Published: 08 December 2022

RESEARCH-ARTICLE

Leveraging the Properties of mmWave Signals for 3D Finger Motion Tracking for Interactive IoT Applications

YILIN LIU, Pennsylvania State University, University Park, PA, United States

SHIJIA ZHANG, Pennsylvania State University, University Park, PA, United States

MAHANTH GOWDA, Pennsylvania State University, University Park, PA, United States

SRIHARI NELAKUDITI, University of South Carolina, Columbia, SC, United States

Proceedings of the ACM on Measurement and Analysis of Computing Systems, Volume 6, Issue 3 (December 2022)
https://doi.org/10.1145/3570613
EISSN: 2476-1249

Leveraging the Properties of mmWave Signals for 3D Finger Motion Tracking for Interactive IoT Applications

YILIN LIU, Pennsylvania State University, USA
SHIJIA ZHANG, Pennsylvania State University, USA
MAHANTH GOWDA, Pennsylvania State University, USA
SRIHARI NELAKUDITI, University of South Carolina, USA

mmWave signals form a critical component of 5G and next-generation wireless networks, which are also being increasingly considered for sensing the environment around us to enable ubiquitous IoT applications. In this context, this paper leverages the properties of mmWave signals for tracking 3D finger motion for interactive IoT applications. While conventional vision-based solutions break down under poor lighting, occlusions, and also suffer from privacy concerns, mmWave signals work under typical occlusions and non-line-of-sight conditions, while being privacy-preserving. In contrast to prior works on mmWave sensing that focus on predefined gesture classification, this work performs continuous 3D finger motion tracking. Towards this end, we first observe via simulations and experiments that the small size of fingers coupled with specular reflections do not yield stable mmWave reflections. However, we make an interesting observation that focusing on the forearm instead of the fingers can provide stable reflections for 3D finger motion tracking. Muscles that activate the fingers extend through the forearm, whose motion manifests as vibrations on the forearm. By analyzing the variation in phases of reflected mmWave signals from the forearm, this paper designs *mm4Arm*, a system that tracks 3D finger motion. Nontrivial challenges arise due to the high dimensional search space, complex vibration patterns, diversity across users, hardware noise, etc. *mm4Arm* exploits anatomical constraints in finger motions and fuses them with machine learning architectures based on encoder-decoder and ResNets in enabling accurate tracking. A systematic performance evaluation with 10 users demonstrates a median error of 5.73° (location error of 4.07 mm) with robustness to multipath and natural variation in hand position/orientation. The accuracy is also consistent under non-line-of-sight conditions and clothing that might occlude the forearm. *mm4Arm* runs on smartphones with a latency of 19*ms* and low energy overhead.

CCS Concepts: • **Human-centered computing** → **Mobile devices**; **Ubiquitous and mobile computing design and evaluation methods**; • **Computing methodologies** → *Neural networks.*

Additional Key Words and Phrases: IoT; Wireless signal; Finger tracking; mmWave sensing

ACM Reference Format:
Yilin Liu, Shijia Zhang, Mahanth Gowda, and Srihari Nelakuditi. 2022. Leveraging the Properties of mmWave Signals for 3D Finger Motion Tracking for Interactive IoT Applications. *Proc. ACM Meas. Anal. Comput. Syst.* 6, 3, Article 52 (December 2022), 28 pages. https://doi.org/10.1145/3570613

Authors' addresses: Yilin Liu, Pennsylvania State University, USA, yzl470@psu.edu; Shijia Zhang, Pennsylvania State University, USA, sbz5188@psu.edu; Mahanth Gowda, Pennsylvania State University, USA, mahanth.gowda@psu.edu; Srihari Nelakuditi, University of South Carolina, USA, srihari@cse.sc.edu.

2476-1249/2022/12-ART52 $15.00
https://doi.org/10.1145/3570613

1 INTRODUCTION

Wireless signals, which are mainly used for communication networks, also have the potential to extend our senses, enabling us to see behind closed doors and track moving objects through walls [21, 61]. Accordingly, there is a growing interest in the community recently to develop novel IoT applications for sensing by exploiting radio frequency signals [30, 44, 45]. Given the compact size of modern wireless devices, this enables ubiquitous applications in the areas of smart healthcare, sports analytics, AR/VR etc. Specifically, as these signals travel in the medium, they traverse occlusions and bounce off different objects before arriving at a receiver; hence, the reflected signals carry information about the environment. By exploiting this property, this paper shows the feasibility of tracking precise 3D finger motion using mmWave signals that are popularly used in 5G networks.

Motivation and Application: This paper presents *mm4Arm*, a system that quantifies the performance of finger motion tracking for interactive applications using mmWave signals through a carefully designed simulation and measurement study. We considered using mmWave signal because FMCW-based radars are being used for ubiquitous applications in the areas of smart healthcare [23], sports analytics, AR/VR [131], autonomous driving [25], etc. Similar to the popular Google Soli platform [117], our main motivation is to enable wearable, mobile computing, and AR/VR applications where conventional touch interaction may be hard. Finger motion-based interfaces over the air are known to be a popular form of human-computer interaction [58, 107]. In contrast to Soli, which can only detect 11 predefined gestures, *mm4Arm* can perform arbitrary 3D motion tracking, thus allowing highly precise control. Decades of prior research have shown that such a finer control can enable rapid and fluid manipulation for highly intuitive interaction [124]. The finer precision of control can be observed in the case of a fluid expert interaction with hand tools (e.g. watchmaker). We believe *mm4Arm*'s accuracy can allow such a finer control. Therefore, regardless of the application, we focus on enabling the core motion tracking framework by solving the underlying challenges. We envision interesting applications of *mm4Arm*, such as developing prosthetic devices for amputees considering that forearm vibrations remain intact despite amputations [35, 85, 87], and discuss them in Section 9. We leave a thorough investigation of the application space for future research.

Radio Frequency (RF) Sensing vs. Vision: Recent works [28, 51, 82] track 3D finger motion using cameras placed in the environment. Powered by the latest advances in machine learning combined with the availability of large-scale training data, precise tracking is possible. However, cameras are susceptible to occlusions, lighting conditions, and interference from objects in the background. Furthermore, they are known to suffer from privacy concerns [31]. In contrast, RF sensing based on mmWave signals as performed by *mm4Arm* can be privacy-preserving and agnostic to lighting, resolution, and ambient conditions. Furthermore, RF sensing can work through materials and non-line-of-sight conditions, allowing it to be embedded into devices and environments.

Tracking Fingers by Observing the ForeArm: In this paper, we not only focus on tracking the 3D finger motion using mmWave reflections, but based on observations via simulations and measurements, we also identify the underlying conditions that enable precise tracking. A critical observation is that the small size of fingers does not provide stable reflections to the level required for tracking. However, the data-driven analysis reveals that it is possible to indirectly track fingers by measuring reflections from the forearm. Finger motion activation involves neuro-muscular interactions, which induce minute muscular motions in the forearm. Such muscular motion produces vibrations in the forearm. Thanks to the short wavelength of mmWave signals, the *phase* measurements are extremely sensitive to small vibrations (up to $0.63\mu m$ [53]), thus opening up opportunities for precise motion tracking. Moreover, the forearm offers a rich texture and curvature

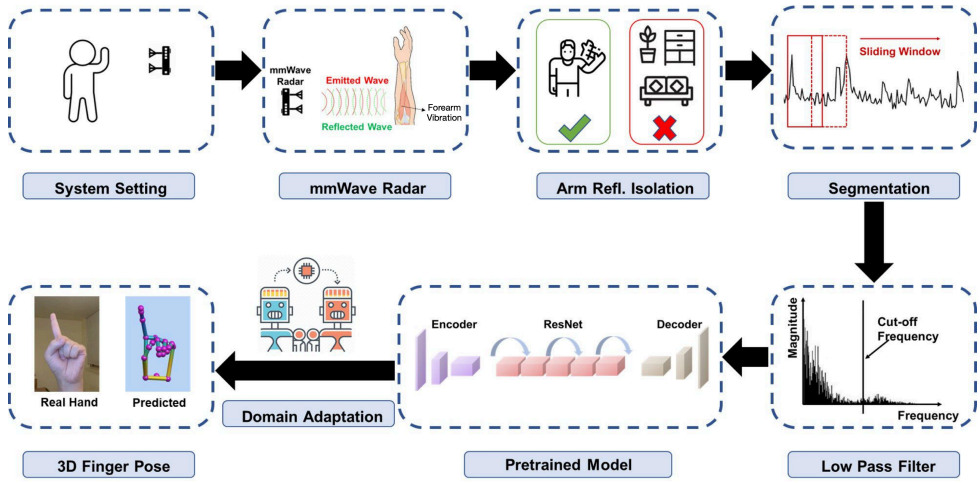


Fig. 1. mmWave reflections are captured from the surroundings from which the *phases* of arm reflections are first isolated. After subjecting the *phase* measurements to preprocessing techniques like low pass filters, deep learning based models are designed for extracting 3D finger motion from the *phase* data. Domain adaptation is incorporated in the design for decreasing the training overhead.

and a much bigger surface for reflections, in contrast to the small size of fingers, which facilitates robust tracking. *mm4Arm* analyzes such forearm vibrations for 3D finger motion tracking.

We reiterate two critical observations made in this paper: (i) When 3D finger motion tracking is of interest in contrast to predefined gesture classification, the reflections obtained directly from fingers do not provide sufficient information. Very few reflections come back to the radar due to the small size of fingers and dominant specular reflections. A similar observation on specularity has been made earlier in the context of autonomous cars [26, 93]. (ii) Vibrations in the forearm during finger motion can capture rich information. Because of the large surface of the forearm and its curvature, the reflections are more stable and robust to natural variation in arm position, height, and orientation. This can be leveraged for 3D finger motion tracking.

Contrast with Key Prior Work: As noted earlier, prior works on finger motion tracking with radar devices are limited to discrete gesture classification. Google Soli [117] exploits reflections from mmWave signals in combination with deep convolutional and recurrent neural networks to track 11 finger motion gestures. mmASL [102] shows the feasibility of detecting 50 ASL gestures using reflections of mmWave signals. mHomeGes [72] uses mmWave signals for tracking 10 hand gestures for user interface applications in settings like smart home. RFWash [55] makes a creative use of mmWave radars for detecting hygienic methods of handwashing and alerting users accordingly. In contrast to gesture and activity classification where the search space is 10-50 predefined discrete classes, *mm4Arm*'s search space is a continuous space of 3D finger motion with 21 *degrees of freedom*. The 3D finger locations predicted by *mm4Arm* can serve as inputs to any gesture classification problem – independent of a specific application. To the best of our knowledge, *mm4Arm* is the first work to perform continuous 3D finger motion tracking using RF signals.

