A Practical System for 3D Hand Pose Tracking using EMG Wearables with Applications to Prosthetics and User Interfaces

Yilin Liu, Shijia Zhang, Mahanth Gowda

***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. In addition, this paper validates the feasibility of *mirrored bilateral training* approach with applications in prosthetic devices. Finally, *NeuroPose* is comprehensively evaluated on both low-end and recent smartphones with a processing latency of $0.019s$ and low energy overhead.**

***Index Terms*—Human centered computing, electromyography, accessibility, representation learning.**

I. Introduction

3D finger pose tracking enables a number of exciting applications in sports analytics [6], healthcare and rehabilitation [103], sign languages [20], augmented reality (AR), virtual reality (VR), haptics [95] 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 [33]. 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 [77], [18].

Web-based augmented/virtual reality applications are becoming popular [45], [62] leading to standardizations of WebXR APIs [112]. 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 [76], [26] 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 [2], [5], [1] with IMU, flex, and capacitative sensors have been popularly used for finger motion tracking, it is shown that gloves hinder with dexterous hand movement [92]. As alternatives to putting sensors on fingers, sensing at wrist with surface acoustic [117], capacitative [104], bioimpedance [119], ultrasonography [73], wrist pressure[37] etc., has been explored, but the sensing is only limited to tens of gestures. Beyond discrete gestures, infrared [56] and thermal cameras [49] mounted on wrist have been explored for continuous 3D pose tracking, but has limitations on hand motion (details in Section. II). 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 [42], [116]. (v) Additionally, a unique motivation for ElectroMyoGraphy (EMG) based

Yilin Liu, Penn State (yzl470@psu.edu)
Shijia Zhang, Penn State (scarlettzhang27@psu.edu)
Mahanth Gowda, Penn State (mahanth.gowda@psu.edu)

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3223600

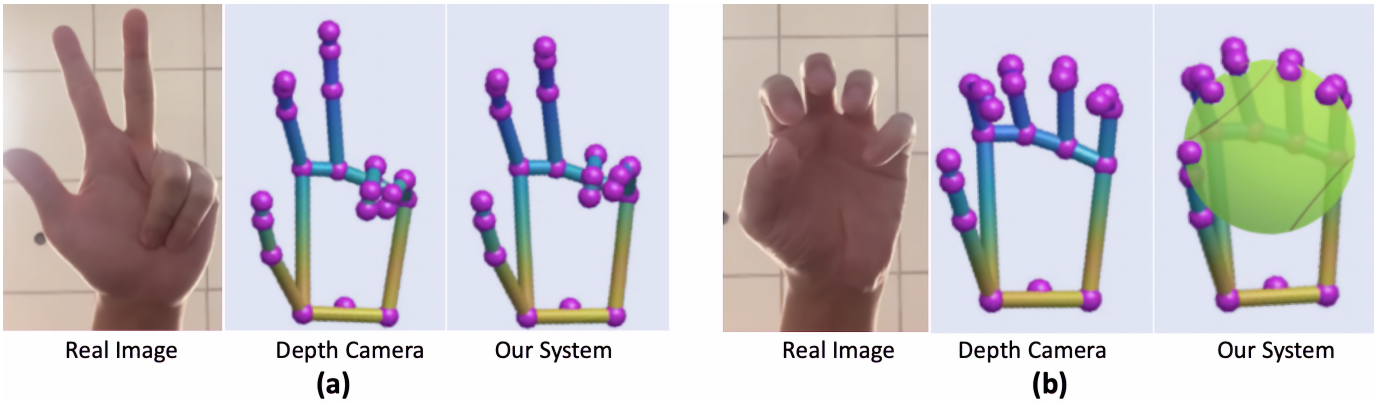


Fig. 1: A comparison between a real image, a depth camera, and *NeuroPose*. Tracking of fine grained hand poses can enable applications like: (a) Word recognition in sign languages (b) Augmented reality by enhancing the tracking output. A short demo is here [15].

tracking over vision and other wearable systems (such as IMU) is that EMG signals from amputees can be used for controlling prosthetic limbs with potential to significantly improve their accessibility needs, the feasibility of which is shown in prior work for index finger motions, wrist motions, gestures etc [81], [80]. Even though the fingers might be missing, studies have shown that the subjects with amputations are capable of generating neuromuscular potentials that is responsible for a particular pattern of finger motion [41], [35], [80], [81]. However, generating training data can be a challenge for such cases. Our preliminary result in Section VI discusses a *mirrored bilateral training* approach for generating training data for amputees for 3D finger motion tracking. The results are promising with applications in development of prosthetic devices with finer control.

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 [11], [3]) 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 [91], [36], [40], [90], [98], or tracking hand poses over a set of discrete gesture related motions [90], [98]. 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. III). 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)[75], Encoder-Decoder[22], ResNets[48] 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 $0.019s$, 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. Performance comparison across both low-end and more recent smartphone platforms demonstrates a competitive performance across a wide spectrum of devices. Further-

more, we show the applicability of *NeuroPose* in real world use cases such as finger-spelling classification in American Sign Language (ASL). 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. In contrast to discrete gesture classification for a particular application, such a generic tracking can provide continuous joint angles which can be used for any application like sports analytics, AR/VR, sign language recognition, etc.

(2) Fusion of anatomical constraints with sensor data into machine learning algorithms for higher accuracy. While the search space of 3D finger motion is very large, we also note that different fingers and their joint motion is not completely independent and exhibit relationships with each other. NeuroPose *incorporates this dependency in the machine learning models for narrowing down the search space and enable accurate tracking.*

(3) Implementation across diverse smartphone platforms and extensive evaluation over diverse users. The accuracy is consistent across diverse users and provides a low latency performance with less power consumption on modern sport phones.

(4) Evaluation of a mirrored bilateral training *[80] scheme with a potential future application for developing prosthetics for amputees with missing fingers.* Through a combination of machine learning model design for noise reduction and real world experiments, we validate that the training data captured from one hand can be applied for inferences on the other hand. We believe this facilitates an easy alternative for collection of training data for prosthetic devices where labelling might be challenging because of missing fingers.

The rest of the paper is organized as follows. We begin with a brief overview of the human hand and it's anatomy. The muscles of interest and how they are captured by the EMG sensor is also explained. We also discuss the mirrored bilateral motion in the context of amputees and how it is incorporated in the paper. The hardware platform of the EMG sensor is discussed next. After this, we discuss the encoder decoder based machine learning model that converts the EMG data into 3-D hand poses. Representation learning and contrastive loss function are introduced to handle noise during the mirror bilateral training for amputees. Finally, we evaluate the paper based on a systematic user study to validate the feasibility of 3-D hand motion tracking as well as mirrored bilateral training for amputees.

II. Related Work

Vision: Depth cameras including kinect[8] and leap motion [7] sensors have revolutionized the gaming industry by gesture interfaces. Use of depth camera is one way to capture finger motion. However, advances in machine learning, availability of large training datasets as well as techniques for creation of synthetic datasets have enabled precise tracking of finger motion even from monocular videos that do not contain depth information [76], [26], [52]. DeepFisheye [84] uses fish eye cameras combined with deep learning to track finger tips with high precision. While such works are truly transformative in nature, 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. Digits [56] uses wrist mounted infrared cameras for 3D finger pose tracking. Similarly, DorsalNet [114] uses wrist mounted visual cameras for 3D finger motion tracking. The dorsal hand region including the motion of bones, muscles, and tendons are analyzed with a two stream convolutional neural network for precise 3D motion tracking. However, the camera needs to sit high enough on the wrist or even reach palm to capture full range of finger motion. Most recently, FingerTrak [49] has innovatively designed wearable thermal cameras that can track 3D finger motion. However, the authors bring-up the following aspects in their paper: (i) If the background temperature is similar or higher (sun, heater etc.), the tracking may not be robust. (ii) In the current prototype, the arm position of the user has to be on the table without which the cameras can shift and affect the tracking results. In contrast, our work explores the use of EMG sensors, which is robust to background conditions as well as changes in wrist position, with a unique potential for applicability to developing prosthetic devices for amputees.

Finger Motion Tracking by Radio Frequency Reflections: Prior works have explored WiFi signals to track motion of hand and classify discrete gestures by using a combination of wireless channel state information (CSI), and doppler shift measurements [63], [74], [96]. SignFi [70] is an innovative work that uses wireless channel measurements from WiFi APs for sign language recognition. ExASL [93] tracks point clouds computed from range-doppler spectrum and angle of arrival spectrum of mmWave reflections from the hand. This is used to classify upto 23 discrete hand motion gestures used in ASL. Google Soli [111] exploits reflections from mmWave signals in combination with deep convolutional and recurrent neural networks to track 11 finger motion gestures. In contrast to discrete gesture classification in the above works, *NeuroPose* performs continuous 3D finger motion tracking. In additon, while the above approaches are limited by range of coverage of mmWave and WiFi signals, *NeuroPose* offers a more ubiquitous tracking.

Sensor Gloves: Gloves with embedded sensors such as IMU, flex sensors, and capacitative sensors have been used for finger pose tracking in a number of applications including sign language translation, gaming, user interface etc [20]. Work in [43] tracks 3D hand pose using an array of 44 stretch sensors. Works [32], [65] extract hand pose using gloves embedded with 17 IMU sensors. Flex sensors have been successfully used in commercially available products such as CyberGlove [2], ManusVRGlove [5],

5DT Glove [1] etc. However, wearing gloves in hands may hinder dexterous and natural hand movements. This precludes the user from performing activities that require fine precision as studied in recent works[92].

