - 5% or $9-45$ seconds) of training data is sufficient to generate a model that is as accurate as a model that uses 90% (or 13.5 minutes) of training data without domain adaptation. This demonstrates the ability in *NeuroPose* to quickly generate a model for a new user with an order of magnitude lesser training overhead than training from scratch.

**User Dependent Training:** Although *NeuroPose* performs semi-supervised domain adaptation to generate a model for a new user without extensive training, we evaluate the performance when extensive training is performed for each user to generate her own model. Fig.14(b) summarizes the result. User dependent training can improve the median error by $1.52^{\circ}$, the domain adaptation techniques adopted by *NeuroPose* is close to this performance.

**Accuracy Breakup by Techniques, Comparison to Prior Work:**

Fig.15 shows the CDF of error comparisons over various techniques and prior work. Prior work-1[71] includes an LSTM architecture augmented with a Gaussian process for modeling the error distribution and performs hand pose tracking over a specific set of seven discrete gestures. Prior work-2 [76] uses a RNN architecture with a Simple Recurrent Unit (SRU) and extends [71] with experiments over six specific wrist angles.

Although the algorithms are trained and tested over discrete gestures in the original works, our implementation of these algorithms over arbitrary finger motion gives a median error of 18.95, 14.18 respectively, with a long tail reaching upto 57.31, 54.49 in the 90%-ile respectively. On the other hand, our LSTM architecture that imposes temporal smoothness constraints across multiple handposes brings down the median accuracy to 10.66, and the 90%-ile accuracy to 35.45. The basic Encoder-Decoder architecture performs slightly better with a median accuracy of 14.40 and a 90%-ile accuracy of 32.52. Finally, *NeuroPose* which exploits deeper features by combining ResNets with Encoder-Decoder architecture outperforms the other techniques dramatically both in the median case and in the tail. The median accuracy is 6.24 and a 90%-ile accuracy is 18.33.

**Latency Profiling:** The encoder-resnet-decoder model takes 5 second sequence of EMG data as input. The inference latency of processing each 5s of data using TensorflowLite is roughly 0.101 second. At each instant, by processing the previous 5s of data as input, the model can provide an output in 0.101 seconds, thus ensuring real-time performance. This will incur a cost of redundant processing to provide real-time performance – we will discuss the tradeoffs (Fig. 16(b)(c)). Furthermore, Fig. 16 (a) depicts the average per-sample processing latency of different techniques – LSTM, encoder-decoder and encoder-resent-decoder (NeuroPose) – for relative comparison. The LSTM has a higher processing latency due to the sequential nature of the model with strong dependencies on previous hidden states. In contrast, the encoder-decoder models can exploit parallelism over the entire 5s segment of data.

**Power Consumption Analysis:** The MYO sensor consumes $40mW$ of power [13], which lasts a day of constant usage. For profiling the energy of the TensorflowLite model, we use Batterystats and Battery Historian[14] tools. We compare the difference in power between two states (i) The device is idle with screen on. (ii) The device is making inferences using TensorflowLite model. The idle display-screen on discharge rate 4.97% per hour while the discharge rates for various models is shown in Fig. 16 (b). The power consumption is very low. Since the encoder-resnet-decoder processes data in chunks of 5s, it will incur a delay of atleast 5s if we process the data only once in 5s. Towards making it real-time, we make a modification where at any given instant of time, previous 5s segment of data is input to the network to obtain instantaneous real-time results. This provides real-time tracking at the expense of power.

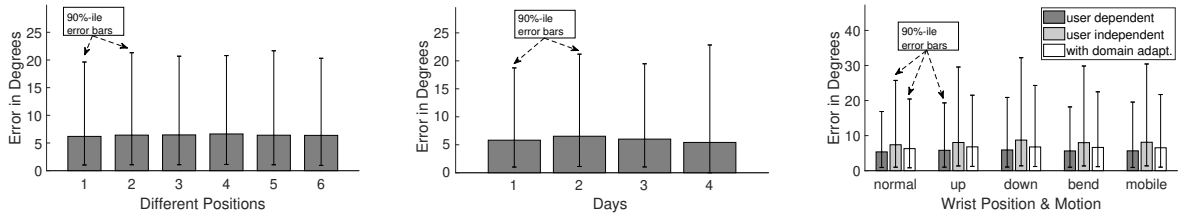


**Figure 11: Robustness to positions (a) change in sensor position within a day (b) across days (c) change in wrist positions**