Challenges and Opportunities: Performing 3D finger motion tracking by sensing forearm vibrations is non-trivial with many challenges: (i) As mentioned above, the search space for the correct hand pose is high dimensional with 21 degrees of freedom. The complexity is comparable to human skeleton tracking; (ii) The vibrations due to motion of individual fingers merge into each other with complex patterns; (iii) The vibration pattern can vary among users, body sizes, anatomy, etc. While these challenges seem daunting, *mm4Arm* exploits a number of opportunities to overcome the challenges: (i) *mm4Arm* leverages anatomical constraints in finger motion towards

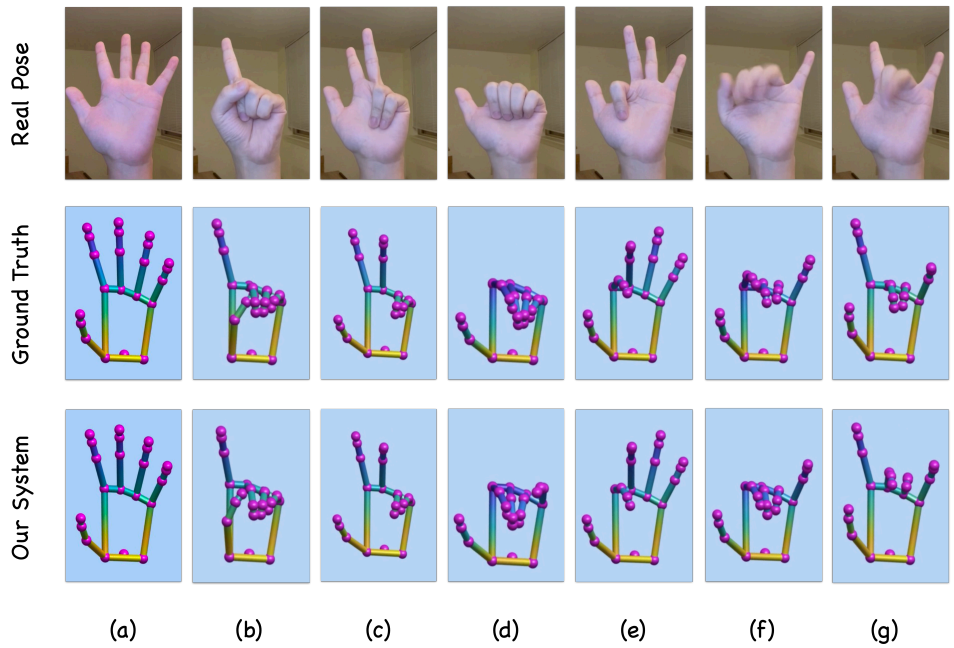


Fig. 2. We present an approach for 3D finger motion tracking using mmWave signals. The figure shows a comparison between several real hand poses and the corresponding tracking results from a depth camera and our proposed system, *mm4Arm*. A short demo is included in this anonymous url [10].

narrowing down the search space of finger motion; (ii) Machine learning (ML) models are designed by incorporating advances in encoder-decoder and skip connections for learning the complex interrelationships between finger motions and the phase measurements while working with limits of training data availability and stability in convergence; (iii) Domain adaptation techniques are designed to develop a robust inferencing model for each user with low training overhead.

System Design: Fig. 1 illustrates the high-level architecture of *mm4Arm*. The radar illuminates the environment and captures reflections from the forearm and other objects in the environment. The forearm reflections are first isolated from other multipath components (wall, furniture, body, etc) based on characteristic phase variation in the forearm reflections. The phase data from forearm reflections are then preprocessed with techniques like low pass filtering for eliminating high-frequency noise. Finally, an encoder-decoder based ML model processes the phase data and generates 3D finger motion sequences by exploiting spatio-temporal constraints of hand motion.

Implementation: *mm4Arm* is implemented using an off-the-shelf radar TI IWR6843ISK[7] operating at 60 GHz band using frequency modulated carrier wave (FMCW). The radar sensor data is pre-processed offline with MATLAB/python, and fed to ML modules implemented in TensorFlow for 3D finger motion tracking. The median error is 5.73 degrees (location error of 4.07mm), validated under a systematic user study with 10 users. The accuracy degrades gracefully with the distance of the user from the radar (evaluated upto 5ft) with robustness to environmental multipath and natural changes in arm position, height and orientation. The accuracy is also consistent under non-line-of-sight conditions and clothing. *mm4Arm* is implemented on modern smartphones - Samsung Galaxy S20, OnePlus 9 Pro – with low power consumption and a latency $\approx$ 19ms.

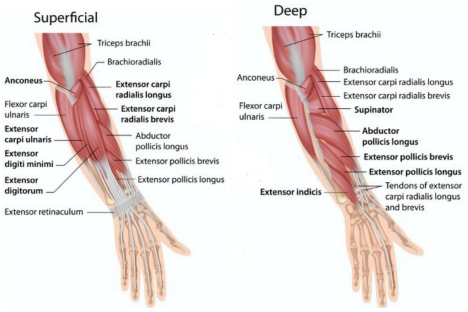
Contributions: We make the following contributions. (i) Feasibility of finger motion tracking by exploiting reflections from the forearm (ii) Free-from 3D finger motion tracking for arbitrary hand motion with mmWave radar. (iii) Design of ML models that fuse anatomical constraints of finger motion with sensor data for accurate 3D finger motion tracking. (iv) A systematic validation with 10 users and implementation on embedded operating systems. Fig. 2 depicts some examples of *mm4Arm*'s tracking quality. A short demo is included in the anonymous url [10].

2 BACKGROUND

We begin with a brief overview of: (i) Relationship between finger motion and forearm vibration. (ii) Anatomical constraints of the human hand to be incorporated in ML models for narrowing down the search space for 3D finger motion tracking.

2.1 Relationship between Finger Motion and Forearm Vibration

Muscles responsible for motion of fingers are located in the forearm (Fig. 3). Depending upon which fingers and the manner in which they need to move, a unique pattern of muscles in the forearm are activated, thus inducing minute vibration in the forearm. *mm4Arm* tracks such forearm vibrations for 3D finger motion tracking. We now provide a brief overview of forearm muscular involvement during finger motions. Several muscles are involved in performing finger motions. Fig. 3a and 3b depict the anatomical structure of the forearm where the muscles move. *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 respectively for adduction and abduction of index, ring, and little fingers towards/away from the middle finger. They are connected to the *proximal phalanx* and *Extensor digitorum*. Other muscles that are involved in large scale motion and supporting strength include *Supinator* for forearm motion, *Anconeus and Brachioradiali* for elbow joint, *Extensor Carpi Ulnaris*, *Extensor Carpi Radialis Longus and Brevis* for wrist joint etc. The motion of these muscles in the forearm induces vibrations in the forearm. The pattern of vibration is a function of what muscles need to move to activate a specific finger motion pattern. *mm4Arm* exploits such forearm vibrations for tracking 3D finger motion.



(a) (b)

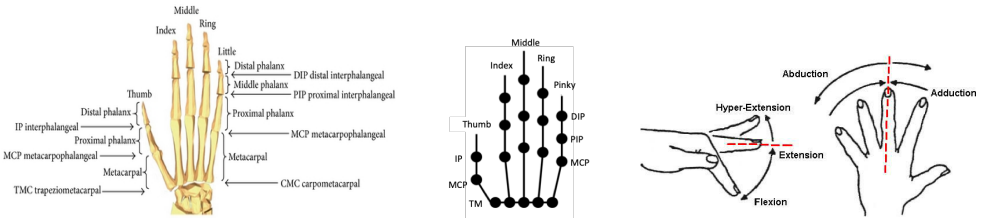
Fig. 3. Muscles responsible for finger motion are located in the forearm [4]. Movement of these muscles causes the forearm to vibrate during finger motion.

2.2 Hand Skeletal Model and Constraints

Human hand consists of various joints that are responsible for complex articulation patterns that generate 3D hand poses. Fig.4a depicts the skeletal structure. A simplified kinematic view is shown in Fig. 4b.

The four fingers consist of three joints: (i) MCP (metacarpophalangeal), (ii) PIP (proximal interphalangeal), and (iii) DIP (distal interphalangeal) joints. The joints at PIP (ϕ_{pip}) and DIP (ϕ_{dip}) can either flex or extend (Fig.4c) the fingers towards or away from the palm. Thus, they exhibit a single degree of freedom (DoF). In contrast, the MCP joint can also undergo adduction and

abduction (side-way motions depicted in Fig.4c) in addition to flexing/extending. Thus, an MCP joint possesses two DoFs, denoted by $\phi_{mcp,f/e}$, and $\phi_{mcp,aa}$ respectively. Thus, each of the four fingers possesses four DoF. On the other hand, the thumb exhibits a slightly different anatomical structure than the other fingers. The MCP and TM (trapeziometacarpal) joints possess both flex and abduction/adduction DoF. The IP (interphalangeal) joint can flex or extend with a single DoF (ϕ_{ip}). Thus, the thumb has five DoF – ϕ_{ip}, $\phi_{mcp,f/e}$, $\phi_{mcp,aa}$, $\phi_{tm,f/e}$, and $\phi_{tm,aa}$. 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 ($\mathbb{R}^{21}$). Finger motion follows certain constraints. As studied in literature,



(a) (b) (c)

Fig. 4. (a) Anatomical details of the hand skeleton [32] (b) Joint notations [70] (c) Flex/extensions and abduction/adductions in finger motion [90]

the joint angles exhibit a high degree of correlation and interdependence [32, 70]. Some of the intra-finger constraints are enumerated below:

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

where ϕ_{dip} denotes angles of the DIP joints, ϕ_{ip} denotes angles of the thumb's IP joints, $\phi_{mcp,f/e}$ denotes angles of the MCP joints with flexing/extending. Assuming no external force is applied on fingers, Equation 1 suggests that in order to bend the DIP joint, the PIP joint must also bend. Similarly, the constraints on thumb joints is described in Equation 1. The range of motion for PIP is very much limited by the MCP joint. The general range of motion constraints for other fingers are enumerated below:

$$-15^\circ \leq \phi_{mcp,aa} \leq 15^\circ,\ 0^\circ \leq \phi_{dip} \leq 90^\circ,\ 0^\circ \leq \phi_{pip} \leq 110^\circ \tag{2}$$

where $\phi_{mcp,aa}$ denotes angles of the MCP joints with abduction/adduction. Compared to flex/extensions, abduction/adduction angles have a smaller range of motion. In addition to these constraints, there are complex interdependencies between finger joint motion patterns that cannot be captured by well-formed equations. The ML models in *mm4Arm* learn such constraints and utilize them for enhancing the accuracy of 3D finger motion tracking.

3 OVERVIEW OF THE EXPERIMENTAL PLATFORM: MMWAVE RADAR AND FMCW

mm4Arm adopts an FMCW radar for tracking forearm vibrations. An FMCW radar works by emitting *chirps*. The *chirp* is reflected back by objects in the environment and based on the time differences between transmission and reception of *chirps* and the doppler shifts, the radar can estimate the *range* (distance) of these objects and velocities.

A *chirp* and the working principle of FMCW radars are visualized in Fig. 5a, which shows a sinusoidal signal with linearly increasing frequency, which is employed by TI *IWR6843ISK* [7] radar, used in *mm4Arm*. Since the transmitted signals are frequency-modulated signals, the reflected

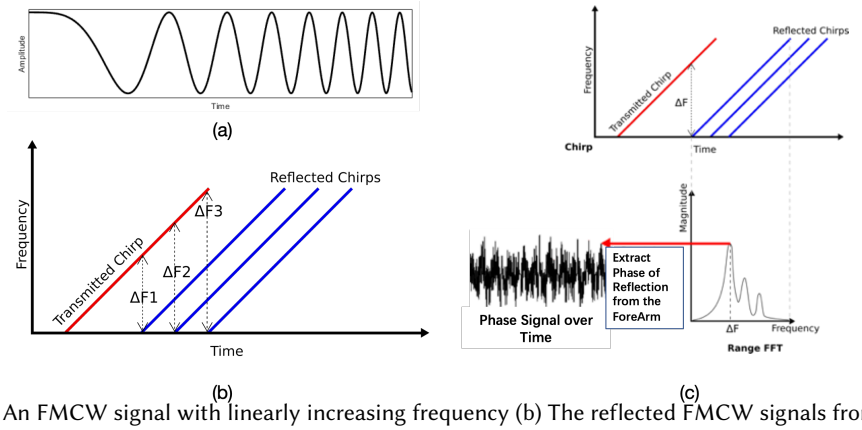


Fig. 5. (a) An FMCW signal with linearly increasing frequency (b) The reflected FMCW signals from objects in the environment (c) A range-FFT will result in multiple *peaks* corresponding to objects in the environment. Tracking the phase of the *peak* from forearm reflections will facilitate finger motion tracking

components will also be frequency-modulated signals. However, because they are delayed, at any given point in time, there is a constant frequency difference between the transmitted and reflected *chirp* as depicted in Fig. 5b. By computing the frequency difference ΔF between the transmitted and received chirps, the distance of the reflecting object can be computed. The below equation precisely converts the frequency difference into the *range* (r) of the object from the radar.

$$r = \frac{\Delta F}{Slope} \tag{3}$$

where *Slope* refers to the rate at which the chirp frequency is linearly modulated.