IMU, Wrist Bands, and Wearable Sensing: IMU and WiFi sensors have been used in a number of localization and human body tracking projects [110], [31], [19], [115], [120]. IMU, WiFi, and Acoustic signals have also been extensively used for hand gesture recognition to enable a number of applications [121], [105], [82], [97]. uWave[67] uses accelerometers for user authentication and interaction with a mobile device. FingerIO [78], FingerPing [117] use acoustic signals for finger gesture detection. WiFi based hand/finger gesture detection has been explored [63], [74], [96] that use wireless channel and doppler shift measurements for hand gesture recognition. WiSee [88] uses Doppler shifts from WiFi reflections for applications such as controlling devices in a smart-home, gaming etc. SignFi [70] is an innovative work that uses wireless channel measurements from WiFi APs for ASL recognition. Capband [104] uses capacitative sensing for recognizing 15 hand gestures. In contrast, NeuroPose develops algorithms for generic finger motion tracking. ElectroRing [55] attaches elctrodes on the index finger and combines them with IMU sensors for detecting six different pinch-like finger gestures. ThumbTrak [102] detects 12 finger gestures by placing 9 proximity sensors on the thumb and measuring the distance from the thumb to the other fingers and palm. ZeroNet [68] extracts training data from videos to classify 50 different hand gestures. Specifically while prior works can only distinguish multi-finger gestures, *NeuroPose* performs free form 3D finger motion tracking. AuraRing [83], a recent work, tracks the index finger precisely using a magnetic wristband and ring on index finger. In contrast, *NeuroPose* tracks all fingers.

ElectroMyoGraphy: The use of EMG signals for hand pose tracking is an active area with decades of research. Prior works perform classification of discrete hand poses[91], [36], [40], [90], [98] or tracking of a predefined sequence of hand poses [90], [98] using EMG sensors with a combination of deep learning techniques based on CNN, RNN etc. Work in [29] can classify multi finger gesture sequences using a 4 channel EMG sensor. A number of popular features based on spectral power magnitudes, hudgins' time domain features, correlation coefficients etc have been used in conjunction with SVMs, nearest neighbors, and linear discriminant analysis based algorithms to show the feasibility of gesture classification. Work in [23] uses Myo armband similar to the one used in this paper to classify 5 gestures such as fist, wave-in, wave-out, open, and pinch etc. A shallow feed forward neural network with 3 layers has been used to perform this classification. Work in [69] shows that muscle synergy can be exploited to reduce the dimensions of feature vectors in EMG based gesture classification. Evaluated over five hand activities such as open, close, pinch, valgus, and grasp, the recorded EMG data from the forearm have been compressed using non negative matrix factorization to extract synergistic myo-electrical activities. The compressed feature set has shown to demonstrate a higher recognition rate. Work in [94] uses forearm EMG signals to control a robotic arm. A set of 9 gestures are detected to contral a 6 degree of freedom robotic arm. Ensembled bagged trees, SVM, and neural networks have been used to perform the classification. Works [99] can track joint angles for arbitrary finger motion, but requires a large array of more than 50 EMG sensors placed over the entire arm. Work in [81] tracks joint angles using EMG sensors but only for one finger. In contrast to these works, *NeuroPose* performs accurate tracking of continuous finger joint angles for arbitrary finger motions with only sparse EMG sensors.

Mirrored Bilateral Training: Work in [80] estimates the force on contralateral arm using EMG signals measured from the other arm. A multilayer perceptron (MLP) based algorithm has been used to make the association between EMG signals and the associated force in the arm. Based on several experiments with tens of individuals , this paper shows that an accurate estimation of forces in the contralateral limb can be done based on the EMG signal from the other arm, thus showing promise. Similarly, work in [81] shows the feasibility of estimation of flex and extension joint angles of one finger based on EMG data collected from the other hand. A number of features such as zero crossings, mean absolute value, waveform length, slope changes etc has been applied on EMG data. Furthermore, a state space model with parameters estimated from contralateral arm is used to estimate the joint angle of one finger on the other arm. The results show an estimation error under 1 degree thus indicating sufficient promise. Work in [41] compares training via mirrored EMG from contralateral arm with training by mimicking gestures on the same arm with potential amputation. Evaluated over more than 20 gestures, a better performance is achieved by mirroring on the contralateral arm instead of mimicking with the same arm that may have amputation. The main challenge with mimicking is identified as the inability to estimate force involved in motion as well as misalignment over time with between the imitation and the actual gesture. Work in [54] can perform wrist motion classification using *mirrored bilateral training*. Based on the EMG data from the contralateral arm, and employing techniques based on artificial neural networks for pattern classification, upto 70% in accuracy has been shown in terms of classification of 4-6 wrist motion gestures. All of the above works show promise in the technique of *mirrored bilateral training*. In contrast to these works which either track discrete gestures or continuous motion of one finger, *NeuroPose* shows the feasibility of *mirrored bilateral training* for continuous estimation of 21 degrees of freedom involved in 3D hand pose estimation.

III. 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. (iii) *Mirrored bilateral training* scheme for generating training data for amputees with missing fingers.

A. 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

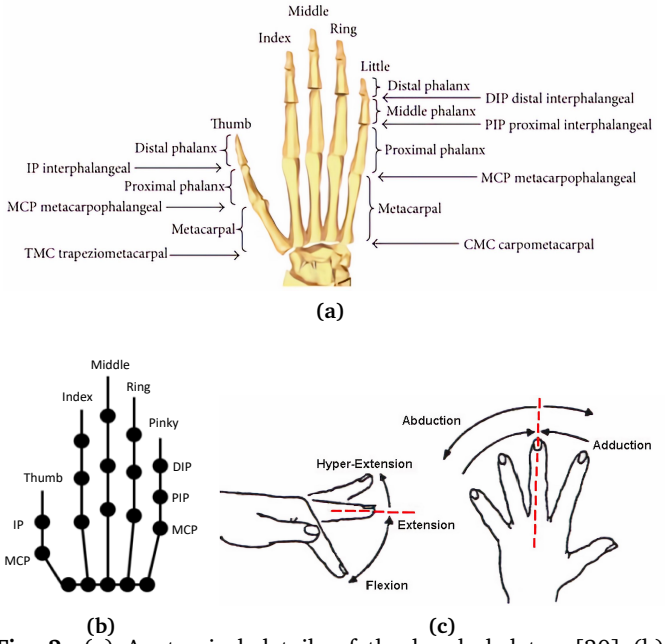


Fig. 2: (a) Anatomical details of the hand skeleton [30] (b) Kinematic structure and joint notations [66] (c) Finger motions include flex/extensions and abduction/adductions [86]

(metacarpophalangeal), PIP (proximal interphalangeal), and DIP (distal interphalangeal) joints. The joint angles at PIP (θ_{pip}) and DIP (θ_{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 (θ_{ip}). The MCP and TM (trapeziometacarpal) joints possesses both flex and abduction/adduction DoF, thus the thumb has five DoF – θ_{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 [66], [30]. 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.

B. 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 [100]. 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

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3223600

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 [59]. 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 V.

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 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 τ_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

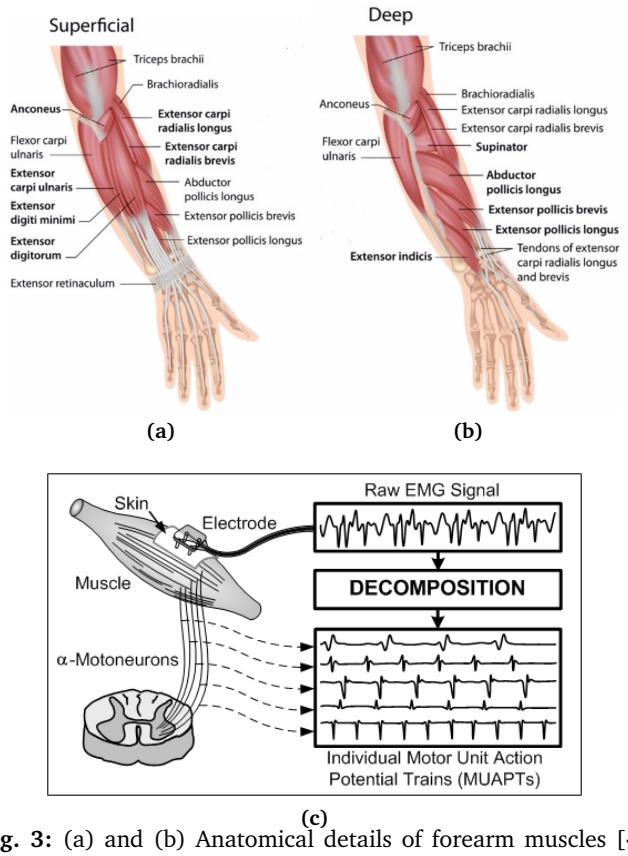


Fig. 3: (a) and (b) Anatomical details of forearm muscles [4] (c) EMG signals from an electrode can be decomposed into constituent motor unit action potential trains (MUAPT) [9]

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 τ_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 jth motor unit fires, δ_{jk} is the kth firing time of the jth 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

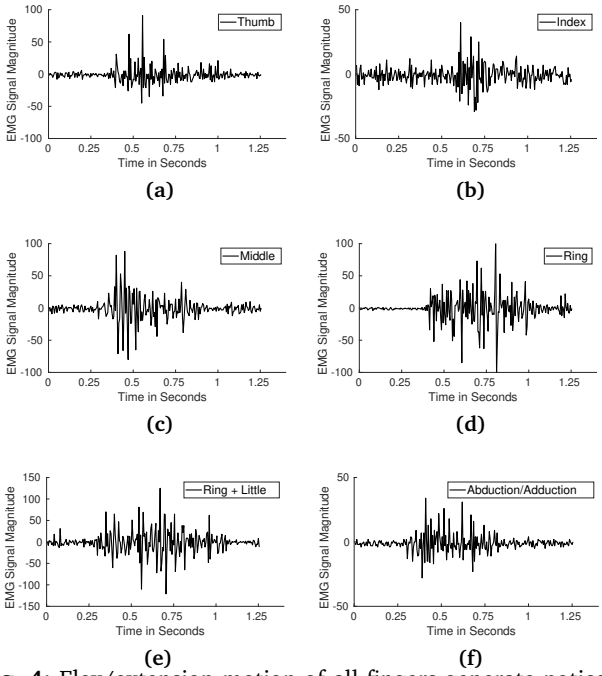


Fig. 4: Flex/extension motion of all fingers generate noticeable "spike" in the EMG data for (a) Thumb (b) Index (c) Middle (d) Ring (e) Little + Ring (the little finger cannot be flexed without jointly moving the ring finger) (f) Abduction/Adduction of all fingers

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 V for signal decomposition.

