Towards Smartphone-based 3D Hand Pose Reconstruction Using Acoustic Signals

SHIYANG WANG, Electrical and Computer Engineering, Purdue University, West Lafayette, United States
XINGCHEN WANG, Electrical and Computer Engineering, Purdue University, West Lafayette, United States
WENJUN JIANG, Samsung Research America, Mountain View, United States
CHENGLIN MIAO, Department of Computer Science, Iowa State University, Ames, United States
QIMING CAO, Electrical and Computer Engineering, Purdue University, West Lafayette, United States
HAOYU WANG, Electrical and Computer Engineering, Purdue University, West Lafayette, United States
KE SUN, University of California, San Diego, La Jolla, United States
HONGFEI XUE, The University of North Carolina at Charlotte, Charlotte, United States
LU SU, Electrical and Computer Engineering, Purdue University, West Lafayette, United States

Accurately reconstructing 3D hand poses is a pivotal element for numerous Human-Computer Interaction applications. In this work, we propose SonicHand, the first smartphone-based 3D hand pose reconstruction system using purely inaudible acoustic signals. SonicHand incorporates signal processing techniques and a deep learning framework to address a series of challenges. First, it encodes the topological information of the hand skeleton as prior knowledge and utilizes a deep learning model to realistically and smoothly reconstruct the hand poses. Second, the system employs adversarial training to enhance the generalization ability of our system to be deployed in a new environment or for a new user. Third, we adopt a hand tracking method based on channel impulse response estimation. It enables our system to handle the scenario where the hand performs gestures while moving arbitrarily as a whole. We conduct extensive experiments on a smartphone testbed to demonstrate the effectiveness and robustness of our system from various dimensions. The experiments involve 10 subjects performing up to 12 different hand gestures in three distinctive environments. When the phone is held in one of the user's hands, the proposed system can track joints with an average error of 18.64 mm.

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

Shiyang Wang and Xingchen Wang contributed equally to this research.

Authors' Contact Information: Shiyang Wang, Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA; e-mail: wang5348@purdue.edu; Xingchen Wang, Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA; e-mail: wang2930@purdue.edu; Wenjun Jiang, Samsung Research America, Mountain View, CA, USA; email: wenjunji@buffalo.edu; Chenglin Miao, Department of Computer Science, Iowa State University, Ames, IA, USA; e-mail: cmiao@iastate.edu; Qiming Cao, Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA; e-mail: cao393@purdue.edu; Haoyu Wang, Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA; e-mail: wang5346@purdue.edu; Ke Sun, University of California, San Diego, La Jolla, CA, USA; e-mail: kesun@eng.ucsd.edu; Hongfei Xue, The University of North Carolina at Charlotte, Charlotte, NC, USA; e-mail: hongfei.xue@charlotte.edu; Lu Su (Corresponding author), Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA; e-mail: lusu@purdue.edu.

ACM 1550-4859/2024/08-ART106
https://doi.org/10.1145/3677122

Additional Key Words and Phrases: Acoustic sensing, hand pose reconstruction, device free, signal processing, deep learning, domain generalization

ACM Reference Format:
Shiyang Wang, Xingchen Wang, Wenjun Jiang, Chenglin Miao, Qiming Cao, Haoyu Wang, Ke Sun, Hongfei Xue, and Lu Su. 2024. Towards Smartphone-based 3D Hand Pose Reconstruction Using Acoustic Signals. *ACM Trans. Sensor Netw.* 20, 5, Article 106 (August 2024), 32 pages. https://doi.org/10.1145/3677122

1 Introduction

With the proliferation of smart devices, there is an increasing need for effective human-computer interaction techniques, which can precisely capture the activities and poses of the human body, especially the hand. By accurately reconstructing hand poses, the users are able to manipulate the objects in the cyber-world with unprecedented precision and enable a large variety of new applications. For instance, in virtual reality games, a player may use his hands to control the complicated movement of a virtual character's hands, which is impossible using traditional interfaces like a keyboard and mouse. In addition, hand pose reconstruction can enable people to control the appliances in a smart home. Reconstructing hand poses can be used in robotics as well. For example, using human hands to control robotic arms is an efficient way to manipulate objects that cannot be directly handled with human hands. The extensive application scenarios raise our interest in hand pose reconstruction tasks.