Depicted in Fig. 5b, multiple reflected chirps from different multipath components can be received at the radar. By performing an FFT operation (called *range FFT*), different multipath components, and their *ranges* can be determined (Fig. 5c). The resolution at which *ranges* can be computed can be expressed as a function of the chirp sweeping bandwidth B as follows [96]:

$$\Delta R = \frac{c}{2B} \tag{4}$$

where c is the speed of light. If the entire working bandwidth of the radar ($3.705GHz$) is effectively swept by a chirp, the above equation predicts a range resolution of $4.05cm$. While this is good for applications like human activity recognition (running, sitting, etc.) where the motion of objects is at larger scales, the resolution is not sufficient for tracking minute micrometer-level vibrations needed for capturing the forearm vibrations during finger motion. Towards capturing a higher resolution range information, *mm4Arm* exploits the *phases*. The *phase* variations can capture minute changes in motion of the reflector, as per the below equation.

$$\Delta\phi = \frac{2\pi\Delta r}{\lambda} \tag{5}$$

Given that the wavelength is in the order of millimeters ($\approx 4mm$), and a typical phase noise of $0.057°$(based on our experimental observations and comparison with the ground truth of the phase error), extremely small changes in range ($\Delta r \approx 0.63\ um$) can be detected from *phase* variations. *mm4Arm* tracks such variations to sense minute vibrations in the forearm. Fig. 5c depicts extraction of continuous phase changes from the radar.

The *range-FFT* will result in multiple *peaks* due to multipath reflections. Among these *peaks*, the *peak* corresponding to reflection from the arm is first isolated (Section 5.1). By measuring the

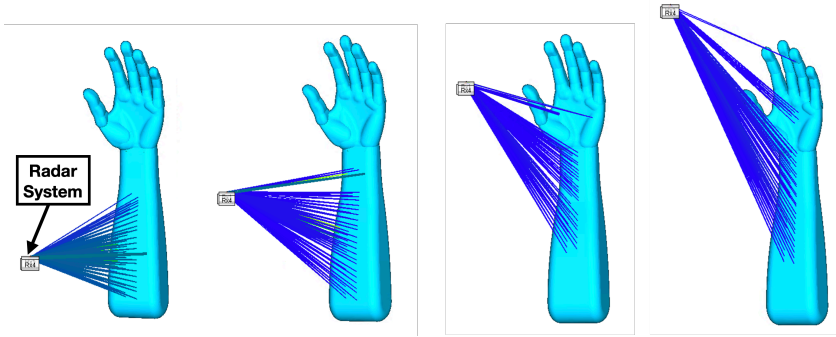


Fig. 6. Simulation results at 60 GHz: The blue lines visualize the rays that are transmitted and returning back to the radar via reflections. We vary the height of the radar to observe all surfaces on the hand that can yield stable reflections back to the radar. Results indicate that the reflections from fingers are negligible even when the radar is placed close to fingers. However, the large surface combined with texture and curvature of the forearm provides stable reflections for 3D finger motion tracking.

phase of this FFT *peak*, and tracking its variations continuously over time and across antennas helps identify rich patterns which are predictive of 3D finger motion.

4 ELECTROMAGNETIC SIMULATIONS OF FEASIBLE REFLECTIONS

We conduct simulations using Remcom WaveFarer toolkit [18] in the 60 GHz spectrum to understand what reflections are feasible for finger motion tracking. WaveFarer uses shoot and bounce ray-tracing technique [71] in addition to techniques based on physical optics [59] (for computing scattered fields), method of equivalent currents [79] (for diffraction effects), and uniform theory of diffraction [62] (for multipath effect between objects). This enables highly accurate simulations [106]. Such techniques have been successfully used in radar sensing applications, such as autonomous driving [17]. Using this platform, we emulate the Texas Instruments *IWR6843ISK* radio [7] at 60 GHz (used by *mm4Arm*) and place it in front of a CAD model of a human arm to obtain a preliminary assessment of feasibility of reflections. The results are elaborated next.

Lack of stable reflections from fingers: Fig. 6 visualizes the simulation results of reflected rays that arrive at the radar. Evidently, very few reflections from the palm and fingers appear at the radar, even when the radar is placed close to the fingers. We observe that this is mainly because of the small size of fingers coupled with the specular nature of the dominant reflections that deflect the rays into random directions. A similar observation on specularity is made in [26, 93], in the context of applications including autonomous cars. Also validated by real experiments in Section 6, the ML models to capture 3D finger motion using such reflections result in very high errors. Therefore, we seek alternative approaches for tracking the motion of fingers.

Stable reflections from the forearm: While the small size of fingers do not provide stable reflections, we observe that there is an opportunity to indirectly track fingers by focusing on forearm reflections. Finger motion activation involves neuro-muscular interactions that trigger minute muscular motions in the forearm, which in turn will induce vibrations in the forearm. Simulation results in Fig. 6 also show that reflections from the forearm can be tracked reliably at the radar owing to its larger surface, texture, and curvature, which can return significant reflections back to the radar. The ability to obtain significant reflections from the forearm allows *mm4Arm* to sense forearm vibrations and hence track finger motion. Note that The mmWave signal doesn't have to penetrate through the fat in the forearm, or have to go inside the human body. We are actually measuring the surface vibration of the forearm caused by muscle activating, and that vibration can cause phase variation of mmWave measurements.

Validation of Simulation Outcome via Real Measurements: In contrast to high-fidelity electromagnetic simulations, the real data does not offer fine enough resolution to visualize the individual reflections from the radar. Therefore, we only provide the end result of 3D hand pose prediction with real data (Fig. 7 shows joint angle errors). Towards this end, we obtain the phase measurements from the mmWave radar, which is a superimposition of phases from individual reflections, and analyze their variation over time in an attempt to capture the rich spatiotemporal relationships to predict the 3D hand pose. We employ a deep learning model for this prediction (detailed in Section 5.2). Specifically, we compare the following three cases: (i) *Finger-only*: Analysis of reflections from fingers-only (forearm blocked by a metal sheet) (ii) *Forearm-only*: Analysis of reflections from forearm-only (fingers blocked by a metal sheet) (iii) *mm4Arm*: Analysis of all reflections from finger and forearm. We compare these three cases with a *naive baseline* in Fig. 7, that always outputs the static hand pose with palm open. We make three observations from Fig. 7. (i) With *Finger-only*, the error is higher and closer to the *naive baseline*, indicating that finger reflections are not sufficient enough to capture 3D hand pose, (ii) *Forearm-only* results in high accuracy which is comparable to vision based approaches (evaluated in Section 7). (iii) This indicates that accuracy with *mm4Arm* as shown in Fig. 7 is mainly due to forearm reflections, and any reflections captured from fingers are too sparse to make any difference in the accuracy. The rest of the paper expands on the details of the deep learning model and associated challenges and provides a thorough performance of quantitative and qualitative results via systematic implementation and measurements.

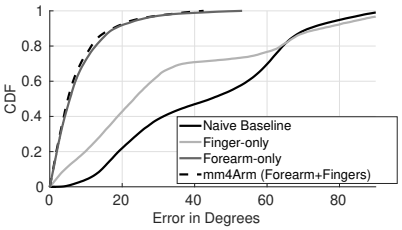


Fig. 7. The reflection from fingers (*Fingers-only*) do not capture sufficient reflections and the accuracy is close to naively predicting an always open palm. On the other hand, forearm reflections (*Forearm-only*) can provide reliable prediction of 3D finger motion. Accordingly, ForeArm reflections mainly contribute to the high accuracy in *mm4Arm*.

5 FROM FOREARM VIBRATIONS TO 3D FINGER JOINT TRACKING

In this section, we describe the following key signal processing and ML modules designed for tracking the 3D finger motion. (i) Isolation of arm reflection from other multipath components (ii) Machine learning model for mapping RF phase data into 3D finger motion pattern by exploiting the spatio-temporal relationships in finger movements. (iii) Domain adaptation techniques for minimizing the training overhead for new users of *mm4Arm*.

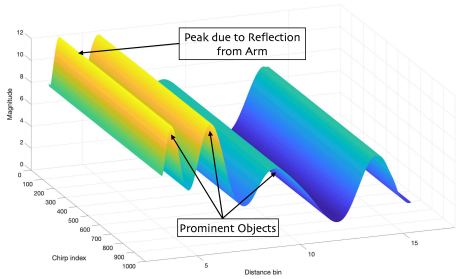


Fig. 8. Tracking of FMCW peaks over time helps eliminate noisy peaks. The phase data corresponding to the peak from the forearm reflection is used for 3D hand pose tracking

5.1 Isolation of Forearm Reflection

As discussed in Section 2, a given *range-FFT* window will include the reflection from the forearm as well as multipath reflections from other objects in the environment. We face two main challenges in isolating the arm reflections: (i) Several noisy peaks show up in the range-bin mainly because of hardware related artifacts. (ii) In addition to the noisy peaks, there will be peaks

corresponding to reflectors in the environment such as walls and furniture. Towards better isolation of signal of interest from the above sources, *mm4Arm* tracks consistent peaks across successive frames. Since the noisy peaks do not consistently appear at a given distance, they are eliminated. This step also eliminates reflections from dynamic multipath such as mobile reflectors. Fig. 8 shows an example where the arm reflection is consistently tracked over time. In addition to arm reflections, there also exists reflections from other objects in the environment. In the real experiment, we are able to eliminate reflections from other environmental reflections even when they are closer to the radar based on the isolation algorithm explained below. The phase of the reflection from the arm would exhibit rapid variations whereas phase from other reflectors will be somewhat monotonous.

Phase Variations of the ForeArm Reflection: The *phase* of the reflection from the arm would exhibit rapid variations whereas phase from other reflectors will be somewhat monotonous. By exploiting this property, *mm4Arm* isolates the reflections from the arm from other multipath components. Fig. 9 depicts an example of phase variation from the arm in comparison with phase variation from a wall reflection. The characteristic and higher level of variations in the arm can be exploited for isolating the arm reflections from other multipath components. By exploiting this property, *mm4Arm* designs a shallow convolutional neural network to first classify reflections into two classes: (i) Reflections due to forearm vibrations (ii) Reflections from other reflectors in the environment. The binary classifier provides high accuracy of 99.4%, thus isolating the forearm reflections from other reflections.

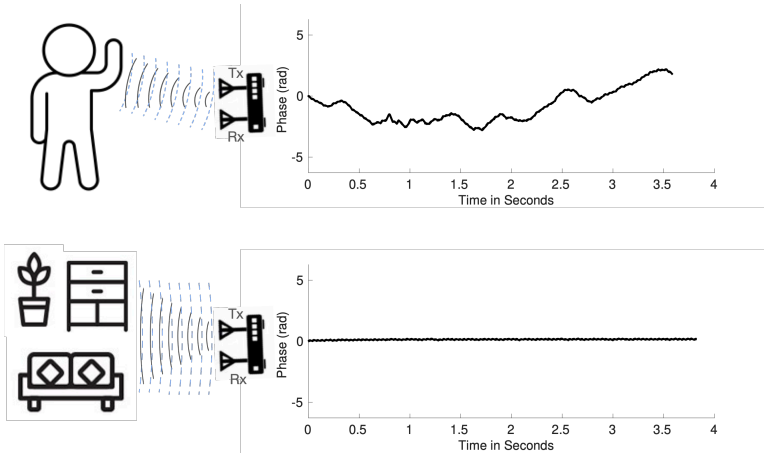


Fig. 9. Phase variation of forearm reflections is more pronounced than the variations from other static objects

5.2 3D Finger Joint Tracking with Encoder Decoder Architecture

We design an encoder-decoder network as illustrated in Fig. 10. The size of convolution filters and the number of filters at each layer are also specified. The network is designed to capture plausible finger pose sequences with spatial constraints across fingers with temporally smooth variations. Instead of looking at one sensor sample at a time, the network captures a holistic view of a large interval of time-series sensor data. This enables the network to enforce and learn the key spatio-temporal constraints, as well as consider historical phase data while making hand pose inferences. The network takes 2*s* of phase data as input and outputs the corresponding 3D hand pose sequence. The various components of the ML model are elaborated next.