C. Mirrored Bilateral Motion

An important application of EMG devices is in developing prosthetic devices for amputees with missing fingers. However, because of missing fingers, it is non-trivial to generate training data that maps EMG signal pattern into corresponding 3D joint angles of various fingers. Towards handling this challenge, we explore a *mirrored bilateral training* [80] scheme. In this subsection, we introduce the biological foundations of *mirrored bilateral training* as well as provide high level details on exploiting this opportunity for generating training data for amputees.

A unilateral motion such as a motion with the right hand induces involuntary muscle activation in the contralateral part of the human body such as the left hand. This is called as *mirrored bilateral motion* and the corresponding activity in electromygraphy signals is called as *mirrored electromyography* (MEMG). MEMG has been consistently observed in both healthy and pathological cases over a number of simple and complex motor activities in daily life. This is known to happen because of a motor overflow that causes the involuntary muscular activity due to interhemispheric communication within the brain during motion activities [72]. Such interhemispheric communication leads to bilateral activation of motor relevant brain regions. Although the evolution in humans have gone through an ontogenetic learning process that decouples both hands to be independent of each other, the MEMG activity is said to be a remnant of the basic mirror movement mode of the central nervous system [107]. This facilitates mirrored motions under voluntary setting where both hands can move synchronously in nearly identical fashion. As elaborated later in this section, *NeuroPose* will exploit this property in an application for developing prosthetic devices for amputees.

A transradial amputation is one where a part of the arm is missing below the elbow beyond a certain point along the radial bone. Such an amputation might occur because of a number of reasons including injury, tumor, frostbite, infection etc. A number of prosthetic devices have been proposed for such cases. This includes cosmetic prosthetic devices which do not move but used solely for the purpose of appearance [34]. On the other hand, a body powered prosthesis is attached to the body by a series of wires [50]. Moving the body in different ways will move the prosthetic device for performing different activities. Finally, a myoelectric prosthesis is the most advanced form of prosthetic device. EMG signals from the brain can be used to control the prosthetic hand resulting in an effect that is similar to a real hand [27].

At a high level, transradial amputees will still retain the neuromuscular structure that is responsible for precise finger motion. Even though the fingers might be missing, studies have shown that the subjects with amputations are capable of generating neuromuscular potentials that is responsible for a particular pattern of finger motion [41], [35], [80], [81]. By identifying the appropriate patterns, an external prosthetic device can be attached to produce those actions thus providing an experience that is close to a real hand.

While mapping the EMG signals to finger patterns for able bodied individuals is easy because machine learning models can be trained to map the EMG signals to finger motions, the same is not feasible with amputees. The lack of a finger precludes training data that maps the EMG signals to the motion of that finger. One possibility to handle this challenge is to let the amputee emulate a few predefined finger motion patterns (flicking a finger etc), and record the EMG signals to be used for training [41]. However, the action of the amputee might differ from the predefined finger pattern in temporal alignment, bio-mechanical coupling as well as the intensity of force applied, thus resulting in poor quality of training data. *Mirrored bilateral motion* as described earlier can be exploited as an opportunity to handle this challenge. The neural activation patterns are known to be similar in both hands for performing similar finger motion activities [107]. Therefore, a machine learning model trained with the non-amputee hand (without missing fingers) while inducing bilateral activation can potentially be used for performing inferences on the hand with missing fingers

(amputated hand). Thus, appropriate control signals can be generated for controlling the prosthetic device attached to the amputated hand. *NeuroPose* exploits this opportunity and provides insights into the feasibility of such a *mirrored bilateral training* approaches for prosthetics capable of performing fine grained 3D finger motion instead of discrete gestures.

IV. Platform Description

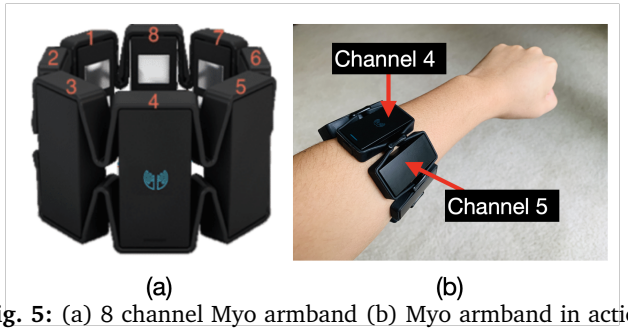


Fig. 5: (a) 8 channel Myo armband (b) Myo armband in action

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 implemented on smartphones (OnePlus 9 Pro, Samsung S20, Sony Xperia Z3) that capture the EMG data and provide 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 VI).

Skin Temperature Calibration: The EMG amplitude may be slightly affected by skin temperature variations [113]. The surface 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 [11], [3] for developing finger tracking technology for its thrust towards AR/VR applications. Other form factors ranging from armbands, tattoos, and arm-gloves have been proposed by both academia and industry with no consensus on what is best [16], [87], [91], [36], [99]. 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 uncertainties 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.

V. Core Technical Modules

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

A. 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,

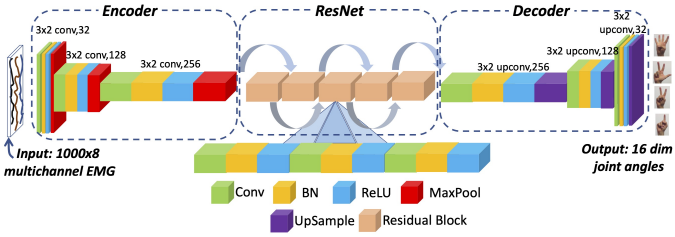


Fig. 6: Encoder Decoder Architecture used in *NeuroPose*:

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 $5s$ 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 (θ_{dip} for four fingers and θ_{ip} for thumb) can be directly computed using Equations 1, 2. Thus, the actual output of the network is only 16 dimensions – $\mathbb{R}^{16}$.

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[61]. 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 [51]. (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 [21]. 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 [108] 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 sub-regions of the initial representation. For example, a max-pool filter of size 2×2 applied to an input of size 100×100, will slide a non-overlapping window of size 2×2 and extracts the maximum element from the input at each overlap resulting in an output of size 50×50.

The first of the CONV-BN-RELU-MAXPOOL layers applies 32 2D-*CONV* filters of size 3×2, 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 (4×2), and (2×2) 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[48] 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. VI, 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 [47]) 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 (5×4),(4×2),(2×2) respectively on temporal and spatial domains thus matching the output shape of 1000×16 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 θ 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 θ_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 θ_{dip} or θ_{ip} because we compute them directly from anatomical constraints: Equations 1, 2.

Finger motion range constraints: As described in Section III, 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 θ_{dip}). We then apply the bounded ReLU activation (bReLU) function [64] 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.

B. 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 [60] pretrained on ImageNet database [38] was fine-tuned for classifying images in medical domain[122], remote-sensing [46] and breast-cancer [79]. Similarly, a pre-trained BERT language model [39] was fine-tuned for tasks such as text-summarizing [118], question answering [89] 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. 17(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* [51] 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 [28], [71].

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* [51], 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 VI, this results in reduction of training overhead on the *target* user by an order of magnitude.

C. RNN Architecture

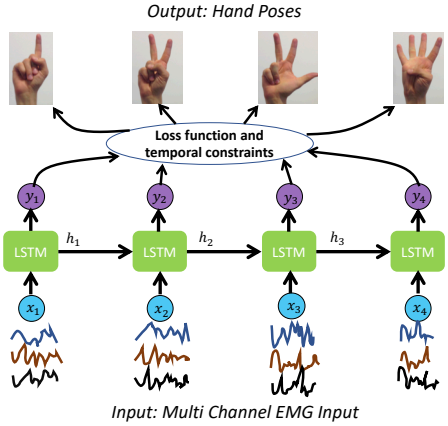


Fig. 7: RNN alternative explored in this paper.

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 [85] 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 [53]) in training with a truncation of 64 time units.

D. Representation Learning for Mirrored Bilateral Training

As discussed in Section III-C, the neuromuscular interactions are such that an amputated hand still preserves the muscular activation and exhibits strong similarities to the muscular activity in the non amputated hand. Therefore, the training data, labels, and models developed from the non amputated hand can be used for performing inferences on the amputated hand. To handle any residual differences in neuromuscular activity between amputated and non amputated hand, we develop an architecture based on representation learning to further improve the accuracy. Fig. 8 depicts the high-level architecture of the representation learning framework used in *NeuroPose*. The raw sensor input x^i is first transformed into two variants (x_1^i and x_2^i) based on data augmentation techniques. The transformations add perturbations to the data while still retaining the overall pattern. x_1^i and x_2^i are then fed as input to the neural network that extracts representations h_1^i and h_2^i as shown in the *self-supervised stage* of the figure. Finally, the representations are projected into a latent space before comparing them using a *contrastive loss function* that attempts to maximize the similarity between y_1^i and y_2^i while minimizing the similarity between y^k and y^j, where $k \neq j$. Since such a network tries to enforce similarity among representations even though the inputs have been perturbed by data augmentation techniques, it is known to learn efficient representations. Finally, the representations thus learned are fine tuned with labeled data for predicting the 3D finger motion joint angles. Evaluated in Section VI using the representations enhances robustness of adaptability of a model trained from a non amputated hand for inference on the amputated hand.

Data Augmentation: Towards learning self-supervised representations, we employ data augmentation techniques to our ML model. This helps avoid overfitting as well as teaches the algorithm to look for stable features that measure similarity. The architecture in Fig. 8 needs two data augmentation techniques at a given instant of time. We take a combination of two from the following three techniques: (i) Temporal masking: We mask parts of the input data along time axis so as to help our model in capturing the temporal dependencies in the sensor data. Such strategy is popular in natural language processing such as BERT [39] where words are masked in a sentence to force the language model to predict these words, thus facilitating learning of efficient representations of sentences. Inspired by BERT, we add temporal masking along time as an data augmentation technique. (ii) Noise Addition: We add Gaussian noise to the input data to create augmented versions of the sensor data. Such a process of adding randomness and enforcing similarity between differently augmented copies of the input will teach the model to look for stable features and also help it to avoid overfitting issues. We believe this also help facilitate the mirrored bilateral training process where the training and test data come from different hands, with potential noise between them. (iii) DTW based data augmentation: The speed of hand motion is one of the metrics that can vary across time and users. Various parts of the finger motion might be performed at a faster or slower pace. Towards making the ML models robust to such variations, we augment the training data by stretching and compressing different parts of the data using DTW [24] based algorithm with different factors.