The existing work on hand pose reconstruction mainly adopts **computer vision (CV)** techniques to generate hand poses with the aid of visual information [9, 22, 30, 58–60, 84, 96]. Despite their superior accuracy, the risk of privacy disclosure leads to serious concerns about these methods. In addition to that, CV-based solutions cannot handle cases where there are occlusions in the **line of sight (LOS)** and bad lighting conditions. The computational cost of vision-based approaches are also high, especially when we want to deploy them on smart devices with limited computational resources. To overcome these limitations, wireless sensing comes into our view. Thus far, substantial prior studies have been conducted to reconstruct the poses of human body using **radio frequency (RF)** signals [1, 20, 69, 83, 91, 92]. However, comparing to the whole body, the size of the hand is much smaller, which demands higher sensing resolution. RF signals that require no additional hardware like Wi-Fi suffer from coarse resolution, which makes it impractical to reconstruct fine-grained hand poses. As for other RF-based solutions with high resolution, they rely on specialized devices like antenna arrays. Therefore, it is unlikely to build an RF-based system that is able to precisely reconstruct hand poses using daily smart devices. Fortunately, acoustic signals are not restricted by the preceding limitations. Low-cost audio infrastructures such as speakers and microphones are pervasive in smart devices, such as smartphones and smartwatches. This makes acoustic sensing systems easy to deploy. Moreover, acoustic signals have the potential to achieve finer resolution due to their relatively low propagation speed. In addition, the omnidirectional sensing angle brings acoustic sensing adequate flexibility for practical utilization. Thus, using acoustic signals to reconstruct the user's hand pose appears to be a feasible solution.

Nevertheless, precisely reconstructing the hand poses presents us with considerable challenges. First, compared with the existing approaches that focus on acoustic gesture recognition [15, 21, 51, 57, 70, 74] which only needs to classify the monitored activity into one of the predefined classes, and hand tracking [27, 32, 38, 41, 73, 86] that localizes the whole hand as a single point, hand pose reconstruction is a more challenging task that has never been studied in the area of acoustic sensing. In this task, we need to infer the 3D location of every hand joint, and

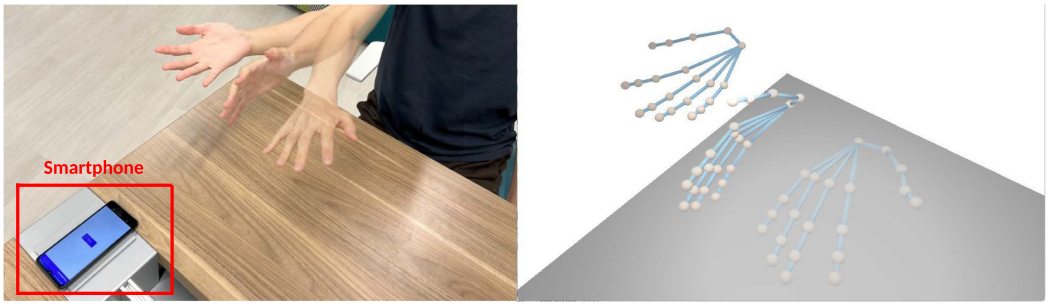


Fig. 1. Illustration of ultrasonic-based 3D hand pose reconstruction.

the generated hand poses should be realistic looking while being smooth and continuous. Second, once the model is trained, it should have the generalization ability for cross-environment and cross-subject inference. Third, the users may also move their hands arbitrarily when performing gestures, so we need to track the location of the hand as well as reconstructing the hand pose. To address the challenges discussed previously, we embrace a congruous combination of acoustic signal processing techniques and deep learning. Specifically, with the help of forward kinematics, we incorporate some prior knowledge of the hand skeleton into the deep learning model to reconstruct the hand poses. It enables our system to generate *realistic* and *smooth* hand poses. In addition, we rely on the **Doppler frequency shift (DFS)** to characterize poses. DFS is only determined by the velocity of the objects that reflect signals. Therefore, it should not be affected by the signals reflected from static objects in the environment. However, conducting cross-environment experiments in real-world scenarios still suffers from performance drop due to the secondary reflections which are reflected by the hand first and then reflected by the background objects again. Therefore, to further improve the cross-environment inference performance and tackle the cross-subject inference problem, we adopt adversarial training on feature representations, which results in a *robust* hand pose reconstruction model with the ability of domain generalization. To handle the case where the hand performs gestures while moving as a whole, we integrate a hand tracking method, based on the **channel impulse response (CIR)** estimation, into our system to measure the hand's location relative to the phone.