(i) Encoder: The encoder-decoder model maps a sequence of input RF phase 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 (ϕ_{dip} for four fingers and ϕ_{ip} for thumb) can be directly computed using Equations 1 since it contains the constraints between the PIP joint and the DIP joints of the 5 fingers of the human hand, which decreases 5 degrees of freedom total as each finger of hand is decreased by 1. Therefore, the actual output of the network is only 16 dimensions. The size of the input is $Y \times T$, which includes phase samples from $Y = 4$ antennas, over $T = 1000$ samples at a sampling rate of 500 Hz (2 seconds). The input first passes through an encoder network that consists of a series of convolutional layers with the input downsized at each layer with maxpool operation. The encoder attempts to capture a compact representation of the input to be used for hand pose extraction. Batch normalization is used at each layer for accelerating convergence by controlling variation in the input distribution at each layer [50].

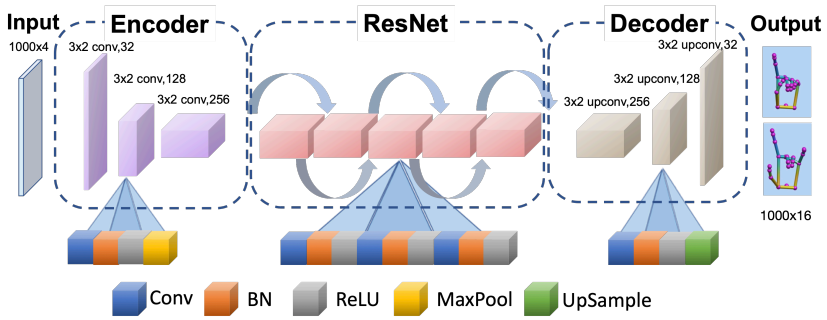


Fig. 10. Encoder Decoder Architecture. BN = Batch Normalization

(ii) Residual Blocks: We introduce residual blocks [48] with skip connections between the encoder and decoder to increase the depth of the network. While the increase in depth allows learning stronger features, the skip connections help achieve fast convergence.

(iii) Decoder: The decoder maps the encoded representations to 3D hand pose. Upsampling layers are introduced so as to incrementally scale the size of output at each layer to eventually match the dimensions of the output. We use nearest-neighbor interpolation technique [47] for performing upsampling. The output size is $D \times T$ where $D = 16$ is the number of joint angles predicted, and $T = 1000$ samples (2 seconds at 500 Hz).

Loss Functions and Optimization: In equations below, $\hat{\phi}$ denotes the prediction by ML models, whereas ϕ denotes training labels from depth camera (leap [8]).

$$L_{mcp,f/e} = \sum_{i=1}^{i=4} (\hat{\phi}_{i,mcp,f/e} - \phi_{i,mcp,f/e})^2 \tag{6}$$

$$L_{pip} = \sum_{i=1}^{i=4} (\hat{\phi}_{i,pip} - \phi_{i,pip})^2 \tag{7}$$

$$L_{mcp,a/a} = \sum_{i=1}^{i=4} (\hat{\phi}_{i,mcp,aa} - \phi_{i,mcp,aa})^2 \tag{8}$$

where $L_{mcp,f/e}$ denotes loss value of MCP joint angles with flex/extensions, L_{pip} denotes loss value of PIP joint angles, and $L_{mcp,a/a}$ is loss value of MCP joint angles with adduction/abduction. The above equations capture the mean squared error (MSE) loss in prediction of joint angles of MCP and PIP joints of the four fingers.

$$L_{thumb} = (\hat{\phi}_{th,mcp,aa} - \phi_{th,mcp,aa})^2 + (\hat{\phi}_{th,mcp,f/e} - \phi_{th,mcp,f/e})^2 \\ +(\hat{\phi}_{th,tm,aa} - \phi_{th,tm,aa})^2 + (\hat{\phi}_{th,tm,f/e} - \phi_{th,mcp,f/e})^2 \tag{9}$$

where L_{thumb} denotes loss value of the thumb. The above equations capture the MSE loss in the MCP and TM joints of the thumb.

$$L_{smoothness} = ||(\nabla\hat{\phi}_t - \nabla\hat{\phi_{t-1}})||_2^2 \tag{10}$$

where $L_{smoothness}$ denotes loss value of the smoothness constraint.The above equation enforces constant velocity smoothness constraint in the predicted joint angles where ϕ_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.

$$L = L_{mcp,f/e} + L_{mcp,aa} + L_{pip} + L_{thumb} + L_{smoothness} \tag{11}$$

Note that the loss function does not include ϕ_{dip} or ϕ_{ip} because we compute them directly from anatomical constraints (Equation 1).

Applying Range of Motion Constraints: The constraints across various finger joints for flex/extensions and abduction/adduction motions were described in Section 2. We apply such constraints to the network in order to facilitate faster learning. Towards this, we first normalize the predicted output of a joint angle by dividing it by the range constraint (for example, by 110° for ϕ_{pip}). We then apply the bounded ReLU activation (bReLU) function [67] to the last activation layer in our network. Bounded ReLU activation(bReLU) is added a upper boundary compared to normal ReLU function:

$$f_{bReLU}(x) = \min\Big(\max(0, x), 1\Big) = \begin{cases} 0 & x \leq 0 \\ x & 0 < x \leq 1 \\ 1 & x > 1 \end{cases} \tag{12}$$

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 10), facilitates predicting anatomically feasible as well as temporally smooth tracking results.

5.3 Decreasing Training Overhead via 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 [63] pretrained on ImageNet database [37] was fine-tuned for classifying images in medical domain[133], remote-sensing [46] and breast-cancer [84]. Similarly, a pre-trained BERT language model [38] was fine-tuned for tasks such as text-summarizing [127], question answering [94], etc. This significantly reduces the burden of training for a new task. In a similar spirit, we use a pretrained model from one user and fine-tune it for a different user to significantly decrease the training overhead without losing much accuracy.

The main steps in the domain adaptation process are as follows: (i) We generate a model for one user (*source*) by extensively training the model with labeled 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 certain layers which are made trainable (elaborated next). Using the few labels from the *target* user, we update the trainable 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. We explore three different approaches for the choice of trainable layers as elaborated next.

Adapting the Batch Normalization Layers: *Finetuning* the BN layers can help 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 such *covariate shift* [50], 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*. Such a strategy has been exploited for image processing applications [65, 66]. 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.

Fine Tuning the Last Layers: Retraining the last layer of the network for a new task, while freezing the pre-trained layers from the rest of the network from another task is a popular approach with applications in image and speech processing [84, 91]. The key intuition is that a network learns meaningful representations through all layers leading upto the last few layers. Thus the initial layers are frozen during the domain adaptation. The last layers are retrained to take the representation computed by the frozen layers and compute the final output. We explore this strategy by only retraining the last layer in Fig. 10 for adapting a pre-trained model from one user for performing inferences on a new user.

Fine Tuning Whole Model: We continue to update the weights of the model pre-trained from another user (without freezing any weights) with limited amounts of training data from the target user. While fine-tuning the whole model might be problematic since the parameter space can be huge, because of high-level similarity in the forearm structure among humans, our experiments suggest that we do not face any issue with convergence. Prior studies also show that fine-tuning the whole network might work for some domains, as has been validated with PatchCamelyon dataset [12] for an image classification problem [54]. With fine-tuning the entire network, our model converges well with improved accuracy with limited amounts of training data. We also note that the accuracy saturates quickly with small amounts of training dataset. Detailed evaluation, and comparison with other strategies on domain adaptation discussed above, is provided in the next section.

6 EXPERIMENTAL SETUP, USER STUDY, AND IMPLEMENTATION

We validate *mm4Arm* based on a systematic user study to analyze the performance across users, distance, multipath environments, joint angles, natural variations in forearm position/orientation, etc. This section details the data collection, size of data for training and testing across settings.

6.1 Data Collection and User Study

The experimental setup is depicted in Fig. 11. We explain the details in this subsection.

Radio Frequency Frontend: *mm4Arm*'s frontend includes Texas Instruments *IWR6843ISK* [7] mmWave radar operating in 60-64 GHz spectrum. Operating with an FMCW bandwidth of $3.705GHz$, we use the DCA1000EVM [14] platform to extract samples at 2 Msps, and obtain reflections from the human arm. The extracted phases are further low pass filtered and down-sampled to a sampling rate of 500 Hz. The phases extracted from the reflections are used for 3D hand pose tracking. Because the technology depends on forearm vibration sensing, it is important for the radar to have

the visibility of the forearm. While the radar does not need to be exactly perpendicular, and it is robust to some variation and arm orientations and height, the accuracy can break down if the forearm is not clearly visible to the radar. However, even with the current setting and a number of applications such as wearable, mobile computing, and AR/VR applications where conventional touch interaction may be hard. The radar beam does not need to be focused, we use off-the-shelf TI IWR6843ISK radar with its natural config, and its field-of-view is +/- 60 degrees in Azimuth and +/- 15 degrees in Elevation[15].

Data Collection Methodology: Our user study protocol has been approved by the IRB committee at our institute. We recruit 10 users (6 males, 4 females) in the age range of 22-47, weight range of 53-94 kgs, and height range of 5.1-6.2 ft. The users face the radar with distances upto $1-5\ ft$ from the radar device as depicted in Fig. 11. We also conduct experiments under non-line-of-sight conditions. For stress-testing *mm4Arm* across all possible 3D finger poses, we follow the guidelines from standard computer vision literature [69]. Accordingly, while the users were allowed to perform arbitrary random finger motions, the study staff also ensured that the users perform all *base states* of possible hand poses as defined in [69]. The majority of possible hand poses are known to be one of such *base states* or transitioning between these poses [110] based on anatomical feasibility constraints. After some practice under the guidance of research staff, the users perform arbitrary finger poses as well as pass through these *base states* in random order. This ensures good convergence of the ML models as well as generalizability. There are no discrete classes of gestures since *mm4Arm* performs tracking in a continuous $\mathbf{R}^{21}$ space.

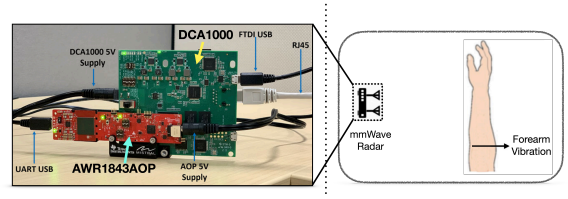
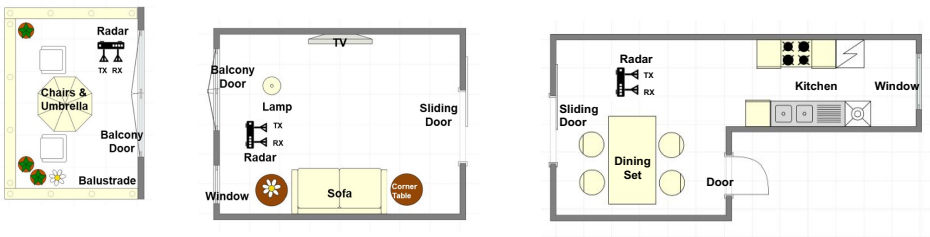


Fig. 11. Experimental setup: Detailed view of *IWR6843ISK* radar and DCA1000EVM board for data collection (Left) [16]. Forearm vibration detection by the radar (Right)

Environment: *mm4Arm* isolates the peak from the forearm reflections (Section 5.1). Thus, the performance is naturally robust to environmental multipath. To better validate this, we conduct the testing under three different environments as shown in Fig. 12 with people moving around in the environment naturally. One-third of the data is collected under each setting with distances varying from 1-5ft. We compare the results across different environments where training and testing data



(a) (b) (c)

Fig. 12. Environments for evaluation of *mm4Arm* (a) Balcony (b) Living Room (c) Kitchen

come from different environments.