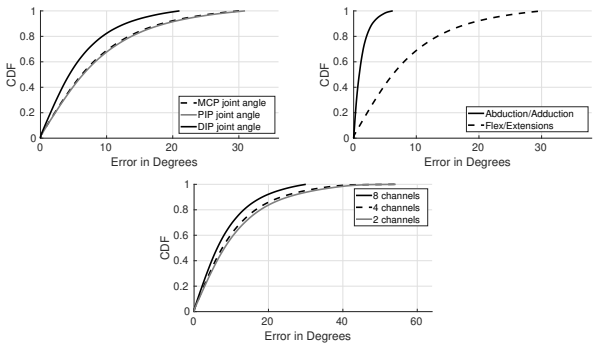


**Figure 12: (a) Accuracy over MCP, PIP, and DIP joints (b) Accuracy over abduction/adductions and flex/extensions (c) Accuracy vs intrusiveness (number of EMG channels)**

## 6 RELATED WORK

*Vision:* Finger motion can be captured by depth cameras like kinect[8] and leap [7] sensors. However, advances in machine learning, availability of large training datasets have enabled precise tracking of finger motion even from monocular videos that do not contain depth information [23, 59]. While such works are truly transformative, we believe wearable based solutions have benefits over vision based approaches which are susceptible to occlusions, lighting, and resolution. In addition, wearable devices offer ubiquitous solution with continuous tracking without the need of an externally mounted camera. Most recently, FingerTrak [41] has innovatively designed wearable thermal cameras to track 3D finger motion. However, tracking may not be robust under changes in background temperature as well as motion of wrist (due to shift in camera positions). In contrast, *NeuroPose*'s EMG sensing is robust to background conditions and wrist motion.

*Sensor Gloves:* Gloves with embedded sensors such as IMU, flex sensors, and capacitative sensors have been used for finger pose tracking in a applications like sign language translation, gaming, HCI etc [19]. Work in [35] tracks hand pose using an array of 44 stretch sensors. Works [26, 51] extracts hand pose using gloves embedded with 17 IMU. Flex sensors have been used in commercially available products such as CyberGlove [2], ManusVRGlove [5], 5DT Glove [1] etc. However, wearing gloves in hands may hinder dexterous/natural hand movements [73].

*IMU and wrist bands:* IMU and WiFi sensors have been used in a number of localization and human body tracking projects [84, 87, 92]. IMU, WiFi, and Acoustics have also been extensively used for hand gesture recognition. [54, 55, 65, 75, 93]. uWave[53] uses accelerometers for user authentication/interaction with mobile devices. FingerIO [61], FingerPing [89] use acoustics for finger gesture detection. Capband [81] uses capacitative sensing for recognizing 15 hand gestures. In contrast, NeuroPose develops algorithms for generic finger motion tracking. Specifically while prior works can only distinguish multi-finger gestures, *NeuroPose* performs free form 3D finger motion tracking. AuraRing [66], a recent work, tracks the index finger precisely using a magnetic wristband and ring on index finger. In contrast, *NeuroPose* tracks all fingers.

*ElectroMyoGraphy:* EMG based gesture tracking is an active area with decades of research. Prior works perform classification of discrete hand poses[28, 32, 71, 72, 76] or tracking of a predefined sequence of hand poses [71, 76] with a combination of deep learning techniques based on CNN, RNN etc. Works [77] can track joint angles for arbitrary finger motion, but requires a large array of over 50 EMG sensors placed over the entire arm. Work in [64] tracks joint angles using EMG sensors but only for one finger. In contrast to these works, *NeuroPose* tracks continuous finger joint angles for arbitrary finger motions with only sparse EMG sensors.

## 7 DISCUSSION AND FUTURE WORK

**Unsupervised Domain Adaptation:** *NeuroPose* only needs 90s of training samples from a new user to customize a pretrained model to the user. However, we will explore unsupervised domain adaptation to customize a pretrained model without requiring any labelled training data. Adversarial domain adaptation [82] is of interest. Here, an unsupervised game theoretic strategy is used to transform the distribution of the feature representations from the new user into the distribution of the source user on whom the model was trained. If successful, the model trained on the source user is directly useful for performing inferences on a new user. Similarly, other architectures for learning feature transformations to adapt the feature representations from a source user to a new users have been proposed [79] which are relevant for future investigation.

**Prosthetic Devices for Amputees:** While the subjects recruited for this study were able-bodied individuals, we will consider design and evaluation of *NeuroPose* for amputees for future work. In particular, given prior research on *mirrored bilateral training* approach [33, 63, 64], we believe there is promise.