To evaluate the performance of our hand pose reconstruction framework, our acoustic system is deployed on a smartphone as shown in Figure 1. With the built-in speaker and microphones, we transmit ultrasonic signals which are completely inaudible and collect the reflected ones. In the meantime, ground truth poses are captured using **Leap Motion Controller (LMC)**, a vision-based device, to supervise the training of our deep neural network. The results show that the average joint localization error for 12 different gestures is 22.87 mm, whose supreme performance is also validated by the visualization of the reconstructed poses. Since it is natural to interact with a smartphone by holding it in the hand, we conduct experiments in such a realistic setting as well. By holding the smartphone in one hand while the other hand performs gestures on the side, the system can track joints with an average error of 18.64 mm. Extensive experiments are conducted to verify the robustness of our system in various dimensions.

The main contributions of our work are summarized as follows:

— To the best of our knowledge, our proposed system, SonicHand, is the first smartphone-based 3D hand pose reconstruction system using purely inaudible acoustic signals.
— To realize cross-environment and cross-subject inference, we leverage adversarial training to achieve satisfying cross-domain inference performance.

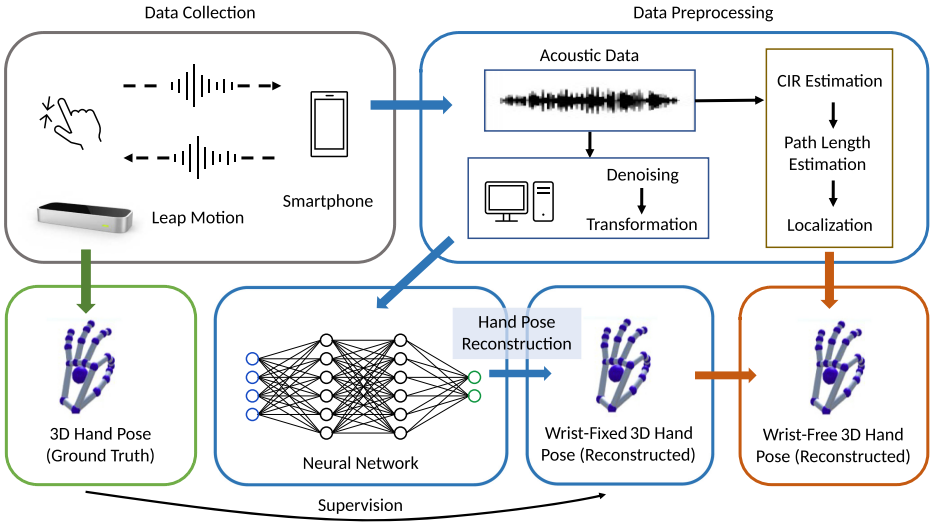


Fig. 2. System overview.

— We design a framework that combines deep learning based hand pose reconstruction and CIR estimation based hand tracking to reconstruct hand poses while tracking the hand as a whole.
— We implement the system on a smartphone and conduct comprehensive experiments and analyze its robustness in extensive scenarios to ensure that our system can be used in practical applications.

The rest of the article is organized as follows. Section 2 introduces the overview of our acoustic system. Section 3 describes the signal processing techniques, the neural network, and the adversarial training in our work. Section 4 presents the experiment devices, setup, and results. Section 5 summarizes the related work. Finally, Section 6 articulates our conclusions.

2 System Overview

We consider a scenario where the human subject is monitored by a smartphone with built-in speakers and microphones. The goal of the proposed system is to reconstruct the subject's 3D hand poses using acoustic signals collected from the smartphone. Figure 2 shows an overview of our acoustic hand pose reconstruction system, which mainly consists of three components: data collection, data preprocessing, and hand pose reconstruction:

— *Data collection*: A smartphone is utilized for transmitting the acoustic signal and collecting the reflected signal from the subject's hand. Only one speaker and two microphones on the smartphone are used. Besides the acoustic data, we also employ the LMC to capture the hand joint location precisely as the ground truth to train and test the neural network.
— *Data Preprocessing*: After obtaining the raw data, we demodulate the received signals and filter out the noise. Then we generate DFS profiles, which would be fed into the deep neural network to predict each joint's 3D location relative to the root joint, or wrist. Therefore, we can consider the wrist to be fixed and call the reconstructed hand pose *wrist-fixed 3D hand pose*. To handle the scenario where the user's hand is moving while performing gestures, we also localize the hand as a whole to obtain the hand pose whose wrist is moving freely. We call it *wrist-free 3D hand pose* as shown in Figure 2.

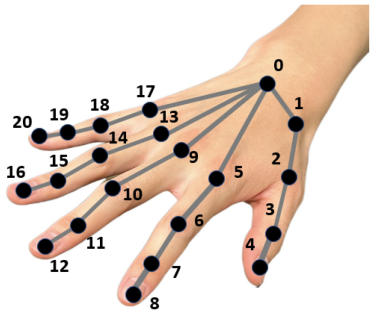


Fig. 3. Joint index illustration.