Labels for Training and Testing: The collected data includes RF phase data from the mmWave radar and the fingers' 3D coordinates and joint angles captured by leap sensor [8]. While the radar provides RF phase data for 3D pose tracking, the leap sensor data serves as the ground truth for validation and provides labels for training *mm4Arm*'s ML models. The radar and leap data

were synchronized by performing three distinct hand waving patterns at the beginning of each experiment and matching the occurrence of such patterns in the leap and radar phase data. Since *mm4Arm* performs continuous finger tracking instead of discrete gesture classification, we use MSE (instead of cross-entropy) between predicted joint angles (from radar) and ground truth (from leap) as the loss function.

Training Data Collection: As discussed before, the data collection is split uniformly across three environments in Fig. 12. Each user participates in 5 sessions in each environment, with one session each over distances of 1, 2, 3, 4, 5 ft. This results in 15 sessions per user. Each session lasts for 300 seconds, with enough rest between sessions. The user exits the study space after a session and returns to continue with the next session. This enables the model to develop robustness to natural changes in hand position, height, and orientation which can vary across sessions.

Test Data: The above collected data is used for developing three kinds of models as described below. For all cases, the training and testing data is taken from different multipath environments (kitchen, balcony, living room). Other specifics about test cases for each model are also described below. (i) **Model with domain adaptation (*mm4Arm*):** This is the default version of *mm4Arm*, where a model for each user where a pre-trained model from a different user is taken and fine-tuned using techniques in Section 5.3 such that only a small fraction (90 seconds) of user-specific training data is used for developing a model for the user. (ii) **Multi-user model:** This is 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. (iii) **User-dependent model:** As a baseline for comparison, we compare our system *mm4Arm* that requires only 90s of user-specific training data as noted above with a *user dependent model* that requires an excessive training overhead of 1800s of training data per user. This training data comes from 6 sessions of that user (from 1-3ft) from two environments. Testing is done on the third environment. All three combinations of train/test split (across the three environments) are considered for evaluation.

Data size: We believe the training data size is sufficient. *mm4Arm* has much fewer parameters compared to some well-known network architectures, for instance, AlexNet has 61M parameters[63] while our model has only 2M parameters. Our data size is also comparable with regular vision solutions since our input frequency is 500 Hz and we require an excessive training overhead of 1800s of pre-training data per user, while vision based dataset usually have a FPS of 30-60Hz. Moreover, we have also applied skip connections in the residual blocks of our network, helping the model to converge with fewer data samples.

6.2 Implementation

mm4Arm is implemented on a combination of desktop and smartphone devices. The ML model is implemented with TensorFlow [20] 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 [60] 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 [27] on each CONV layer with a parameter of 0.05, and also add dropouts [115] with a parameter of 0.05 following each RELU activation. We apply anatomical constraints to the network in order to facilitate faster learning. Towards this, we first normalize the predicted output of a joint angle by dividing it by the range constraint, then apply the bounded ReLU activation (bReLU) function to the last activation layer in our network. Our system is implemented on a combination of desktop and smartphone devices. Once a model is generated from training, the inference is made entirely on smartphones using TensorFlowLite [43]. We perform the evaluation on OnePlus 9 Pro and Samsung Galaxy S20 smartphones.

7 PERFORMANCE RESULTS

This section provides a systematic evaluation of *mm4Arm* based on insights gathered from the simulations, the experimental platform, and the corresponding data collected as elaborated in the previous section. First, we summarize our findings and later expand on the details.

- The reflections captured directly from fingers are negligible, whereas the prominent reflections generated from the forearm can provide reliable accuracy. The median error in joint angle tracking is 5.73°. The median error in location is 4.07*mm*.
- *mm4Arm* can track fingers under non-line-of-sight conditions and even when the forearm is occluded from wearing clothes. The accuracy is robust to user diversity and natural variation in arm position, height, and orientation. The multipath environment does not impact the accuracy since the forearm reflections are isolated from other multipath components.
- *mm4Arm* can track all finger joints as well as flex and abduction motions reliably.
- The model trained on left hand can be easily transferred for performing inferences on the right hand. This is a key result with applications in the development of prosthetic devices for amputees based on *mirrored bilateral training* (elaborated in Section 9).
- *mm4Arm* runs on smartphones with a latency of ≈ 19ms and low power consumption.

Unless explicitly stated otherwise, the reported results are obtained under the following conditions: (i) The *model with domain adaptation* as described above is used. This is the default version of *mm4Arm*. The user-independent case is separately evaluated under *multi-user models* (Fig. 13). The performance of user-dependent models is shown separately (Fig. 19b). (ii) Results from data collected over a distance of 1-3ft from the radar are included. Other results corresponding to 4ft and 5ft are discussed separately in Fig. 17a. (iii) The errors reported are for flex/extension angles as they are prone to more errors with a high range of motion. Errors for abduction and adduction are discussed separately (Fig. 17d).

To give a brief overview of our evaluation, we have a summary of the measurement results: Fig. 13 depicts the accuracy as a function of different users averaged across all joint angles. Fig. 14a depicts a setting where a user is wearing a long sleeve jacket. Fig. 14b shows the non-line of sight setting where the hand is hidden behind a room divider wall. We also evaluate such robustness in Fig. 16a with users across 9 sessions at distances of 1-3ft over 3 environments. Fig. 16b provides a breakup of accuracy over different heights of the forearm measured relative to the radar. Fig 16c depicts the accuracy breakup over different settings. Fig. 17 depicts the accuracy vs distance, different fingers, finger joints and abduction/adductions and flex/extensions. Fig 19a,b depicts Accuracy comparison of different domain adaptation techniques and size of training data. Based on these results, the factor that may obviously affect the accuracy includes distances between radar and users' forearm and domain adaptation techniques.

Qualitative Results: A short demo is included in this anonymous url [10]. Fig. 2 shows samples of 3D finger motion tracking. We visualize the real hand with the corresponding tracking by the leap sensor (ground truth) and our system *mm4Arm* (mmWave radar). Tracking by *mm4Arm* closely follows the leap ground truth and the real hand. Samples in (f), and (g), indicate that the tracking is consistent even when the fingers are moving. The overall results suggest that *mm4Arm* can track the 3D finger motion pattern with good accuracy.

Overall Accuracy vs Users: Fig. 13 depicts the accuracy as a function of different users averaged across all joint angles. While the multi-user model where the training data is generated from 9 users and tested on an unknown user performs well with a median error of 8.47°, the tail errors and the deviation can be large. Domain adaptation in *mm4Arm* can dramatically cut down the tail errors in addition to improving the median error to 5.73° (location error of 4.07mm), thus leading

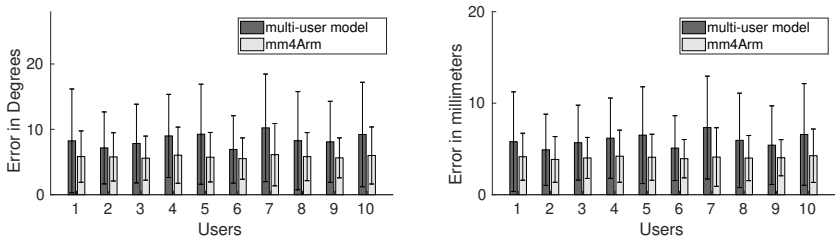


Fig. 13. *mm4Arm* with domain adaptation outperforms *multi-user model* for all users: (a) Error in degrees (b) Error in millimeters

to overall better accuracy, which is comparable to vision based systems [29, 82]. Note that the location error is calculated by averaging the joint location difference of predicted joint location and the ground truth joint location. The ground truth joint locations can be fetched from Leap API[8]. The predicted joint locations can be calculated by predicted joint angles, which is the output of our model, assuming we have the information of users' finger lengths. The accuracy is consistent across users, gender, body sizes, etc.

Non-Line-of-Sight Setting and Clothing: Fig. 14a (last bar) depicts a setting where a user is wearing a long sleeve jacket (Hanes Full-Zip Eco-Smart Hoodie [5]) so that the forearm is not directly visible to the mmWave radar. The accuracy does not affect much because the thickness of the clothing material is typically much smaller to cause any significant attenuation of mmWave signals. Similarly, Fig. 14b shows the non-line of sight setting where the hand is hidden behind a room divider wall (YASRKML 3 Panel Room Divider [19]), with a distance of 3ft between the radar and the hand. The chosen material is similar to typical materials used for partitioning indoor spaces. The median error under this setting is 5.97° whereas the median error under line-of-sight setting at the same distance was 5.73°. This indicates the basic feasibility of sensing under non-line-of-sight conditions. Therefore, the sensing device can be embedded into materials and environments thereby enhancing the ease of deployment.

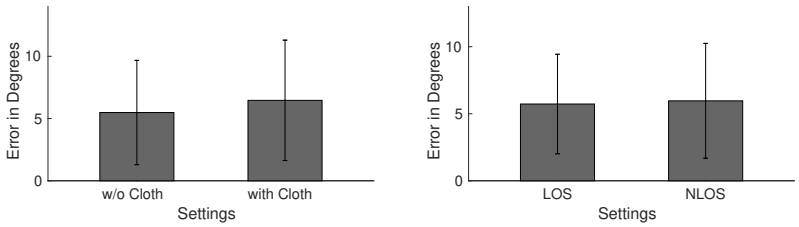
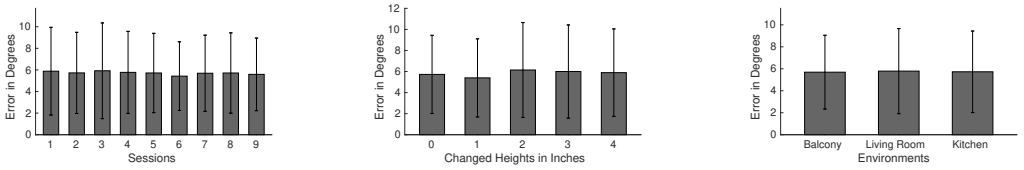


Fig. 14. (a) The accuracy remains stable when users wear long sleeve cloth (b) *mm4Arm* is robust to Non-Line-of-Sight conditions

Performance Analysis over an Application in Gesture Recognition: *mm4Arm* performs 3D tracking of finger motion in a generic context and independent of any application. The tracking results can be used for any application. We evaluate the feasibility of *mm4Arm* over a real-world application in recognition of alphabets in American Sign Language (ASL) as defined in [2]. The classification was performed by comparing the $\mathbf{R}^{21}$ space of joint angles of the users with the joint angles corresponding to each gesture class. The gesture class with the minimum Euclidean distance from the user's finger joint angles is declared as the inferred gesture. Fig. 15 depicts the confusion matrix of the classification. Evidently, most gestures are classified correctly with an overall accuracy



(a) (b) (c)

Fig. 16. (a) Accuracy over sessions with variations in arm position/orientation (b) Accuracy over different heights of the forearm relative to radar (c) Accuracy over environmental settings.

of 92.23%. Gestures such as *A* and *S* are misclassified sometimes because their hand-poses are similar. This demonstrates the feasibility of using *mm4Arm* in real-world applications.