Contrastive Loss Function: The encoded representations h are passed through a projection head as shown in Fig. 8, resulting in an output $y = p(h)$ where p represents the action of the projection head. We apply the contrastive loss function on y that maximizes the similarity between two differently augmented copies of the same input. The contrastive loss function is applied on y whereas we use representations h in the later phases for prediction of 3D finger motion. The reasons for such a design choice are as follows. (i) Since the contrastive loss function's main goal is to maximize the similarity between differently augmented versions of the same input, it might lose some information during the process. (ii) On the other hand, the encoded representation h is one level before the projection head, and it offers the best trade-off between capturing high-level robust representations without losing much information.

The mathematical form of the contrastive loss function that enforces similarity between differently augmented samples of the same input x^i is given by:

$$\ell^i = -\log \frac{\exp(sim(y_1^i, y_2^i)/\tau)}{\sum_{k=1}^{2N} 1_{i \neq j} \exp(sim(y^i, y^j)/\tau)}, \qquad sim(u, v) = \frac{u^T v}{||u|| \cdot ||v||} \tag{15}$$

Here, (y_1^i,y_2^i) represents the output of the projection head from Fig. 8 that acts on differently augmented versions x_1^i, x_2^i of the same input x^i. Given a batch of N input examples $\{x^i,\ \forall\ i\ \in\ [1, N]\}$, we have $2N$ similarity examples of the form $\{(x_1^i, x_2^i),\ \forall\ i\ \in\ [1, N]\}$. For each similarity pair of the form (x_1^i, x_2^i), we have $2(N-1)$ dissimilar pairs of the form $\{(x^i, x^j), i \neq j\}$. $1_{i \neq j}$ indicates dissimilar pairs when $i \neq j$, and τ denotes a temperature parameter that controls the penalty strength on dissimilar samples [109]. In our experiments, we set τ to 0.2 to encourage the model to have tolerance for similar samples within a batch. Both similar and dissimilar pairs are considered in evaluating the contrastive loss function in Equation 15 thus training the network to maximize the similarity between similar pairs as well as minimize the similarity between dissimilar pairs. The similarity metric $sim(u, v)$ is also indicated in Equation 15.

Prediction of 3D Finger Joints from Self-Supervised

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3223600

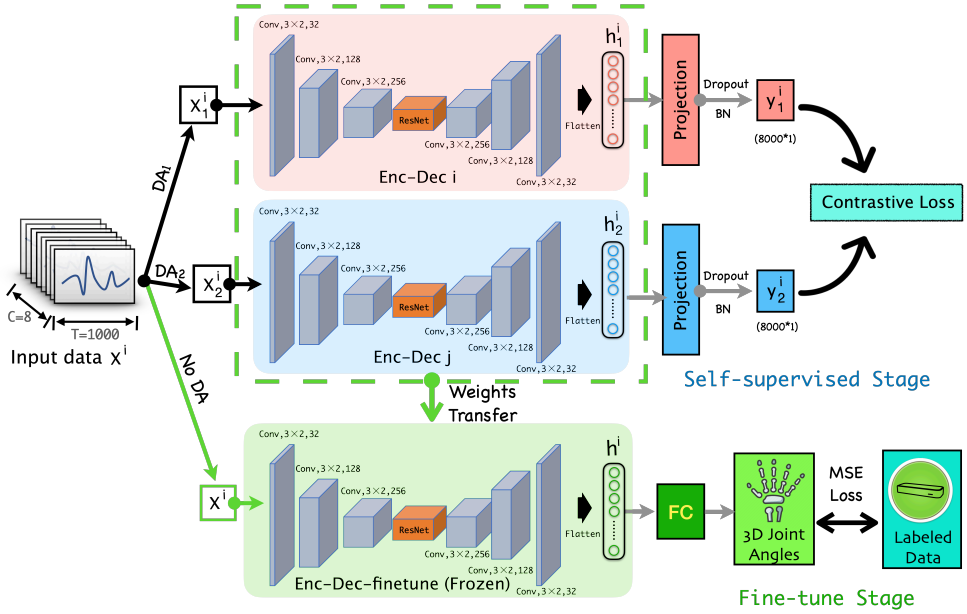


Fig. 8: Architecture for self-supervised and fine-tuning stages (DA = Data Augmentation)

Representations: The representations h learned above based on the architecture in Fig. 8 are used for estimating 3D finger joint angles. This is indicated as the *fine-tune stage* in Fig. 8. The input EMG data is first passed through the encoder-decoder which extracts representations h. For predicting the finger joint angles using h, we follow a widely used evaluation protocol [58] which can be used as a proxy indicator for the efficiency of self-supervised learning. Specifically, a simple linear model with two fully-connected layers takes the representations h and predicts joint angles. The weights of the linear model are trained on top of the encoder-decoder network (encoder-decoder's weights are frozen after *self-supervised stage* in Fig. 8) in a supervised fashion.

VI. 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.

A. 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: Our study was approved by the IRB committee at our institute (STUDY00014754, Pennsylvania State University). 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, because people don't need to remember a sequence of gestures, they can perform any gesture freely.

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 benefit of using leap is it can automatically generate the ground truth labels without having to have human annotation. 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.

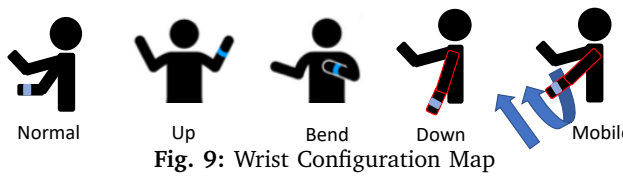


Fig. 9: Wrist Configuration Map

Training Data Collection : Fig. 10 shows qualitative results from *NeuroPose*. 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

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3223600

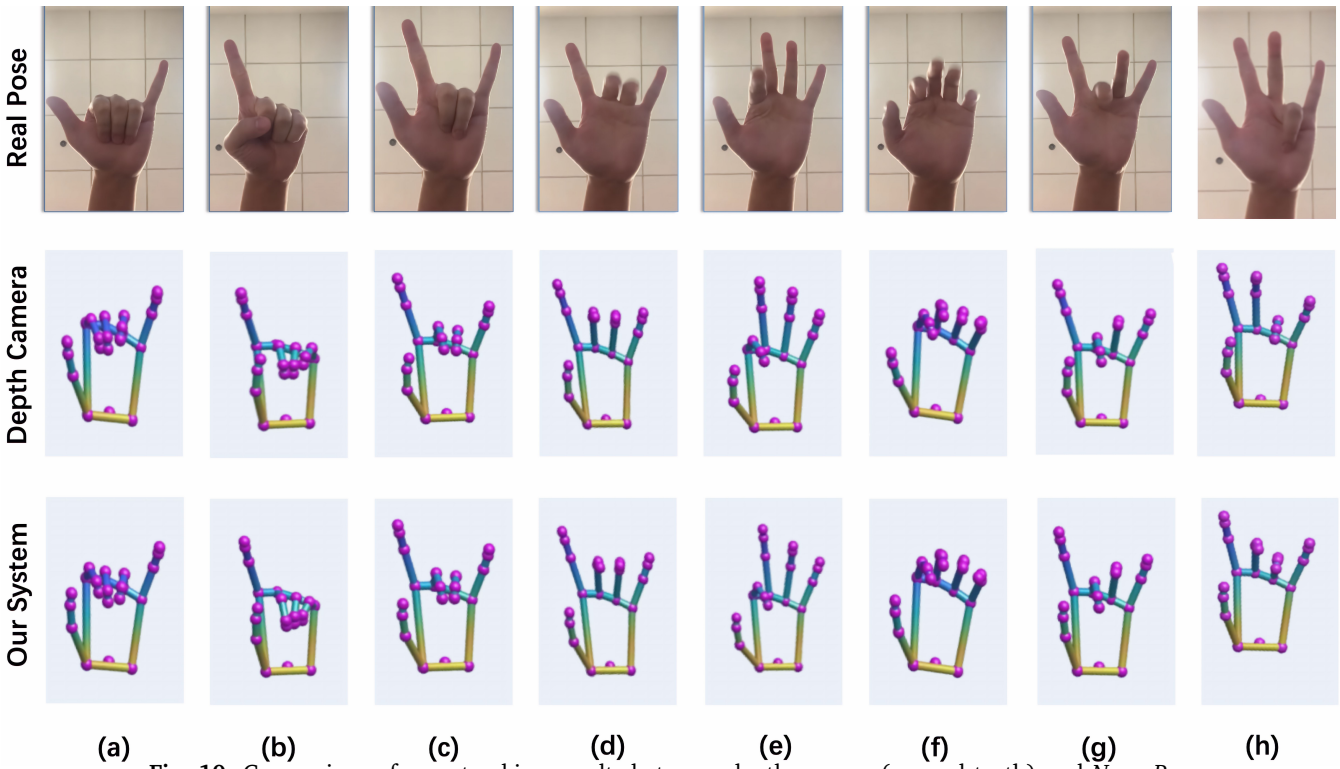


Fig. 10: Comparison of pose tracking results between depth camera (ground truth) and *NeuroPose*.

(wrist maintained at the "normal" position depicted in Fig. 9). For the each of the last 6 sessions, we remove and remount the sensor, to validate robustness of *NeuroPose* to natural changes in sensor position and orientation during typical usage. 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. 9. 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 four 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. Though this model might reach higher accuracy, it requires lots of data from the user side. (ii) **Model with domain adaptation:**To build a model with more generalization, we propose 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 V-B 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. For example, for user No.1, we will train the model with the input of data from users No.2 - No.12.