**Tracking Fingers while Holding Objects in Hand:** When holding an object, signals from certain muscles that support strength will interfere with muscles responsible for finger motion. While we believe there are enough applications in augmented reality and prosthetics where a user does hold an object, we will carefully refine *NeuroPose*'s algorithms to minimize the interference from additional muscle signals when a user is holding an object.

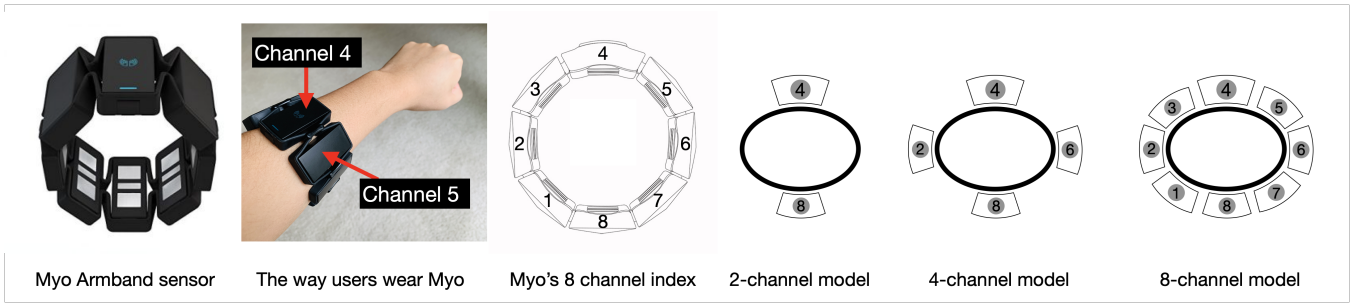


**Figure 13: 2-channel model and 4-channel model compared to 8-channel model for Myo armband sensor**

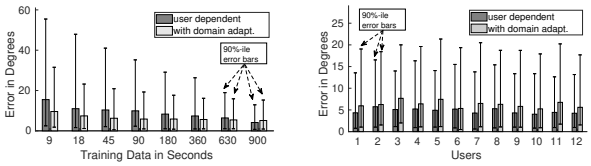


**Figure 14: (a) Domain adaptation minimizes training overhead by an order of magnitude (b) Performance of domain adaptation is close to user dependent training**

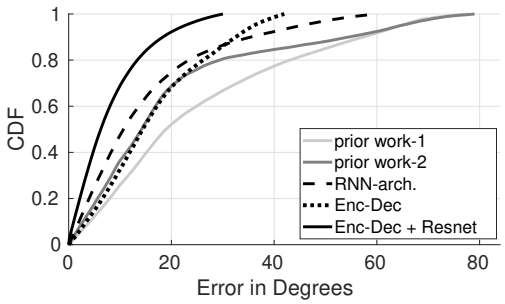


**Figure 15: Encoder-Decoder-ResNet outperforms other techniques**

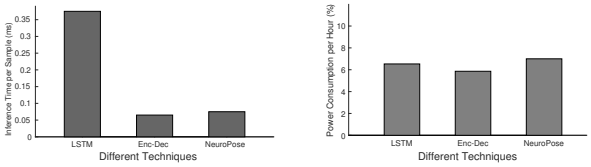


**Figure 16: (a) Latency comparison (b) Power consumption analysis**

## 8 CONCLUSION

This paper shows the feasibility of 3D hand pose tracking using wearable EMG sensors. A number of applications in Augmented Reality, Sports Analytics, Healthcare, and Prosthetics can benefit from fine grained tracking of finger joints. While the sensor data is noisy and involves superimposition of signals from different fingers in complex patterns, we exploit anatomical constraints as well as temporal smoothness in motion patterns to decompose the sensor data into motion pattern of constituent fingers. These constraints are incorporated in an encoder-decoder machine learning model to achieve a high accuracy over diverse joint angles, different type of gestures etc. Semi supervised adaptation strategies show promise in adapting a pretrained model from one user to a new user with minimal training overhead. Finally, the inference runs in realtime on a smartphone platform with a low energy footprint.