Robustness to Variation in Arm Position, Height, and Orientation: The ML models need to be robust to natural variation in arm position, height, and orientation. We evaluate such robustness in Fig. 16a with users across 9 sessions at distances of 1-3ft over 3 environments in Fig. 12. The domain adaptation data for test sessions come from a different environment (and hence different session) than the training environment. As discussed in the user-study methodology, users exit the study space after each session before coming back to start a new session. This introduces natural variations in arm position, height, and orientation across sessions. Yet, the accuracy is stable across all sessions, thus indicating that *mm4Arm* is robust to the above variations. Furthermore, across these sessions, Fig. 16b provides a breakup of accuracies over different heights of the forearm measured relative to the radar. The height is measured from the wrist joint. When the radar is pointing directly at the wrist joint, the height is 0, and increases as we move downwards from the wrist to elbow. The accuracy is robust to the height of the arm because the training data incorporates such diversity.

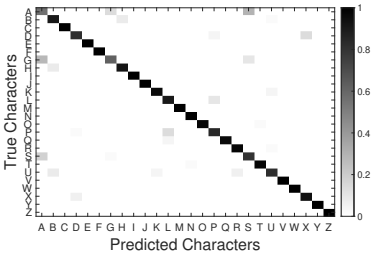
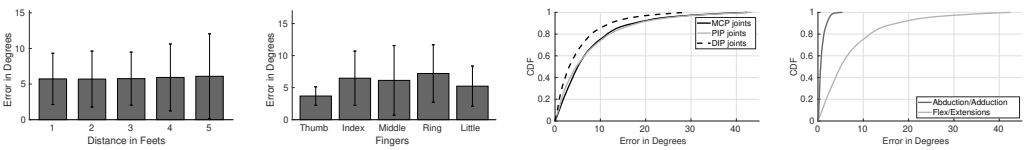


Fig. 15. Confusion Matrix of classification.

Robustness to Environmental Setting: Fig 16c depicts the accuracy breakup over different settings. Since *mm4Arm* eliminates other multipath components before further processing steps (Section 5.1), there would be almost no impact of multipath interference on the accuracy. The accuracy is consistent across different settings.

Accuracy vs Distance: Fig 17a depicts that the accuracy is almost similar for 1-3ft. However, beyond that distance, the accuracy starts to degrade gracefully. While the median errors do not show much degradation the accuracy starts decreasing in the tail. With the increasing distance of the user, the SNR of the forearm reflection decreases which results in higher tail errors.

Accuracy vs Fingers: Fig. 17b depicts the accuracy for the four fingers and thumb. The accuracy is averaged over $\phi_{mcp,f/e}$, ϕ_{pip}, and ϕ_{dip} for the four fingers. For the thumb, the accuracy is computed over $\phi_{mcp,f/e}$, $\phi_{tm,f/e}$, and ϕ_{ip}. The ML models can accurately predict the motion of all fingers. The



(a) (b) (c) (d)

Fig. 17. Accuracy vs (a) Distance (b) Fingers (c) Finger Joints (d) abduction/adductions and flex/extensions

accuracy of thumb is slightly higher because of a smaller range of motion.

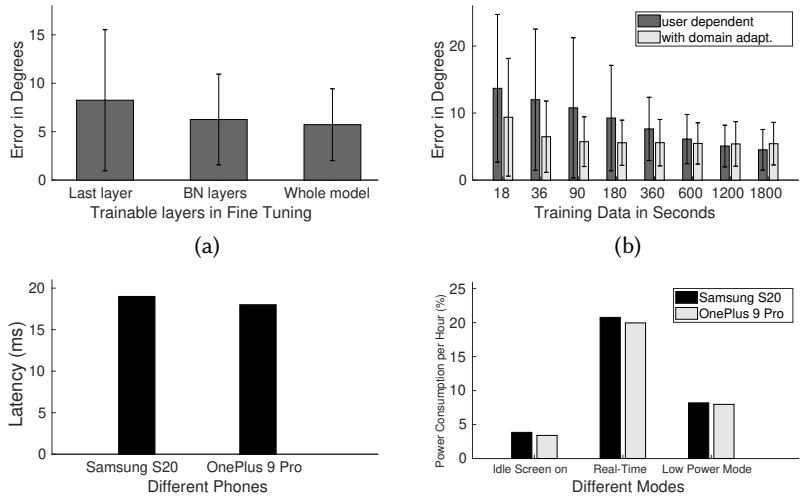


Fig. 19. (a) Accuracy comparison of different domain adaptation techniques. (b) Accuracy vs size of training data. (c) Latency of Execution and (d) Power Consumption on Smartphones

Accuracy vs Finger Joints: Fig. 17c depicts the accuracy as a function of the three finger joints – $\phi_{mcp,f/e}$, ϕ_{pip}, and ϕ_{dip}. *mm4Arm* tracks all finger joints with consistent performance. Fig. 17d depicts the accuracy as a function of flex/extensions and abduction/adductions. Abduction/adduction angles have higher accuracy than flex/extensions because of a limited range of motion.

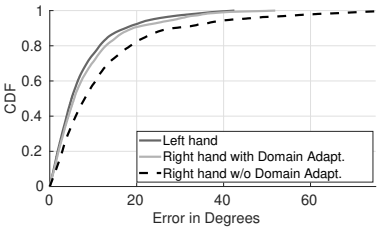


Fig. 18. Transfer of model from left hand to the right hand.

Transferring Model from Left Hand to Right hand: Fig. 18 depicts the accuracy when the model trained for the left hand is transferred for inferences on the right hand. Even without domain adaptation, the direct use of a model trained on the left hand provides good accuracy for inference on the right hand. After domain adaptation with small training data from the right hand (90s), the accuracy is comparable to the left hand. This opens up possibilities of developing ML models for amputees with missing fingers. A model can first be learnt from the hand without amputation, which can then be transferred to the hand with amputation.

Comparison of Domain Adaptation Strategies: Fig. 19a depicts the comparison of three domain adaptation strategies discussed in Section 5.3. Because of the high-level similarity in forearm-vibration pattern across users, it turns out that fine-tuning the whole model is feasible and achieves the best accuracy with stable convergence even with limited domain adaptation data. Therefore, *mm4Arm* adopts a strategy that updates the whole network during domain adaptation.

Accuracy vs Size of Training Data: Fig. 19b depicts the accuracy variation with the size of training data for *mm4Arm* with domain adaptation in comparison to a baseline of the *user dependent model*. With only 5% (90 s) of training data as the *user dependent model*, the *mm4Arm* with domain adaptation achieves a performance close to the *user dependent model*. Thus, *mm4Arm* can adapt to a practical setting with limited training data.

Comparison with Vision: To give an idea of how *mm4Arm* performs compared to some other solutions, we compare *mm4Arm* with SOTA camera based techniques – *Vision1* [29] and *Vision2* [82] – as shown in Fig. 20. We note that *mm4Arm*'s accuracy is comparable to camera-based approaches

Table 1. Scope of *mm4Arm* in the context of key prior works. To our best knowledge, *mm4Arm* is the first work that performs 3D hand pose tracking with 21 degrees of freedom using RF signals with benefits as highlighted in the table.

System	Sensing Band	Robustness to Lighting and Ambience	Non Line of Sight	21 DoF Tracking
Google Soli [117]	mmWave (60 GHz)	✓	✓	✗
mmASL [102]	mmWave (60 GHz)	✓	✓	✗
RFWash [55]	mmWave (60 GHz)	✓	✓	✗
SignFi [77]	WiFi (5 GHz)	✓	✓	✗
WiSee [92]	WiFi (5 GHz)	✓	✓	✗
FingerIO [83]	Ultrasound (18-20 kHz)	✓	✓	✗
LLAP [118]	Ultrasound (48 kHz)	✓	✓	✗
GANerated [81]	Visible Light	✗	✗	✓
MediaPipe [126]	Visible Light	✗	✗	✓
Leap [8]	Visible Light and Infrared	✗	✗	✓
***mm4Arm* (This paper)**	**mmWave (60 GHz)**	✓	✓	✓

while offering other benefits over cameras such as not being privacy-invasive, agnostic to lighting conditions, and the ability to work under basic occlusions, thus allowing the *mm4Arm* system to be embedded in everyday devices and environments.

System Profiling: Latency, Power Consumption, and Processing Overhead: Fig. 19c depicts the latency of *mm4Arm*'s ML models on modern smartphones - Samsung S20 and OnePlus 9 Pro – with TensorFlowLite. The latency figures denote the overall time spent in processing 2 seconds of input sensor data using the encoder-decoder architecture. The latency is under 20ms on both smartphones which indicates low processing overhead. We use Batterystats and Battery Historian[13] tools for profiling the energy of the TensorflowLite model. We compare the difference in power between the following two states. (i) The device is idle with screen on. (ii) The device is making inferences using TensorflowLite model. Depicted in Fig. 19d, the idle display-screen on discharge rate is 3.85%, 3.40% per hour for two phones. The discharge rates while executing the ML models is also summarized in the figure. Since the encoder-decoder processes data in chunks of 2s, it will incur a delay of atleast 2s if we process the data only once in 2s. Processing the model once in 2s will result in a discharge rate of 8.19%, 7.97% per hour for the two phones. Towards making it real-time, we make a modification where at any given instant of time, previous 2s 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. 19d, this entails continuous/redundant processing thus increasing the discharge rate to $\approx$ 20.77%, 19.96% per hour for the two phones. The low-power mode trades off real-time performance (2s delay) for power savings. Depending on requirements of real-time latency or energy efficiency, a user can choose between the two modes.

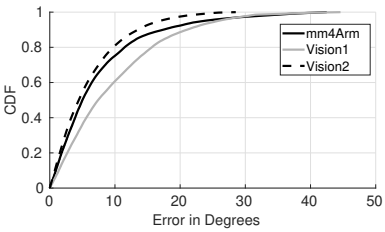


Fig. 20. Comparison with Vision

8 RELATED WORK

Table 1 provides a brief overview of *mm4Arm* in the context of key prior work. In the table, *21 DoF Tracking* refers to the ability to track all finger joints which have a total of 21 degrees of freedom

(DoF) as elaborated in Section 2.2. The related work falls under three categories as elaborated below. We compare and contrast the need for *mm4Arm* with respect to each of these areas.

Vision: Depth cameras like kinect[9] and leap [8] sensors can track fingers. They have revolutionized the gaming industry by gesture interfaces. Recently, even monocular RGB cameras are able to capture 3D motion of fingers by exploiting advances in ML together with the availability of large-scale training data [28, 51, 82]. While such works are truly transformative in nature, vision-based approaches can be privacy-invasive and susceptible to changes in lighting, background, and resolution. Digits [57] uses wrist-mounted infrared cameras for 3D finger pose tracking. Similarly, DorsalNet [120] uses wrist-mounted visual cameras for 3D finger motion tracking. FingerTrak [49] has innovatively designed wearable thermal cameras to track 3D finger motion but has issues with background temperature stability and the shifting of the camera on the hand as noted by the authors. In contrast to approaches based on external cameras, or wearable cameras, we believe *mm4Arm*'s approach provides a solution that is completely passive with robustness to lighting, resolution, and background conditions. Furthermore, *mm4Arm* can track through materials and non-line-of-sight conditions, allowing the system to be embedded into devices and environments

Radio Frequency Reflections: RF signals have been used for human body motion sensing [21, 105, 123]. They are also used to track the motion of the hand and classify discrete gestures by using a combination of wireless channel state information (CSI), and Doppler shifts [64, 78, 103]. mmWrite [98] performs handwriting recognition using mmWave radars. RFWash [55] detects hand wash hygiene using mmWave radars near bathroom mirrors. SignFi [77] uses CSI from WiFi APs for sign language recognition. ExASL [101] tracks point clouds computed from range-doppler spectrum and angle of arrival spectrum of mmWave radars. This is used to classify upto 23 discrete gestures used in ASL. Google Soli [117] and works in [99, 121, 129] uses reflections from mmWave signals to track up to 11 finger motion gestures. *mm4Arm* differs from above works in two ways: (i) In contrast to gesture and activity classification where the search space is limited to 10 to 50 predefined discrete classes, *mm4Arm*'s search space is a continuous space of 3D finger motion with 21 *degrees of freedom* (DoF). The 3D finger locations predicted by *mm4Arm* can serve as inputs to any gesture classification problem – independent of a specific application. (ii) *mm4Arm* senses vibrations in the forearm for 3D finger motion tracking, which results in robust tracking in comparison to reflection from fingertips that are not stable (details in Section 4).