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 with different sensor rotation angles and positions. (ii) The wrist position is completely different from the one used to train the models.

Hyperparameters of *NeuroPose*: The hyperparameters include learning rate, L2 regularization factors, kernel sizes for convolutional layers, dropout rates, the number of resnet blocks, and the number of convolutional filters per convolutional layer. The above parameters were varied using a grid search as follows: learning rate in the set of $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$, L2 regularization rate were varied in the set $\{0.001, 0.005, 0.01, 0.05\}$, kernel sizes $\{2, 3, 4, 5\}$, dropout rates $\{0.01, 0.05, 0.1, 0.2\}$, number of resnet blocks $\{3, 5, 7\}$, number of convolutional filters in the deep convolutional layers as $\{32, 64, 128, 256\}$. We use randomized cross-validation to tune the hyperparameters for the model, and run multiple cross-validation programs on a campus GPU cluster concurrently.

Learning Curve: Fig. 11 shows the learning curve of *NeuroPose*. The Loss value of both of the training and validation are high due to the L2 regularization at the beginning stage (first 50 epochs), and after that, both loss values decrease sharply. While the training loss is then continuously decreasing, the validation loss is converging

and varying within a certain small range. We also find out that having the residual blocks in between the encoder and decoder will not only decrease the loss values, but also make the validation loss line closer to the training loss line, which means over-fitting problem will be reduced.

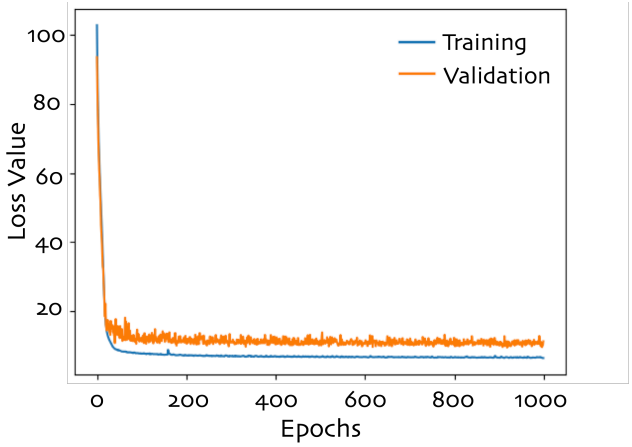


Fig. 11: learning curve

B. 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[57] with a learning rate of 1e-3, β_1 of 0.9 and β_2 of 0.999. To avoid over-fitting issues that may happen in the training process, we apply the L2 regularization[25] on each CONV layer with a parameter of 0.01 and also add dropouts[108] 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 [44]. We perform implementation on three brands of smartphones. This includes two recent brands of smartphones (OnePlus 9 Pro, Samsung S20), and an older model of smartphone (Sony Xperia Z3).

C. 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. 14(a)), and *multi-user models* (Fig. 12). The performance of user-dependent models are also shown separately (Fig. 17). (iii) Combines the Encoder-Decoder architecture (including ResNets) in Section V-A with semi supervised domain adaptation in Section V-B because that is the best performing design with minimal training overhead for different users. The RNN design presented in Section V-C is evaluated separately (Fig. 18). (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 separately (Fig. 15(b)). (v) The error values are in degrees and we choose to show the median and error bars because it gives a quick information about the data distribution. 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].

The predicted hand pose matches closely with reality for a number of example applications including holding virtual objects, ASL signs, pointing gestures etc. Figs. 10(a) to (c) include static positions, whereas Figs. 10(d) to (g) capture the pose while in motion. Fig. 10(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 10(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.12 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.

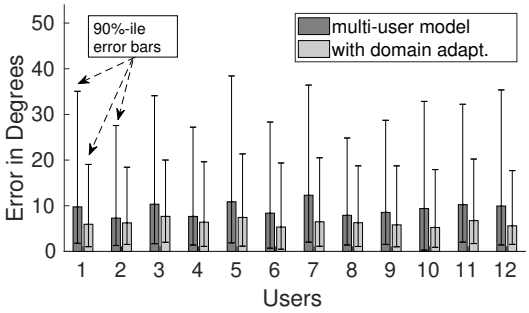


Fig. 12: Domain adaptation significantly reduces errors over users

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.

TABLE I: Robustness to change in sensor position within a day

	Position 1	Position 2	Position 3	Position 4	Position 5	Position 6
10%-ile Error in Degrees	1.03	1.07	1.07	1.12	1.08	1.04
50%-ile Error in Degrees	6.09	6.27	6.30	6.37	6.22	6.16
90%-ile Error in Degrees	17.63	19.20	18.57	18.63	18.86	17.88

Representative Tracking Results over Time: Figure

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3223600

13 shows some representative cases for the comparison between Leap and *NeuroPose* over time. Evidently, *NeuroPose* follows the ground truth accurately, even under sharp changes in the finger angle. This is because the machine learning models can exploit the spatio-temporal relationships within the EMG channels and the finger motion constraints for accurate tracking. These results are consistent with the demo [15], thus providing a stable accuracy of joint angle tracking over time.

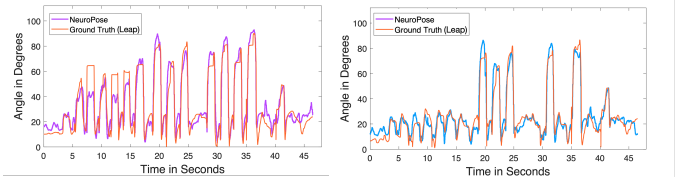


Fig. 13: (a) NeuroPose vs Leap (Index Finger) (b) NeuroPose vs Leap (Little Finger)

Robustness to Natural Variations in Sensor Position and Orientation: We evaluate robustness to natural variations in sensor position by removing the sensor and remounting. Table. I 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). While the sensor was worn naturally during each remounting, minor variations in the sensor mounting position and orientation is expected across sessions. 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. Table. II shows the accuracy when the sensor position change happens across multiple days (with a random wrist position).

TABLE II: Robustness to change in sensor position across days

	Day 1	Day 2	Day 3	Day 4
10%-ile Error in Degrees	1.01	1.12	1.03	0.61
50%-ile Error in Degrees	5.91	6.52	6.30	5.71
90%-ile Error in Degrees	17.73	20.09	18.06	21.33

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 and also maintain the skin temperature to ensure the EMG data has a higher quality [12], we believe this helps in robustness for temperature change over days.

Robustness to Wrist Position and Mobility: Fig. 14(a) shows how the accuracy is consistent despite changes in wrist positions. Users were asked to place their wrist in 4

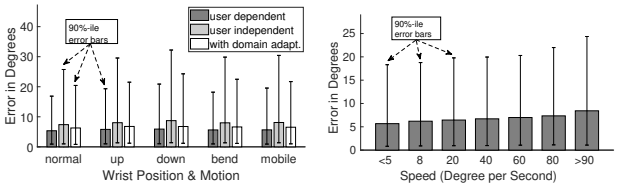


Fig. 14: (a) Robustness to change in wrist positions (b) There is a graceful degradation in accuracy over finger speed

different positions as shown in Fig. 9. *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: Table.III 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.

TABLE III: Accuracy is consistent across fingers

	Thumb	Index	Middle	Ring	Little
10%-ile Error in Degrees	0.68	1.03	1.13	1.12	0.83
50%-ile Error in Degrees	3.81	6.48	6.70	6.04	4.73
90%-ile Error in Degrees	10.53	24.28	22.70	17.63	15.59

Impact of Finger Speed: Fig.14(b) provides a break up of accuracy over various finger speeds. Even at a high finger speed of 90 deg/s there is only a graceful degradation of accuracy suggesting the efficacy of *NeuroPose* in tracking highly dynamic hand poses. This is because the underlying EMG sensor can capture the electric potentials generated by the skeletal muscles in finer detail.

Accuracy over Flex/Extension Joint Angles: Fig.15(a) shows the accuracy breakup between the three flex angles

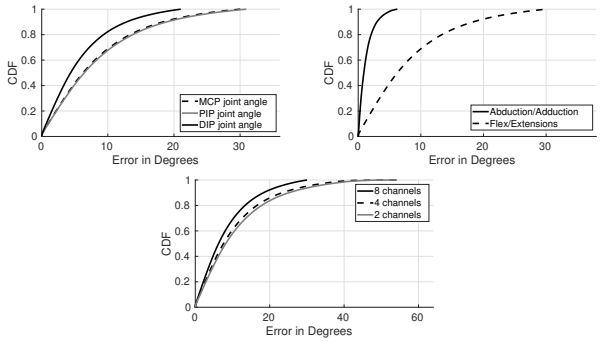


Fig. 15: (a) Accuracy over MCP, PIP, and DIP joints (b) Accuracy over abduction/adductions and flex/extensions (c) Accuracy vs intrusiveness (number of EMG channels)

– $\theta_{mcp,f/e}$, θ_{pip}, and θ_{dip}. Evidently, *NeuroPose* maintains similar accuracy for all joint angles. Fig.15(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.15(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. 16) 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.17(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.17(b) summarizes the result. User dependent training can improve the median error by $1.52°$, the domain adaptation techniques adopted by *NeuroPose* is close to this performance.

Accuracy Breakup by Techniques, Comparison to Prior Work: Fig.18 shows the CDF of error comparisons over various techniques and prior work. Prior work-1[90] 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 [98] uses a RNN architecture with a Simple Recurrent Unit (SRU) and extends [90] 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 hand-poses 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 Comparison over Phone Models: Fig. 19(a) shows the comparison of latency over three different phone models - Sony Xperia Z3, Samsung Galaxy S20, OnePlus 9 Pro. Latency estimates indicate the time elapsed between the actual finger motion and the availability the tracking results in *NeuroPose*. The *NeuroPose* (encoder-resnet-decoder) model takes 5 second sequence of EMG data as input. The inference latency of processing each 5s of data using TensorflowLite on the three different brands of smartphones are 0.067s, 0.012s, and 0.019s respectively. At each instant, by processing the previous 5s of data as input, the model can provide an output in 0.067 seconds even in the worst case of scenario of an older smartphone model (Sony Xperia Z3). This shows how the machine learning models are lightweight thus ensuring real-time performance even on low-end smartphones. However, the encoder-resnet-decoder model will incur a cost of redundant processing to provide real-time performance – we will discuss the tradeoffs (Fig. 19(b)(c)). Furthermore, Fig. 19 (a) depicts the average per-sample processing latency of different techniques – LSTM, encoder-decoder and encoder-resent-decoder (NeuroPose) across all three brands of smartphone models – 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. While the increase in latency with LSTM in comparison to encoder-decoder architecture is $\approx 2x$ for newer smartphones (OnePlus 9 Pro, Samsung S20), the increase in latency can be upto 5x for older smartphones (Sony Xperia Z3).