— *Hand pose reconstruction*: We leverage the power of the deep neural network to learn the spatial and temporal patterns from the input. With the prior knowledge of the hand structure, we can generate realistic-looking hand poses that are close to the ground truths. In addition, adversarial training is incorporated to enhance the generalization ability of the system.

3 Methodology

3.1 Overview

We aim to reconstruct hand poses using acoustic signals, which can be deployed on smartphones as software without any hardware modification.

The reason we pick the smartphone as the device to play and record acoustic signals is twofold. On the one hand, the smartphone is the most prevailing smart device with built-in speakers and microphones. Thus, smartphone-based sensing systems can be widely deployed in the real world. On the other hand, many applications on smartphones like mobile games take the hand pose as one of their interaction interfaces to support various fantastic functions. Thus, the smartphone serves as a proper platform for our system.

We make use of the DFS of the reflected acoustic signal as the feature of the hand movement. This is because DFS is only determined by the velocities of moving objects. With such a design, our system possesses the potential to be generalized to different environments. Moreover, generating DFS through **short-time Fourier transform (STFT)** is computational efficient, which can cut down on the time cost of feature extraction. After generating the DFS, we feed it into our deep learning model to reconstruct the hand pose. To obtain distinguishable DFS, we choose a 20-kHz sinusoidal signal as the transmitted signal. The reason is that the frequency shift caused by the hand movement will be centered around 20 kHz. Then we can easily remove the static reflection with unchanged frequency and capture the DFS.

Despite DFS containing the velocity information about the hand movement, its resolution is still coarse for fine-grained hand pose reconstruction. In addition, the generated hand poses should look realistic while being smooth, which is a nontrivial task. To properly reconstruct hand poses, we make use of forward kinematics [20, 62] to encode the prior knowledge of human hands' skeletal structure into our model. To be specific, we treat the segments of the hand as individual cylinders as shown in Figure 3 and represent the skeleton of the hand as a tree. The hand joints serve as the nodes, and the hand segments serve as the edges. Then we can estimate the rotations of hand segments instead of the absolute location of each hand joint, and recursively generate the joint location from the root joint to the leaf joints by applying the rotations to the skeletal structure. In this way, our system makes full use of the topological information of the human hand structure that ensures the generated hand poses look real. Besides that, in our neural

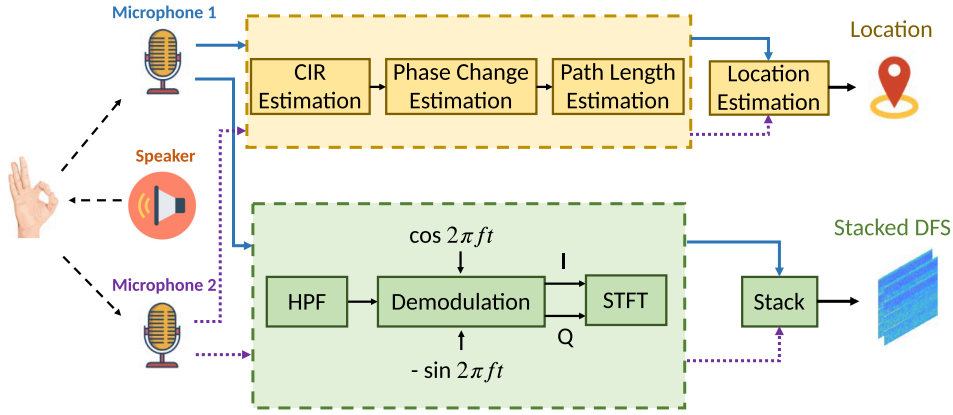


Fig. 4. Signal processing module.

network, we use a **convolutional neural network (CNN)** to extract the spatial information and a **recurrent neural network (RNN)** to extract temporal features of hand motion. The details about our model design will be discussed in Section 3.3.

In a wireless sensing area, the signals captured by sensors often contain substantial information related to both the surrounding environment where the activities are recorded and the human subject who performs the activities. Thus, a model trained by the data collected from certain subjects in a specific room usually cannot be applied directly to the data acquired from different subjects or other rooms. In our case, although DFS is only affected by reflections from moving targets, the patterns of some secondary reflections with nonzero velocities may also change in different environments. Different subjects have hands of different sizes and their unique characteristics when performing gestures. Those are the barriers preventing us from achieving robust cross-environment and cross-subject hand pose reconstruction. To address this challenge, we adopt the domain generalization technique to enable cross-environment and cross-subject inference. Specifically, we propose utilizing adversarial training as a means of improving the system's generalization capability. By doing so, we can facilitate cross-domain inference and enhance the overall performance of the system.