Wearable Sensing: Sensor embedded gloves that use a combination of sensors like IMU, flex, capacitive, pressure, etc are popular [1, 3, 22, 34, 68]. However, wearing gloves precludes the user from performing natural and dexterous activities with fine precision as studied in recent works[100]. Localization and human body tracking projects exploit IMU, and WiFi sensors [21, 33, 116, 122, 130]. Similarly, IMU, WiFi, and acoustic signals have also been used for hand gesture recognition [88, 104, 113, 132]. FingerIO [83], FingerPing [125] use acoustic signals for finger gesture detection. uWave[73] uses accelerometers for user authentication and interaction with a mobile device. Tomo [128] uses electrical impedance tomography with 8 electrodes on the arm for performing classification of 8 gestures. Interferri [52] uses acoustic transducers for classification of 11 hand gestures Capacitative sensing has been systematically investigated by Capband [112] for recognition of 15 hand gestures. ElectroRing [56] attaches electrodes on the index finger and IMU sensors for detecting six different pinch-like finger gestures. DeepASL [41] uses wearable camera for ASL translation of sentences with 56 commonly occurring ASL words. ThumbTrak [109] detects 12 finger gestures by measuring relative distance between thumb and other fingers using proximity sensors. ZeroNet [76] extracts training data from videos to classify 50 hand gestures. In contrast to gesture and activity classification, as noted earlier, *mm4Arm* performs continuous 3D finger motion tracking. AuraRing [89], tracks the index finger precisely using a magnetic wristband and ring on index finger. In contrast to AuraRing, *mm4Arm* tracks all fingers. With a combination of deep

learning techniques based on CNN, RNN, etc, prior works on EMG sensing perform classification of discrete hand poses [36, 40, 95, 97, 108] or track a predefined sequence of hand poses [95, 108]. The Myo armband has been used for 3D finger motion tracking [74, 75]. However, EMG sensors need calibration and warming of the skin to be in proper contact with the electrodes which can even take up to 5 minutes during each instance of wearing, leading to usability issues [11, 111, 119]. In contrast to using wearable devices, *mm4Arm*'s RF-based sensing is passive, since the user does not need to wear any sensor on the body.

9 APPLICATIONS, LIMITATIONS, AND FUTURE WORK

■ **Prosthetic Devices:** A key benefit of *mm4Arm* lies in the ability to sense finger motion directly from forearm vibrations instead of sensing from the fingers. Prior research has shown that subjects with amputation in the hand will still retain forearm muscular activity [35, 42, 85, 87], which manifests into forearm vibrations. Therefore, we plan to exploit the findings in this paper for the development of prosthetic devices for amputees by detecting forearm vibrations. However, because of missing fingers, it is non-trivial to generate training data that map phase patterns into corresponding 3D joint angles of various fingers. Towards handling this challenge, we plan to explore a *mirrored bilateral training* [85] scheme. At a high level, the forearm muscular activity (and the corresponding vibrations) are known to be similar in both hands for performing similar finger motion activities [114]. Therefore, an ML 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). The results in Fig. 18 shows the basic feasibility of such an approach, since the model trained on one hand can be used for performing inferences on the other hand. However, we leave a thorough investigation for future work.

■ **Touchless Interaction for IoT applications:** We believe *mm4Arm* can enable a number of touchless user interfaces such as typing on a virtual keyboard or gesture-based user interfaces. This is particularly useful for interaction with devices with small form factors such as a smartwatch, miniature IoT devices, game controls, robotic home assistants, mobile spectroscopy, etc. Security-based applications can be enabled where a user can lock and unlock an IoT device with a signature based on 3D finger motion pattern.

■ **Smart Assistants for Deaf People:** We envision a future application in accessibility. Voice assistants like Amazon Alexa and Google Home are popular, but inaccessible to the deaf community. The population of the deaf community is upward of 10 million in the US and about 466 million globally [6, 80, 86]. In this context, we believe touchless and fluid interaction enabled by *mm4Arm* with 21 DoF 3D finger motion can enable deaf people to interact with voice assistants by issuing complex commands like a natural language without being limited to a set of predefined gestures.

■ **Robotic Teleoperation:** Complex and unstructured robotic operation, especially in an unregulated environment may require human intelligence in addition to mechanical sturdiness and robustness of a robot. This might include applications ranging from controlling a home assistant robotic avatars or a robotic avatar in a dangerous industrial setting [39] in tasks including grasping and manipulating objects in complex ways. Towards this end, we believe 21 DoF finger motion tracking in *mm4Arm* can provide a solution for robotic avatar control, which is particularly useful if the control is desired from anywhere, anytime.

■ **Tracking Multiple Users Simultaneously:** While this paper focuses on tracking a single user, in principle, the algorithms presented could track concurrent changes from multiple forearms as long as they occur at different distances from the radar. The reflections from different users will fall in different range bins after the range-FFT (Section 3, Fig. 5). These reflections can be isolated

from other multipath reflections (Section 5.1), and the phase variations of all users can be analyzed for tracking finger motion. We will conduct more studies in the future to validate this approach.

■ **Alternative ways of mapping forearm vibrations to finger motion:** We considered designing heuristics to map forearm vibrations to finger motion. To the best of our knowledge, there is no closed form mapping between muscles, forearm vibrations and finger motions, and we believe the neural network could learn the complex mapping above, for example, WR-Hand [74] tracks 3D hand poses by inference EMG data to neural network. We plan to explore alternative methods of doing this in the future.

■ **Thick obstacle object in NLoS experiments:** We did our non-line-of-sight(NLoS) experiments using a dividing wall that is commonly used in office settings (YASRKML 3 Panel Room Divider[19]), which is thinner than typical walls used to separate rooms. We plan to explore the experiment in separate rooms with a thick wall in between in the future.

■ **Potential weighting in loss functions:** Regarding the Loss function(Equation 11), We are inspired by prior computer vision works[82], who have a similar loss equation as ours. However, different weightings in the loss function might potentially optimize *mm4Arm*'s accuracy, and we will explore it in the future.

■ **Size of Sensing Device:** The current experimental setup is bulky. However, we note that the actual mmWave chip is only 2cm×2cm in size, and the dimensions of the antenna is 2.5cm×3cm. This can be integrated into a compact PCB with a SoC microcontroller to stream the range-FFT results from the radar to a smartphone. The development board used in *mm4Arm* is only for the 'prototypying phase' as this is the standard procedure in many IoT applications to extensively test the prototype before rolling out on a compact PCB [24]. Our future work will include testing the feasibility of such a fabrication to create a smaller sensing device.

10 CONCLUSION

Because of the ability to sense the sense environment around us, mmWave signals are being increasingly considered for sensing applications in addition to high-speed networking. Through a combination of high-fidelity electromagnetic simulations and real-world measurements, this paper shows the feasibility of sensing forearm vibrations for 3D finger motion tracking using mmWave signals. Anatomical constraints of finger motions were fused with ML advances in encoder-decoder, Resnets, and domain adaptation in achieving reliable accuracy with low training overhead. The inference is done with low processing and energy overhead on smartphones. Despite progress, we believe we have only scratched the surface. Opportunities exist for developing IoT applications on top of *mm4Arm* in the areas of touchless user interfaces, accessibility, prosthetic devices, etc.

11 ACKNOWLEDGMENTS

We sincerely thank the reviewers and our Shepherd, Prof. Shaileshh Bojja Venkatakrishnan, for their constrcutive comments and feedback. This work is partially supported by the NSF grants NSF-2046972, NSF-2008384 and NSF-1956276.

REFERENCES

[1] 5dt data glove ultra - 5dt. https://5dt.com/5dt-data-glove-ultra/.
[2] American manual alphabet. https://en.wikipedia.org/wiki/American_manual_alphabet.
[3] Cyberglove systems llc. http://www.cyberglovesystems.com/.
[4] Forearm muscles : Attachment, nerve supply action. https://anatomyinfo.com/forearm-muscles/.
[5] Hanes men's full-zip eco-smart hoodie. https://www.amazon.com/Hanes-EcoSmart-Fleece-Hoodie-Black/dp/B00JUM4CT4/.

[6] How many deaf people are there in united states. https://research.gallaudet.edu/Demographics/deaf-US.php.
[7] Iwr6843isk. https://www.ti.com/tool/IWR6843ISK.
[8] Leap motion developer. https://developer.leapmotion.com/.
[9] Microsoft kinect2.0. https://developer.microsoft.com/en-us/windows/kinect.
[10] mm4arm demo video. https://www.dropbox.com/s/j14oh4udhaxqa05/mm4Arm_demo.mp4?dl=0.
[11] Myo official tutorial. https://support.getmyo.com/hc/en-us/articles/203910089-Warm-up-while-wearing-your-Myo-armband.
[12] patch camelyon tensorflow datasets. https://www.tensorflow.org/datasets/catalog/patch_camelyon.
[13] Profile battery usage with batterystats and battery historian. https://developer.android.com/topic/performance/power/setup-battery-historian.
[14] Real-time data-capture adapter for radar sensing evaluation module. https://www.ti.com/tool/DCA1000EVM.
[15] Ti mmwave radar. https://dev.ti.com/tirex/explore/node.
[16] tiuserguide. https://www.ti.com/lit/ug/spruix8/spruix8.pdf?ts=1631234356918.
[17] Using wavefarer automotive radar simulation software and chirp doppler to assess radar performance for drive scenario. https://resources.remcom.com/automotive-radar/publications-wavefarer-chirp-doppler-for-drive-scenarios-comcas2019.
[18] Wavefarer radar simulation software. https://www.remcom.com/wavefarer-automotive-radar-software.
[19] Yasrkml 3 panel room divider. https://www.amazon.com/YASRKML-Partition-Separators-Freestanding-102x71-3/dp/B092HQ5W7D/.
[20] Abadi, M., et al. Tensorflow: A system for large-scale machine learning. In *OSDI* (2016).
[21] Adib, F., Hsu, C.-Y., Mao, H., Katabi, D., and Durand, F. Capturing the human figure through a wall. *ACM Transactions on Graphics (TOG) 34*, 6 (2015), 1–13.
[22] 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.
[23] An, S., and Ogras, U. Y. Mars: mmwave-based assistive rehabilitation system for smart healthcare. *ACM Transactions on Embedded Computing Systems (TECS) 20*, 5s (2021), 1–22.
[24] Ashwini, A. Everything you need to know about iot prototyping, Mar 2020. https://medium.com/swlh/everything-you-need-to-know-about-iot-prototyping-e4ad2739bc6a.
[25] Bansal, K., Rungta, K., Zhu, S., and Bharadia, D. Pointillism: Accurate 3d bounding box estimation with multi-radars. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems* (2020), pp. 340–353.
[26] Bansal, K., Rungta, K., Zhu, S., and Bharadia, D. Pointillism: accurate 3d bounding box estimation with multi-radars. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems* (2020), pp. 340–353.
[27] 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.
[28] Cai, Y., Ge, L., Cai, J., and Yuan, J. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *ECCV* (2018).
[29] Cao, Z., Radosavovic, I., Kanazawa, A., and Malik, J. Reconstructing hand-object interactions in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 12417–12426.
[30] Chang, Z., Zhang, F., Xiong, J., Ma, J., Jin, B., and Zhang, D. Sensor-free soil moisture sensing using lora signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 6*, 2 (2022), 1–27.
[31] Chen, A. T.-Y., Biglari-Abhari, M., and Wang, K. I.-K. Context is king: Privacy perceptions of camera-based surveillance. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (2018), pp. 1–6.
[32] Chen Chen, F., et al. Constraint study for a hand exoskeleton: human hand kinematics and dynamics. *Journal of Robotics 2013* (2013).
[33] Chintalapudi, K., et al. Indoor localization without the pain. In *ACM MobiCom* (2010).
[34] Connolly, J., et al. Imu sensor-based electronic goniometric glove for clinical finger movement analysis. *IEEE Sensors Journal* (2017).
[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] Deng, J., et al. Imagenet: A large-scale hierarchical image database. In *IEEE CVPR* (2009).
[38] Devlin, J., et al. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[39] Du, G., Zhang, P., Mai, J., and Li, Z. Markerless kinect-based hand tracking for robot teleoperation. *International Journal of Advanced Robotic Systems 9*, 2 (2012), 36.