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 the following two states across all three smartphone models (i) The device is idle with screen on. (ii) The device is making inferences using TensorflowLite model. The idle display-screen on discharge rate $3-5\%$ per hour while the discharge rates for various models is shown in Fig. 19 (b). The power consumption is very low across all brands of phones. 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. Depicted in Fig. 19 (c) this entails continuous/redundant processing thus increasing the discharge rate to $\approx 20\%$ across all phones. The low-power mode trades off real time performance (5s delay) for power savings. Depending on requirements of real-time latency or energy-efficiency, a user can choose between the two modes. The above presented measurements on latency

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2022.3223600

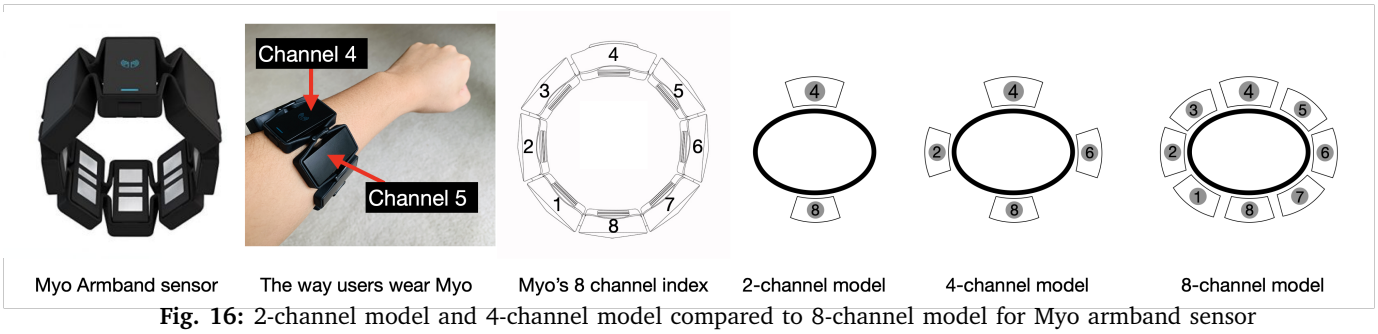


Fig. 16: 2-channel model and 4-channel model compared to 8-channel model for Myo armband sensor

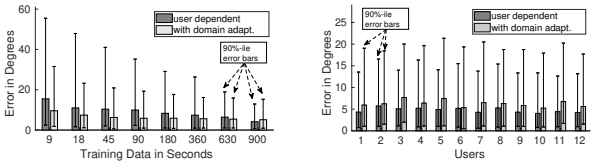


Fig. 17: (a) Domain adaptation minimizes training overhead by an order of magnitude (b) Performance of domain adaptation is close to user dependent training

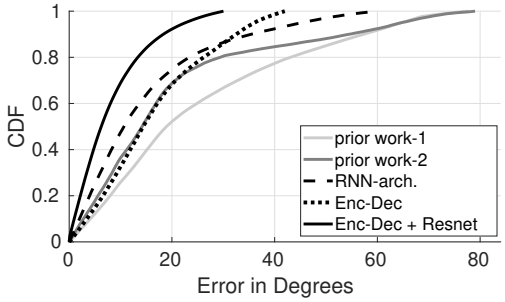


Fig. 18: Encoder-Decoder-ResNet outperforms other techniques

and power consumption show the ability of NeuroPose to perform effectively on a range of embedded smartphone operating systems.

Mirrored Bilateral Training: The right and left hands are mirror images of each other. Thus, the model built from one hand might be usable for the other hand (discussed in Section III-C), provided that the EMG channel numbers are replaced by their corresponding mirror images (For example, from Fig.5, the mirror of channel 5 is channel 3. The exact mirrors of each EMG channel

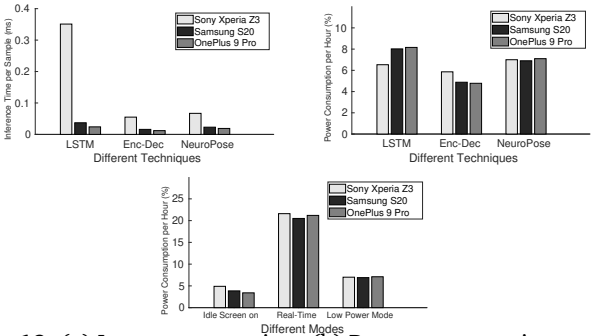


Fig. 19: (a) Latency comparison (b) Power consumption analysis (c) Power consumption across real-time and energy saving modes

is illustrated in Fig. 20). To validate this, we perform more experiments where users perform arbitrary finger motions with *mirrored bilateral training*. The training data from the left hand was then used for performing inferences on test data from the right hand. The ideas in self-supervised learning from Section V-D have been used for processing the sensor data to further reduce the noise due to difference in distribution of data across two hands. Fig.21 shows the performance. The results

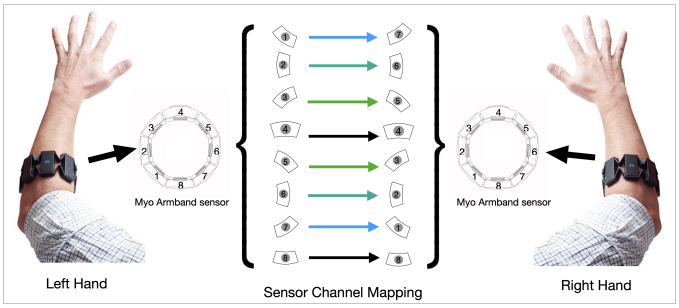


Fig. 20: Mapping of EMG channels for doing inference on the right hand with training data from left hand

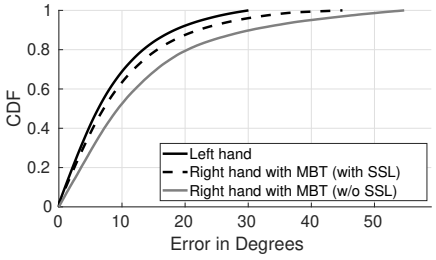


Fig. 21: Model learnt for the left hand is easily adopted for inferences on the right hand with MBT (MBT means Mirrored Bilateral Training, and SSL means self-supervised learning)

of a direct transfer of a left-handed model to the right hand even without the self supervised representation learning is also promising as indicated in the figure. However, with self-supervised representation learning in Section V-D, the errors decrease further. We believe this provides a basic validation of *mirrored bilateral training* [80], and has applications in collecting training data for amputees where training labels cannot be obtained for the hand with missing fingers. Prior research has also shown sufficient neuro-muscular activity is retained despite amputation [41], [35], [80], [81]. Thus, given

the encouraging results on *mirrored bilateral training* with *NeuroPose*, we believe there is promise in extending this work towards developing prosthetic devices for amputees with missing fingers.

Performance Analysis over an Application in Gesture Recognition: *NeuroPose* performs 3D tracking of finger motion with a number of applications in augmented reality, virtual reality, sports analytics, sign language recognition etc. We evaluate the feasibility of *NeuroPose* over a real world application in recognition of alphabets in American Sign Language (ASL) shown in Figure 22(a). Four users were recruited to wear the Myo armband and perform the 26 ASL alphabets 10 times each. Training data was collected from one user who performed these same gestures. For each gesture, we have the accuracy defined as follows:

$$Accuracy = \frac{N_{Correct}}{N_{Total}} \tag{16}$$

where $N_{Correct}$ is the number of times the gesture was detected correctly, and N_{Total} is the total number of the gesture's occurrences in the experiments. The classification was performed by comparing the $\mathbf{R^{21}}$ space of joint angles of the test users with that of the training data. The gesture in the training database with the minimum euclidean distance is declared as the inferred gesture. Fig. 22(b) depicts the confusion matrix of the classification. Evidently, most gestures are classified correctly with an overall average accuracy of 80.22%. Gestures such as A and S are miss classified sometimes because their handpose is similar. This demonstrates the feasibility of using *NeuroPose* in real world applications.

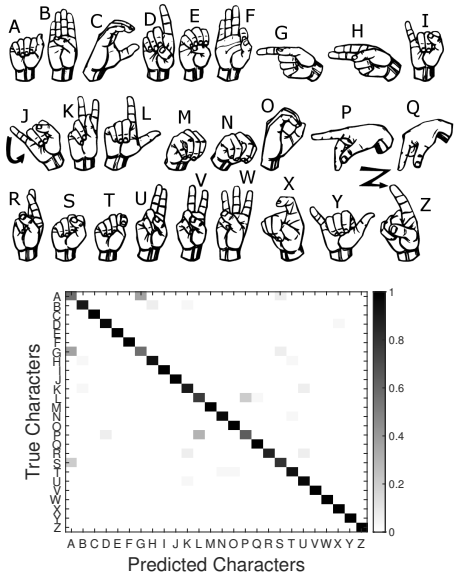


Fig. 22: (a) ASL alphabets (b) Confusion matrix of *NeuroPose*'s performance in ASL alphabet classification

VII. 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 [106] 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 [101] which are relevant for future investigation.

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.