## REFERENCES

[1] [n.d.]. 5DT Data Glove Ultra - 5DT. https://5dt.com/5dt-data-glove-ultra/.
[2] [n.d.]. CyberGlove Systems LLC. http://www.cyberglovesystems.com/.
[3] [n.d.]. Facebook Might Have Just Given Us a Peek at Our Wild AR Future. https://www.gizmodo.com.au/2020/09/facebook-might-have-just-given-us-a-peek-at-our-wild-ar-future/.
[4] [n.d.]. Forearm Muscles. https://teachmeanatomy.info/upper-limb/muscles/posterior-forearm/.
[5] [n.d.]. Industry leading VR techology - Manus VR. https://manus-vr.com/.
[6] [n.d.]. Knuckleball Grip, Part 3: Depth of the Baseball. "https://knuckleballnation.com/how-to/knuckleballgrip3/".
[7] [n.d.]. Leap Motion Developer. https://developer.leapmotion.com/.
[8] [n.d.]. Microsoft Kinect2.0. https://developer.microsoft.com/en-us/windows/kinect.
[9] [n.d.]. Muscles Alive: Information and Data Sciences. https://www.bu.edu/ids/research-projects/muscles-alive/.
[10] [n.d.]. Myo official tutorial. https://support.getmyo.com/hc/en-us/articles/203910089-Warm-up-while-wearing-your-Myo-armband.
[11] [n.d.]. Myo resurrected? Facebook acquires CTRL-labs in device gesture-control research push. https://www.zdnet.com/article/facebook-acquires-ctrl-labs-in-machine-mind-control-research-push/.
[12] [n.d.]. Myo warmup. https://support.getmyo.com/hc/en-us/articles/203910089-Warm-up-while-wearing-your-Myo-armband.
[13] [n.d.]. powerconsump. https://pdfs.semanticscholar.org/6c0c/af5d51def3730bb746535099252b724ddd31.pdf.
[14] [n.d.]. Profile battery usage with Batterystats and Battery Historian. https://developer.android.com/topic/performance/power/setup-battery-historian.
[15] [n.d.]. A short video demo of our system. https://www.dropbox.com/s/sssl4e219w2c9al/3D_handpose_demo.avi?dl=0.
[16] [n.d.]. Smart skin: Electronics that stick and stretch like a temporary tattoo. https://www.vice.com/en_us/article/nee8qm/this-tattoo-can-monitor-your-heart-rate-and-brain-waves.
[17] Martín Abadi et al. 2016. Tensorflow: A system for large-scale machine learning. In *OSDI*.
[18] Soumyadipta Acharya et al. 2007. Towards a brain-computer interface for dexterous control of a multi-fingered prosthetic hand. In *2007 3rd International IEEE/EMBS Conference on Neural Engineering*.
[19] Mohamed Aktham Ahmed et al. 2018. A review on systems-based sensory gloves for sign language recognition state of the art between 2007 and 2017. *Sensors* 18, 7 (2018), 2208.
[20] Raman Arora et al. 2016. Understanding deep neural networks with rectified linear units. *arXiv preprint arXiv:1611.01491* (2016).
[21] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence* (2017), 2481–2495.
[22] Mario Bertero, Christine De Mol, and Giovanni Alberto Viano. 1980. The stability of inverse problems. In *Inverse scattering problems in optics*. Springer, 161–214.
[23] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. 2018. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *ECCV*.
[24] Woong-Gi Chang et al. 2019. Domain-specific batch normalization for unsupervised domain adaptation. In *IEEE CVPR*.
[25] Fai Chen Chen et al. 2013. Constraint study for a hand exoskeleton: human hand kinematics and dynamics. *Journal of Robotics* 2013 (2013).
[26] James Connolly et al. 2017. IMU sensor-based electronic goniometric glove for clinical finger movement analysis. *IEEE Sensors Journal* (2017).

[27] Francesca Cordella et al. 2012. Patient performance evaluation using Kinect and Monte Carlo-based finger tracking. In *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 1967–1972.

[28] Ashwin De Silva et al. 2020. Real-Time Hand Gesture Recognition Using Temporal Muscle Activation Maps of Multi-Channel sEMG Signals. *arXiv:2002.03159* (2020).

[29] Artem Dementyev and Joseph A Paradiso. 2014. WristFlex: low-power gesture input with wrist-worn pressure sensors. In *ACM UIST*.

[30] Jia Deng et al. 2009. Imagenet: A large-scale hierarchical image database. In *IEEE CVPR*.

[31] Jacob Devlin et al. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[32] Yu Du et al. 2017. Semi-Supervised Learning for Surface EMG-based Gesture Recognition.. In *IJCAI*.

[33] Jacob A George et al. 2020. Bilaterally mirrored movements improve the accuracy and precision of training data for supervised learning of neural or myoelectric prosthetic control. *arXiv preprint* (2020).