[40] Du, Y., et al. Semi-supervised learning for surface emg-based gesture recognition. In *IJCAI* (2017).

[41] Fang, B., Co, J., and Zhang, M. Deepasl: Enabling ubiquitous and non-intrusive word and sentence-level sign language translation. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems* (2017), pp. 1–13.

[42] 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).

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

[44] Ha, U., Leng, J., Khaddaj, A., and Adib, F. Food and liquid sensing in practical environments using {RFIDs}. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)* (2020), pp. 1083–1100.

[45] Ha, U., Madani, S., and Adib, F. Wistress: Contactless stress monitoring using wireless signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 5*, 3 (2021), 1–37.

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

[51] Iqbal, U., et al. Hand pose estimation via latent 2.5 d heatmap regression. In *ECCV* (2018).

[52] Iravantchi, Y., Zhang, Y., Bernitsas, E., Goel, M., and Harrison, C. Interferi: Gesture sensing using on-body acoustic interferometry. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019), pp. 1–13.

[53] Jiang, C., Guo, J., He, Y., Jin, M., Li, S., and Liu, Y. mmvib: micrometer-level vibration measurement with mmwave radar. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking* (2020), pp. 1–13.

[54] Kandel, I., and Castelli, M. How deeply to fine-tune a convolutional neural network: a case study using a histopathology dataset. *Applied Sciences 10*, 10 (2020), 3359.

[55] Khamis, A., Kusy, B., Chou, C. T., McLaws, M.-L., and Hu, W. Rfwash: a weakly supervised tracking of hand hygiene technique. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems* (2020), pp. 572–584.

[56] 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.

[57] Kim, D., et al. Digits: freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In *ACM UIST* (2012).

[58] Kim, D., Hilliges, O., Izadi, S., Butler, A. D., Chen, J., Oikonomidis, I., and Olivier, P. Digits: freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology* (2012), pp. 167–176.

[59] Kim, J. H., Chun, H. J., Hong, I. P., Kim, Y. J., and Park, Y. B. Analysis of fss radomes based on physical optics method and ray tracing technique. *IEEE Antennas and Wireless Propagation Letters 13* (2014), 868–871.

[60] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[61] Kong, H., Xu, X., Yu, J., Chen, Q., Ma, C., Chen, Y., Chen, Y.-C., and Kong, L. m3track: mmwave-based multi-user 3d posture tracking. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services* (2022), pp. 491–503.

[62] Kouyoumjian, R. G., and Pathak, P. H. A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface. *Proceedings of the IEEE 62*, 11 (1974), 1448–1461.

[63] Krizhevsky, A., et al. Imagenet classification with deep convolutional neural networks. In *NIPS* (2012).

[64] 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.

[65] Li, Y., Wang, N., Shi, J., Hou, X., and Liu, J. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition 80* (2018), 109–117.

[66] Li, Y., Wang, N., Shi, J., Liu, J., and Hou, X. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779* (2016).

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

[68] Lin, B.-S., et al. Design of an inertial-sensor-based data glove for hand function evaluation. *Sensors* (2018).

[69] Lin, J., and Wu, T. S. H. Modeling the constraints of human hand motion.

[70] 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.

[71] Ling, H., Chou, R.-C., and Lee, S.-W. Shooting and bouncing rays: Calculating the rcs of an arbitrarily shaped cavity. *IEEE Transactions on Antennas and propagation 37*, 2 (1989), 194–205.

[72] Liu, H., Wang, Y., Zhou, A., He, H., Wang, W., Wang, K., Pan, P., Lu, Y., Liu, L., and Ma, H. Real-time arm gesture recognition in smart home scenarios via millimeter wave sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 4*, 4 (2020), 1–28.

[73] Liu, J., et al. uwave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing* (2009).

[74] Liu, Y., Lin, C., and Li, Z. Wr-hand: Wearable armband can track user's hand. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 5*, 3 (2021), 1–27.

[75] Liu, Y., Zhang, S., and Gowda, M. Neuropose: 3d hand pose tracking using emg wearables. In *Proceedings of the Web Conference 2021* (2021), pp. 1471–1482.

[76] 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.

[77] 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.

[78] 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.

[79] Michaeli, A. Equivalent edge currents for arbitrary aspects of observation. *IEEE Transactions on Antennas and Propagation 32*, 3 (1984), 252–258.

[80] Mitchell, R. E. How many deaf people are there in the united states? estimates from the survey of income and program participation. *Journal of deaf studies and deaf education 11*, 1 (2005), 112–119.

[81] Mueller, F., Bernard, F., Sotnychenko, O., Mehta, D., Sridhar, S., Casas, D., and Theobalt, C. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 49–59.

[82] Mueller, F., et al. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *IEEE CVPR* (2018).

[83] Nandakumar, R., et al. Fingerio: Using active sonar for fine-grained finger tracking. In *ACM CHI* (2016).

[84] Nawaz, W., et al. Classification of breast cancer histology images using alexnet. In *International conference image analysis and recognition* (2018), Springer.

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

[86] Organization, W. H. Deafness and hearing loss. https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss.

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

[88] Parate, A., et al. Risq: Recognizing smoking gestures with inertial sensors on a wristband. In *ACM MobiSys* (2014).

[89] Parizi, F. S., Whitmire, E., and Patel, S. Auraring: Precise electromagnetic finger tracking. *ACM IMWUT* (2019).

[90] Peña Pitarch, E. *Virtual human hand: Grasping strategy and simulation.* Universitat Politècnica de Catalunya, 2008.

[91] Peng, Y., Yan, S., and Lu, Z. Transfer learning in biomedical natural language processing: an evaluation of bert and elmo on ten benchmarking datasets. *arXiv preprint arXiv:1906.05474* (2019).

[92] 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.

[93] Qian, K., He, Z., and Zhang, X. 3d point cloud generation with millimeter-wave radar. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 4*, 4 (2020), 1–23.

[94] 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).

[95] 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.

[96] Rao, S. Introduction to mmwave sensing: Fmcw radars. *Texas Instruments (TI) mmWave Training Series* (2017).

[97] Raurale, S., et al. Emg acquisition and hand pose classification for bionic hands from randomly-placed sensors. In *IEEE ICASSP* (2018).

[98] Regani, S. D., Wu, C., Wang, B., Wu, M., and Liu, K. R. mmwrite: Passive handwriting tracking using a single millimeter wave radio. *IEEE Internet of Things Journal* (2021).

[99] Ren, Y., Lu, J., Beletchi, A., Huang, Y., Karmanov, I., Fontijne, D., Patel, C., and Xu, H. Hand gesture recognition using 802.11 ad mmwave sensor in the mobile device. In *2021 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)* (2021), IEEE, pp. 1–6.

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

[101] 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.

[102] Santhalingam, P. S., Hosain, A. A., Zhang, D., Pathak, P., Rangwala, H., and Kushalnagar, R. mmasl: Environment-independent asl gesture recognition using 60 ghz millimeter-wave signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 4*, 1 (2020), 1–30.

[103] 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.

[104] Sherman, M., et al. User-generated free-form gestures for authentication: Security and memorability. In *ACM MobiSys* (2014).

[105] Shi, C., Lu, L., Liu, J., Wang, Y., Chen, Y., and Yu, J. mpose: Environment-and subject-agnostic 3d skeleton posture reconstruction leveraging a single mmwave device. *Smart Health* (2021), 100228.

[106] Skidmore, G., Chawla, T., and Bedrosian, G. Combining physical optics and method of equivalent currents to create unique near-field propagation and scattering technique for automotive radar applications. In *2019 IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS)* (2019), IEEE, pp. 1–6.

[107] Song, J., Sörös, G., Pece, F., Fanello, S. R., Izadi, S., Keskin, C., and Hilliges, O. In-air gestures around unmodified mobile devices. In *Proceedings of the 27th annual ACM symposium on User interface software and technology* (2014), pp. 319–329.

[108] 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.

[109] 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).

[110] Tomasi, C., Petrov, S., and Sastry, A. 3d tracking= classification+ interpolation. In *ICCV* (2003), vol. 3, p. 1441.

[111] Torres, T. Myo gesture control armband review. https://www.pcmag.com/reviews/myo-gesture-control-armband, 2015.

[112] Truong, H., et al. Capband: Battery-free successive capacitance sensing wristband for hand gesture recognition. In *ACM SenSys* (2018).

[113] Tung, Y.-C., and Shin, K. G. Echotag: Accurate infrastructure-free indoor location tagging with smartphones. In *ACM MobiCom* (2015).

[114] 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).

[115] Wager, S., Wang, S., and Liang, P. S. Dropout training as adaptive regularization. In *Advances in neural information processing systems* (2013), pp. 351–359.

[116] Wang, J., et al. Ubiquitous keyboard for mobile devices: harnessing multipath fading for fine-grained keystroke localization. In *ACM MobiCom* (2014).

[117] 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.

[118] Wang, W., Liu, A. X., and Sun, K. Device-free gesture tracking using acoustic signals. Association for Computing Machinery.

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

[120] 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.

[121] Xia, Z., Luomei, Y., Zhou, C., and Xu, F. Multidimensional feature representation and learning for robust hand-gesture recognition on commercial millimeter-wave radar. *IEEE Transactions on Geoscience and Remote Sensing 59*, 6 (2020), 4749–4764.

[122] Xiong, J., and Jamieson, K. Arraytrack: A fine-grained indoor location system. In *USENIX NSDI* (2013).

[123] Xue, H., Ju, Y., Miao, C., Wang, Y., Wang, S., Zhang, A., and Su, L. mmmesh: Towards 3d real-time dynamic human mesh construction using millimeter-wave. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services* (2021), pp. 269–282.

[124] Zhai, S., Milgram, P., and Buxton, W. The influence of muscle groups on performance of multiple degree-of-freedom input. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1996), pp. 308–315.
[125] Zhang, C., et al. Fingerping: Recognizing fine-grained hand poses using active acoustic on-body sensing. In *ACM CHI* (2018).
[126] Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.-L., and Grundmann, M. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214* (2020).
[127] Zhang, H., Xu, J., and Wang, J. Pretraining-based natural language generation for text summarization. *arXiv preprint arXiv:1902.09243* (2019).
[128] Zhang, Y., and Harrison, C. Tomo: Wearable, low-cost electrical impedance tomography for hand gesture recognition. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (2015), pp. 167–173.
[129] Zhang, Z., Tian, Z., Zhou, M., Nie, W., and Li, Z. Riddle: Real-time interacting with hand description via millimeter-wave sensor. In *2018 IEEE International Conference on Communications (ICC)* (2018), IEEE, pp. 1–6.
[130] Zhao, M., et al. Through-wall human mesh recovery using radio signals. In *IEEE CVPR* (2019).
[131] Zhao, Y., Sark, V., Krstic, M., and Grass, E. Novel approach for gesture recognition using mmwave fmcw radar. In *2022 IEEE 95th Vehicular Technology Conference:(VTC2022-Spring)* (2022), IEEE, pp. 1–6.
[132] Zhou, P., et al. Use it free: Instantly knowing your phone attitude. In *ACM MobiCom* (2014).
[133] Zhou, Z., et al. Fine-tuning convolutional neural networks for biomedical image analysis: actively and incrementally. In *IEEE CVPR* (2017).

Received August 2022; revised October 2022; accepted November 2022