VIII. 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. The feasibility of *mirrored bilateral training* has been shown for 3D finger motion tracking with potential to develop prosthetic devices for amputees. 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] 5dt data glove ultra - 5dt. https://5dt.com/5dt-data-glove-ultra/.
[2] Cyberglove systems llc. http://www.cyberglovesystems.com/.
[3] 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] Forearm muscles : Attachment, nerve supply action. https://anatomyinfo.com/forearm-muscles/.
[5] Industry leading vr techology - manus vr. https://manus-vr.com/.
[6] Knuckleball Grip, Part 3: Depth of the Baseball. "https://knuckleballnation.com/how-to/knuckleballgrip3/".
[7] Leap motion developer. https://developer.leapmotion.com/.
[8] Microsoft kinect2.0. https://developer.microsoft.com/en-us/windows/kinect.
[9] Muscles alive: Information and data sciences. https://www.bu.edu/ids/research-projects/muscles-alive/.

[10] Myo official tutorial. https://support.getmyo.com/hc/en-us/articles/203910089-Warm-up-while-wearing-your-Myo-armband.

[11] 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] Myo warmup. https://support.getmyo.com/hc/en-us/articles/203910089-Warm-up-while-wearing-your-Myo-armband.

[13] powerconsump. https://pdfs.semanticscholar.org/6c0c/af5d51def3730bb746535099252b724ddd31.pdf.

[14] Profile battery usage with batterystats and battery historian. https://developer.android.com/topic/performance/power/setup-battery-historian.

[15] A short video demo of our system. https://www.dropbox.com/s/sssl4e219w2c9al/3D_handpose_demo.avi?dl=0.

[16] 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] ABADI, M., ET AL. Tensorflow: A system for large-scale machine learning. In *OSDI* (2016).

[18] ACHARYA, S., ET AL. Towards a brain-computer interface for dexterous control of a multi-fingered prosthetic hand. In *2007 3rd International IEEE/EMBS Conference on Neural Engineering* (2007).

[19] ADIB, F., ET AL. 3d tracking via body radio reflections. In *USENIX NSDI* (2014).

[20] AHMED, M. A., ET AL. A review on systems-based sensory gloves for sign language recognition state of the art between 2007 and 2017. *Sensors 18*, 7 (2018), 2208.

[21] ARORA, R., ET AL. Understanding deep neural networks with rectified linear units. *arXiv preprint arXiv:1611.01491* (2016).

[22] BADRINARAYANAN, V., KENDALL, A., AND CIPOLLA, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence* (2017), 2481–2495.

[23] BENALCÁZAR, M. E., ANCHUNDIA, C. E., ZEA, J. A., ZAMBRANO, P., JARAMILLO, A. G., AND SEGURA, M. Real-time hand gesture recognition based on artificial feed-forward neural networks and emg. In *2018 26th European Signal Processing Conference (EUSIPCO)* (2018), IEEE, pp. 1492–1496.

[24] BERNDT, D. J., AND CLIFFORD, J. Using dynamic time warping to find patterns in time series. In *KDD workshop* (1994), vol. 10, Seattle, WA, pp. 359–370.

[25] BERTERO, M., DE MOL, C., AND VIANO, G. A. The stability of inverse problems. In *Inverse scattering problems in optics*. Springer, 1980, pp. 161–214.

[26] CAI, Y., GE, L., CAI, J., AND YUAN, J. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *ECCV* (2018).

[27] CAREY, S. L., LURA, D. J., AND HIGHSMITH, M. J. Differences in myoelectric and body-powered upper-limb prostheses: Systematic literature review. *Journal of Rehabilitation Research & Development 52*, 3 (2015).

[28] CHANG, W.-G., ET AL. Domain-specific batch normalization for unsupervised domain adaptation. In *IEEE CVPR* (2019).

[29] CHEN, X., AND WANG, Z. J. Pattern recognition of number gestures based on a wireless surface emg system. *Biomedical Signal Processing and Control 8*, 2 (2013), 184–192.

[30] CHEN CHEN, F., ET AL. Constraint study for a hand exoskeleton: human hand kinematics and dynamics. *Journal of Robotics 2013* (2013).

[31] CHINTALAPUDI, K., ET AL. Indoor localization without the pain. In *ACM MobiCom* (2010).

[32] CONNOLLY, J., ET AL. Imu sensor-based electronic goniometric glove for clinical finger movement analysis. *IEEE Sensors Journal* (2017).

[33] CORDELLA, F., ET AL. 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)* (2012), IEEE, pp. 1967–1972.

[34] Finger and partial hand prosthetic options. https://www.armdynamics.com/our-care/finger-and-partial-hand-prosthetic-options, 2021.

[35] DAVIS, T. S., WARK, H. A., HUTCHINSON, D., WARREN, D. J., O'NEILL, K., SCHEINBLUM, T., CLARK, G. A., NORMANN, R. A., AND GREGER, B. Restoring motor control and sensory feedback in people with upper extremity amputations using arrays of 96 microelectrodes implanted in the median and ulnar nerves. *Journal of neural engineering 13*, 3 (2016), 036001.

[36] DE SILVA, A., ET AL. Real-time hand gesture recognition using temporal muscle activation maps of multi-channel semg signals. *arXiv:2002.03159* (2020).

[37] DEMENTYEV, A., AND PARADISO, J. A. Wristflex: low-power gesture input with wrist-worn pressure sensors. In *ACM UIST* (2014).

[38] DENG, J., ET AL. Imagenet: A large-scale hierarchical image database. In *IEEE CVPR* (2009).

[39] DEVLIN, J., ET AL. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[40] DU, Y., ET AL. Semi-supervised learning for surface emg-based gesture recognition. In *IJCAI* (2017).

[41] GEORGE, J. A., ET AL. Bilaterally mirrored movements improve the accuracy and precision of training data for supervised learning of neural or myoelectric prosthetic control. *arXiv preprint* (2020).

[42] GIESER, S. N., ET AL. 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* (2017), Springer.

[43] GLAUSER, O., ET AL. Interactive hand pose estimation using a stretch-sensing soft glove. *ACM Transactions on Graphics (TOG)* (2019).

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

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

[46] HAN, X., ET AL. Pre-trained alexnet architecture with pyramid pooling and supervision for high spatial resolution remote sensing image scene classification. *Remote Sensing* (2017).

[47] HASAN, S., AND LINTE, C. A. U-netplus: a modified encoder-decoder u-net architecture for semantic and instance segmentation of surgical instrument. *arXiv preprint arXiv:1902.08994* (2019).

[48] HE, K., ET AL. Deep residual learning for image recognition. In *IEEE CVPR* (2016).

[49] HU, F., ET AL. 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).

[50] HUININK, L. H., BOUWSEMA, H., PLETTENBURG, D. H., VAN DER SLUIS, C. K., AND BONGERS, R. M. Learning to use a body-powered prosthesis: changes in functionality and kinematics. *Journal of neuroengineering and rehabilitation 13*, 1 (2016), 1–12.

[51] IOFFE, S., ET AL. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).

[52] IQBAL, U., ET AL. Hand pose estimation via latent 2.5 d heatmap regression. In *ECCV* (2018).

[53] JAEGER, H. *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the" echo state network" approach*, vol. 5. GMD-Forschungszentrum Informationstechnik Bonn, 2002.

[54] JIANG, N., VEST-NIELSEN, J. L., MUCELI, S., AND FARINA, D. Emg-based simultaneous and proportional estimation of wrist/hand kinematics in uni-lateral trans-radial amputees. *Journal of neuroengineering and rehabilitation 9*, 1 (2012), 1–11.

[55] KIENZLE, W., WHITMIRE, E., RITTALER, C., AND BENKO, H. Electroring: Subtle pinch and touch detection with a ring. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (2021), pp. 1–12.

[56] KIM, D., ET AL. Digits: freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In *ACM UIST* (2012).

[57] KINGMA, D. P., AND BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[58] KOLESNIKOV, A., ZHAI, X., AND BEYER, L. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 1920–1929.

[59] KOSHIO, T., ET AL. Identification of surface and deep layer muscles activity by surface emg. In *2012 Proceedings of SICE Annual Conference* (2012), IEEE.

[60] KRIZHEVSKY, A., ET AL. Imagenet classification with deep convolutional neural networks. In *NIPS* (2012).

[61] LAWRENCE, S., ET AL. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks* (1997).

[62] LAYONA, R., ET AL. Web based augmented reality for human body anatomy learning. *Procedia Computer Science* (2018).

[63] LI, H., YANG, W., WANG, J., XU, Y., AND HUANG, L. Wifinger: talk to your smart devices with finger-grained gesture. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (2016), ACM, pp. 250–261.

[64] LIEW, S. S., ET AL. Bounded activation functions for training stability of deep neural networks on visual pattern recognition problems. *Neurocomputing* (2016).

[65] LIN, B.-S., ET AL. Design of an inertial-sensor-based data glove for hand function evaluation. *Sensors* (2018).

[66] LIN, J., WU, Y., AND HUANG, T. S. Modeling the constraints of human hand motion. In *Proceedings workshop on human motion* (2000), IEEE, pp. 121–126.

[67] LIU, J., ET AL. uwave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing* (2009).

[68] LIU, Y., ZHANG, S., AND GOWDA, M. When video meets inertial sensors: Zero-shot domain adaptation for finger motion analytics with inertial sensors. In *Proceedings of the International Conference on Internet-of-Things Design and Implementation* (2021), pp. 182–194.

[69] LUO, X. Y., WU, X. Y., CHEN, L., HU, N., ZHANG, Y., ZHAO, Y., HU, L. T., YANG, D. D., AND HOU, W. S. Forearm muscle synergy reducing dimension of the feature matrix in hand gesture recognition. In *2018 3rd International Conference on Advanced Robotics and Mechatronics (ICARM)* (2018), IEEE, pp. 691–696.

[70] MA, Y., ZHOU, G., WANG, S., ZHAO, H., AND JUNG, W. Signfi: Sign language recognition using wifi. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 2*, 1 (2018), 23.

[71] MANCINI, M., ET AL. Boosting domain adaptation by discovering latent domains. In *IEEE CVPR* (2018).