[34] Shawn N Gieser et al. 2017. Evaluation of a low cost emg sensor as a modality for use in virtual reality applications. In *International Conference on Virtual, Augmented and Mixed Reality*. Springer.

[35] Oliver Glauser et al. 2019. Interactive hand pose estimation using a stretch-sensing soft glove. *ACM Transactions on Graphics (TOG)* (2019).

[36] Google. 2019. Deploy machine learning models on mobile and IoT devices. "https://www.tensorflow.org/lite".

[37] google ar web 2021. Augmented reality for the web. https://developers.google.com/web/updates/2018/06/ar-for-the-web.

[38] Xiaobing Han et al. 2017. Pre-trained alexnet architecture with pyramid pooling and supervision for high spatial resolution remote sensing image scene classification. *Remote Sensing* (2017).

[39] SM Hasan and Cristian A Linte. 2019. U-NetPlus: a modified encoder-decoder U-Net architecture for semantic and instance segmentation of surgical instrument. *arXiv preprint arXiv:1902.08994* (2019).

[40] Kaiming He et al. 2016. Deep residual learning for image recognition. In *IEEE CVPR*.

[41] Fang Hu et al. 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* (2020).

[42] Sergey Ioffe et al. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).

[43] Herbert Jaeger. 2002. *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the" echo state network" approach*. Vol. 5. GMD-Forschungszentrum Informationstechnik Bonn.

[44] David Kim et al. 2012. Digits: freehand 3D interactions anywhere using a wrist-worn gloveless sensor. In *ACM UIST*.

[45] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[46] Toshiki Koshio et al. 2012. Identification of surface and deep layer muscles activity by surface EMG. In *2012 Proceedings of SICE Annual Conference*. IEEE.

[47] Alex Krizhevsky et al. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

[48] Steve Lawrence et al. 1997. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks* (1997).

[49] Rita Layona et al. 2018. Web based augmented reality for human body anatomy learning. *Procedia Computer Science* (2018).

[50] Shan Sung Liew et al. 2016. Bounded activation functions for training stability of deep neural networks on visual pattern recognition problems. *Neurocomputing* (2016).

[51] Bor-Shing Lin et al. 2018. Design of an inertial-sensor-based data glove for hand function evaluation. *Sensors* (2018).

[52] John Lin, Ying Wu, and Thomas S Huang. 2000. Modeling the constraints of human hand motion. In *Proceedings workshop on human motion*. IEEE, 121–126.

[53] Jiayang Liu et al. 2009. uWave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing* (2009).

[54] Yilin Liu et al. 2020. Application Informed Motion Signal Processing for Finger Motion Tracking Using Wearable Sensors. In *IEEE ICASSP*.

[55] Yilin Liu et al. 2020. Finger Gesture Tracking for Interactive Applications: A Pilot Study with Sign Languages. *ACM IMWUT* (2020).

[56] Massimiliano Mancini et al. 2018. Boosting domain adaptation by discovering latent domains. In *IEEE CVPR*.

[57] Jess McIntosh et al. 2017. Echoflex: Hand gesture recognition using ultrasound imaging. In *2017 CHI Conference on Human Factors in Computing Systems*.

[58] Tomáš Mikolov et al. 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

[59] Franziska Mueller et al. 2018. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *IEEE CVPR*.

[60] Ander Ramos Murguialday et al. 2007. Brain-computer interface for a prosthetic hand using local machine control and haptic feedback. In *2007 IEEE 10th International Conference on Rehabilitation Robotics*. IEEE.

[61] Rajalakshmi Nandakumar et al. 2016. Fingerio: Using active sonar for fine-grained finger tracking. In *ACM CHI*.

[62] Wajahat Nawaz et al. 2018. Classification of breast cancer histology images using alexnet. In *International conference image analysis and recognition*. Springer.

[63] Johnny LG Nielsen et al. 2010. Simultaneous and proportional force estimation for multifunction myoelectric prostheses using mirrored bilateral training. *IEEE Transactions on Biomedical Engineering* (2010).

[64] Lizhi Pan et al. 2014. Continuous estimation of finger joint angles under different static wrist motions from sEMG signals. *Biomedical Signal Processing and Control* (2014).

[65] Abhinav Parate et al. 2014. Risq: Recognizing smoking gestures with inertial sensors on a wristband. In *ACM MobiSys*.

[66] Farshid Salemi Parizi, Eric Whitmire, and Shwetak Patel. 2019. AuraRing: Precise Electromagnetic Finger Tracking. *ACM IMWUT* (2019).