[72] MAUDRICH, T., KENVILLE, R., LEPSIEN, J., VILLRINGER, A., RAGERT, P., AND STEELE, C. J. Mirror electromyografic activity in the upper and lower extremity: a comparison between endurance athletes and non-athletes. *Frontiers in human neuroscience 11* (2017), 485.

[73] MCINTOSH, J., ET AL. Echoflex: Hand gesture recognition using ultrasound imaging. In *2017 CHI Conference on Human Factors in Computing Systems* (2017).

[74] MELGAREJO, P., ZHANG, X., RAMANATHAN, P., AND CHU, D. Leveraging directional antenna capabilities for fine-grained gesture recognition. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (2014), ACM, pp. 541–551.

[75] MIKOLOV, T., ET AL. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association* (2010).

[76] MUELLER, F., ET AL. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *IEEE CVPR* (2018).

[77] MURGUIALDAY, A. R., ET AL. Brain-computer interface for a prosthetic hand using local machine control and haptic feedback. In *2007 IEEE 10th International Conference on Rehabilitation Robotics* (2007), IEEE.

[78] NANDAKUMAR, R., ET AL. Fingerio: Using active sonar for fine-grained finger tracking. In *ACM CHI* (2016).

[79] NAWAZ, W., ET AL. Classification of breast cancer histology images using alexnet. In *International conference image analysis and recognition* (2018), Springer.

[80] NIELSEN, J. L., ET AL. Simultaneous and proportional force estimation for multifunction myoelectric prostheses using mirrored bilateral training. *IEEE Transactions on Biomedical Engineering* (2010).

[81] PAN, L., ET AL. Continuous estimation of finger joint angles under different static wrist motions from semg signals. *Biomedical Signal Processing and Control* (2014).

[82] PARATE, A., ET AL. Risq: Recognizing smoking gestures with inertial sensors on a wristband. In *ACM MobiSys* (2014).

[83] PARIZI, F. S., WHITMIRE, E., AND PATEL, S. Auraring: Precise electromagnetic finger tracking. *ACM IMWUT* (2019).

[84] PARK, K., KIM, S., YOON, Y., KIM, T.-K., AND LEE, G. Deepfisheye: Near-surface multi-finger tracking technology using fisheye camera. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (2020), pp. 1132–1146.

[85] PASCANU, R., MIKOLOV, T., AND BENGIO, Y. Understanding the exploding gradient problem. *CoRR, abs/1211.5063 2* (2012).

[86] PEÑA PITARCH, E. *Virtual human hand: Grasping strategy and simulation*. Universitat Politècnica de Catalunya, 2008.

[87] POLYGERINOS, P., ET AL. Emg controlled soft robotic glove for assistance during activities of daily living. In *2015 IEEE international conference on rehabilitation robotics (ICORR)* (2015), IEEE.

[88] PU, Q., GUPTA, S., GOLLAKOTA, S., AND PATEL, S. Whole-home gesture recognition using wireless signals. In *Proceedings of the 19th annual international conference on Mobile computing & networking* (2013), ACM, pp. 27–38.

[89] QU, C., ET AL. Bert with history answer embedding for conversational question answering. In *ACM SIGIR Conference on Research and Development in Information Retrieval* (2019).

[90] QUIVIRA, F., ET AL. Translating semg signals to continuous hand poses using recurrent neural networks. In *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)* (2018), IEEE.

[91] RAURALE, S., ET AL. Emg acquisition and hand pose classification for bionic hands from randomly-placed sensors. In *IEEE ICASSP* (2018).

[92] RODA-SALES, A., ET AL. Effect on manual skills of wearing instrumented gloves during manipulation. *Journal of biomechanics* (2020).

[93] SANTHALINGAM, P. S., DU, Y., WILKERSON, R., ZHANG, D., PATHAK, P., RANGWALA, H., KUSHALNAGAR, R., ET AL. Expressive asl recognition using millimeter-wave wireless signals. In *2020 17th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)* (2020), IEEE, pp. 1–9.

[94] SCHABRON, B., ALASHQAR, Z., FUHRMAN, N., JIBBE, K., AND DESAI, J. Artificial neural network to detect human hand gestures for a robotic arm control. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (2019), IEEE, pp. 1662–1665.

[95] SCHEGGI, S., ET AL. Touch the virtual reality: using the leap motion controller for hand tracking and wearable tactile devices for immersive haptic rendering. In *ACM SIGGRAPH Posters*. 2015.

[96] SHANG, J., AND WU, J. A robust sign language recognition system with multiple wi-fi devices. In *Proceedings of the Workshop on Mobility in the Evolving Internet Architecture* (2017), ACM, pp. 19–24.

[97] SHERMAN, M., ET AL. User-generated free-form gestures for authentication: Security and memorability. In *ACM MobiSys* (2014).

[98] SOSIN, I., ET AL. 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)* (2018), IEEE.

[99] STAPORNCHAISIT, S., KIM, Y., TAKAGI, A., YOSHIMURA, N., AND KOIKE, Y. Finger angle estimation from array emg system using linear regression model with independent component analysis. *Frontiers in Neurorobotics 13* (2019).

[100] STASHUK, D. Emg signal decomposition: how can it be accomplished and used? *Journal of Electromyography and Kinesiology 11*, 3 (2001), 151–173.

[101] SUN, B., AND SAENKO, K. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision* (2016), Springer.

[102] SUN, W., LI, F. M., HUANG, C., LEI, Z., STEEPER, B., TAO, S., TIAN, F., AND ZHANG, C. Thumbtrak: Recognizing micro-finger poses using a ring with proximity sensing. *arXiv preprint arXiv:2105.14680* (2021).

[103] SUSANTO, E. A., ET AL. Efficacy of robot-assisted fingers training in chronic stroke survivors: a pilot randomized-controlled trial. *Journal of neuroengineering and rehabilitation* (2015).

[104] TRUONG, H., ET AL. Capband: Battery-free successive capacitance sensing wristband for hand gesture recognition. In *ACM SenSys* (2018).

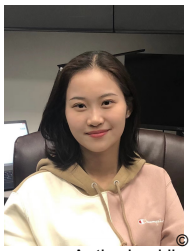
[105] TUNG, Y.-C., AND SHIN, K. G. Echotag: Accurate infrastructure-free indoor location tagging with smartphones. In *ACM MobiCom* (2015).

[106] TZENG, E., ET AL. Adversarial discriminative domain adaptation. In *CVPR* (2017).

[107] UTTNER, I., KRAFT, E., NOWAK, D. A., MÜLLER, F., PHILIPP, J., ZIERDT, A., AND HERMSDÖRFER, J. Mirror movements and the role of handedness: isometric grip forces changes. *Motor control 11*, 1 (2007).

[108] WAGER, S., WANG, S., AND LIANG, P. S. Dropout training as adaptive regularization. In *Advances in neural information processing systems* (2013), pp. 351–359.
[109] WANG, F., AND LIU, H. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 2495–2504.
[110] WANG, J., ET AL. Ubiquitous keyboard for mobile devices: harnessing multipath fading for fine-grained keystroke localization. In *ACM MobiCom* (2014).
[111] WANG, S., SONG, J., LIEN, J., POUPYREV, I., AND HILLIGES, O. Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (2016), pp. 851–860.
[112] Web xr device api. https://www.w3.org/TR/webxr/, 2021.
[113] WINKEL, J., ET AL. Significance of skin temperature changes in surface electromyography. *European journal of applied physiology and occupational physiology* (1991).
[114] WU, E., YUAN, Y., YEO, H.-S., QUIGLEY, A., KOIKE, H., AND KITANI, K. M. Back-hand-pose: 3d hand pose estimation for a wrist-worn camera via dorsum deformation network. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (2020), pp. 1147–1160.
[115] XIONG, J., AND JAMIESON, K. Arraytrack: A fine-grained indoor location system. In *USENIX NSDI* (2013).
[116] XU, S., ET AL. The effectiveness of virtual reality in safety training: Measurement of emotional arousal with electromyography. In *ISARC* (2019).
[117] ZHANG, C., ET AL. Fingerping: Recognizing fine-grained hand poses using active acoustic on-body sensing. In *ACM CHI* (2018).
[118] ZHANG, H., XU, J., AND WANG, J. Pretraining-based natural language generation for text summarization. *arXiv preprint arXiv:1902.09243* (2019).
[119] ZHANG, Y., ET AL. Tomo: Wearable, low-cost electrical impedance tomography for hand gesture recognition. In *ACM UIST* (2015).
[120] ZHAO, M., ET AL. Through-wall human mesh recovery using radio signals. In *IEEE CVPR* (2019).
[121] ZHOU, P., ET AL. Use it free: Instantly knowing your phone attitude. In *ACM MobiCom* (2014).
[122] ZHOU, Z., ET AL. Fine-tuning convolutional neural networks for biomedical image analysis: actively and incrementally. In *IEEE CVPR* (2017).

Yilin Liu Yilin Liu is a PhD student in department of Computer Science and Engineering at Penn State. He is working with Prof. Mahanth Gowda. He obtained his Bachelors in Electronic Engineering from University of Science and Technology of China(USTC). His research interests include IoT and human behavioral sensing, machine learning and deep learning applications, wireless networking and mobile computing. He had his publications in top conferences like WWW, IMWUT, IoTDI, etc.



Shijia Zhang Shijia Zhang is a PhD student in department of Computer Science and Engineering at Penn State. She is working with Prof. Mahanth Gowda. She obtained her Bachelors in Statistics from University of Science and Technology of China(USTC). Her research interests include IoT and machine learning and deep learning applications and wearable devices. She had publications in top conferences like IoTDI, WWW, etc.

Mahanth Gowda Mahanth Gowda is an Assistant Professor in Computer Science and Engineering at Penn State. He obtained his PhD from University of Illinois at Urbana Champaign and Bachelors from Indian Institute of Technology, Varanasi. His research interests include wireless networking, mobile sensing, and wearable computing, with applications to IoT, cyber physical systems, and human gesture recognition. He is a recipient of a NSF CAREER Award in 2021 and a best paper award in IoTDI 2021. He has published across diverse research forums including NSDI, MobiCom, WWW, Oakland, Hotnets, ASPLOS, etc.