[67] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2012. Understanding the exploding gradient problem. *CoRR, abs/1211.5063* 2 (2012).

[68] Esteve Peña Pitarch. 2008. *Virtual human hand: Grasping strategy and simulation*. Universitat Politècnica de Catalunya.

[69] Panagiotis Polygerinos et al. 2015. EMG controlled soft robotic glove for assistance during activities of daily living. In *2015 IEEE international conference on rehabilitation robotics (ICORR)*. IEEE.

[70] Chen Qu et al. 2019. BERT with history answer embedding for conversational question answering. In *ACM SIGIR Conference on Research and Development in Information Retrieval*.

[71] Fernando Quivira et al. 2018. Translating sEMG signals to continuous hand poses using recurrent neural networks. In *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. IEEE.

[72] Sumit Raurale et al. 2018. Emg acquisition and hand pose classification for bionic hands from randomly-placed sensors. In *IEEE ICASSP*.

[73] Alba Roda-Sales et al. 2020. Effect on manual skills of wearing instrumented gloves during manipulation. *Journal of biomechanics* (2020).

[74] Stefano Scheggi et al. 2015. Touch the virtual reality: using the leap motion controller for hand tracking and wearable tactile devices for immersive haptic rendering. In *ACM SIGGRAPH Posters*.

[75] Michael Sherman et al. 2014. User-generated free-form gestures for authentication: Security and memorability. In *ACM MobiSys*.

[76] Ivan Sosin et al. 2018. Continuous gesture recognition from sEMG sensor data with recurrent neural networks and adversarial domain adaptation. In *International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE.

[77] Sorawit Stapornchaisit, Yeongdae Kim, Atsushi Takagi, Natsue Yoshimura, and Yasuharu Koike. 2019. Finger Angle estimation from Array EMG system using linear regression model with Independent Component Analysis. *Frontiers in Neurorobotics* 13 (2019).

[78] Dan Stashuk. 2001. EMG signal decomposition: how can it be accomplished and used? *Journal of Electromyography and Kinesiology* 11, 3 (2001), 151–173.

[79] Baochen Sun and Kate Saenko. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*. Springer.

[80] Evan A Susanto et al. 2015. Efficacy of robot-assisted fingers training in chronic stroke survivors: a pilot randomized-controlled trial. *Journal of neuroengineering and rehabilitation* (2015).

[81] Hoang Truong et al. 2018. CapBand: Battery-free Successive Capacitance Sensing Wristband for Hand Gesture Recognition. In *ACM SenSys*.

[82] Eric Tzeng et al. 2017. Adversarial discriminative domain adaptation. In *CVPR*.

[83] Stefan Wager, Sida Wang, and Percy S Liang. 2013. Dropout training as adaptive regularization. In *Advances in neural information processing systems*. 351–359.

[84] Junjue Wang et al. 2014. Ubiquitous keyboard for mobile devices: harnessing multipath fading for fine-grained keystroke localization. In *ACM MobiCom*.

[85] Web XR API 2021. Web XR Device API. https://www.w3.org/TR/webxr/.

[86] Jørgen Winkel et al. 1991. Significance of skin temperature changes in surface electromyography. *European journal of applied physiology and occupational physiology* (1991).

[87] Jie Xiong and Kyle Jamieson. 2013. Arraytrack: A fine-grained indoor location system. In *USENIX NSDI*.

[88] S Xu et al. 2019. The Effectiveness of Virtual Reality in Safety Training: Measurement of Emotional Arousal with Electromyography. In *ISARC*.

[89] Cheng Zhang et al. 2018. FingerPing: Recognizing Fine-grained Hand Poses using Active Acoustic On-body Sensing. In *ACM CHI*.

[90] Haoyu Zhang, Jianjun Xu, and Ji Wang. 2019. Pretraining-based natural language generation for text summarization. *arXiv preprint arXiv:1902.09243* (2019).

[91] Yang Zhang et al. 2015. Tomo: Wearable, low-cost electrical impedance tomography for hand gesture recognition. In *ACM UIST*.

[92] Mingmin Zhao et al. 2019. Through-wall human mesh recovery using radio signals. In *IEEE CVPR*.

[93] Pengfei Zhou et al. 2014. Use it free: Instantly knowing your phone attitude. In *ACM MobiCom*.

[94] Zongwei Zhou et al. 2017. Fine-tuning convolutional neural networks for biomedical image analysis: actively and incrementally. In *IEEE CVPR*.