Although our model can precisely predict each joint's location relative to the root joint, which is the wrist in our case, there is still a limitation that it assumes the wrist stays at a fixed location. However, in real-life scenarios, users may perform hand gestures while moving their hands as a whole. To address this problem, we add a hand tracking function in the signal processing module to localize the wrist joint while reconstructing the hand pose. Due to the fact that pure tone signal cannot measure the absolute distance, other signals are required in our system to achieve accurate hand tracking. We adopt the **Zadoff Chu (ZC)** sequence [47] for the hand tracking task. The ZC sequence is able to measure the path length of the reflected signal by estimating the CIR. Details about our hand tracking method will be discussed in Section 3.2.

3.2 Signal Processing

After we obtain the mixed reflected signals which contain a pure tone sinusoidal signal and ZC sequence, we process them separately as shown in Figure 4.

Pure Tone Signal Modeling. The transmitted signal would be reflected by the hand and collected by the smartphone. We first apply a high pass filter to extract the pure tone signal $R(t)$ in the received signals. $R(t)$ is a mixture of signals reflected from different paths. We have

$$R(t) = R_s(t) + \sum_{l=1}^{L} 2A'_l \cos\left(2\pi f t - 2\pi f \frac{d_l(t)}{c} - \theta_l\right), \tag{1}$$

where $R_s(t)$ represents the signals from static paths (the LOS signal and signals reflected by stationary objects like walls) and the second term models the signals from the dynamic paths (the signals reflected by moving objects like hands). Considering the l-th dynamic path $R_l(t) = 2A'_l \cos(2\pi f t - 2\pi f \frac{d_l(t)}{c} - \theta_l)$, $2A'_l$ is the amplitude of the received signal, $d_l(t)$ is the time-varying path length of the l-th path, $2\pi f \frac{d_l(t)}{c}$ represents the phase delay caused by the propagation, and c is the propagation speed of sound in the air which is 340 m/s in our setting. The remaining term θ_l is the initial phase that is resulted from the hardware delay and phase inversion due to the reflection [73].

Pure Tone Signal Demodulation. After extracting the pure tone signals, we use the coherent detector to demodulate the received sound signals to baseband signals for further processing [61]. Two duplicated received signals are multiplied with the $\cos(2\pi f t)$ and $-\sin(2\pi f t)$, respectively. The multiplication will generate a low-frequency component and a high-frequency component. The high-frequency component of each signal can be removed by a low pass filter, then the in-phase signal and quadrature signal are generated correspondingly. For the l-th dynamic path, we can get the in-phase signal as $A'_l \cos(-2\pi f \frac{d_l(t)}{c} - \theta_l)$ and the quadrature signal as $A'_l \sin(-2\pi f \frac{d_l(t)}{c} - \theta_l)$. Combing these two components as the real and imaginary parts of a complex signal, we can represent the baseband signal from the L-th dynamic path in the following complex form:

$$B(t) = B_s(t) + \sum_{l=1}^{L} A'_l e^{-j(2\pi f d_l(t)/c + \theta_l)}, \tag{2}$$

where $B_s(t)$ is the static component after demodulation and the second term is the dynamic component containing L paths. Since only moving objects can cause frequency shift of the signals, the static component $B_s(t)$ can be easily filtered out. Thus, we can consider the sound signals that travels through the dynamic paths alone. We use $B_d(t)$ to represent it.

Doppler Frequency Shift. The movement of the subject will lead to the Doppler effect, which shifts the frequency of the signal collected by the microphone. The DFS is defined as the change rate of the length of the signal propagation path $d(t)$ as [20, 49, 72]

$$f_D(t) = -\frac{1}{\lambda}\frac{d}{dt}d(t), \tag{3}$$

where λ is the wavelength of the ultrasonic signals. Then we can finalize our multipath formulation of the dynamic component with DFS as

$$B_d(t) = \sum_{l=1}^{L} A'_l e^{j(2\pi \int_{-\infty}^{t} f_{D_l}(u)du - \theta_l)}. \tag{4}$$

Traditional fast Fourier transform generates a spectrum that contains the frequency components from the entire time period, making it difficult to describe the velocities at each time point. Since we want to generate a fine-grained DFS profile from $B_d(t)$, STFT is adopted here to quantify the change of a signal's frequency content over time [76]. In STFT, a sliding window function is used to extract the frequency information of a short period of time. In our case, the Blackman window is chosen to serve this purpose [2]. To achieve desirable resolution in both the time domain and the frequency domain, we choose a small window size and perform zero padding to increase the number of data points.

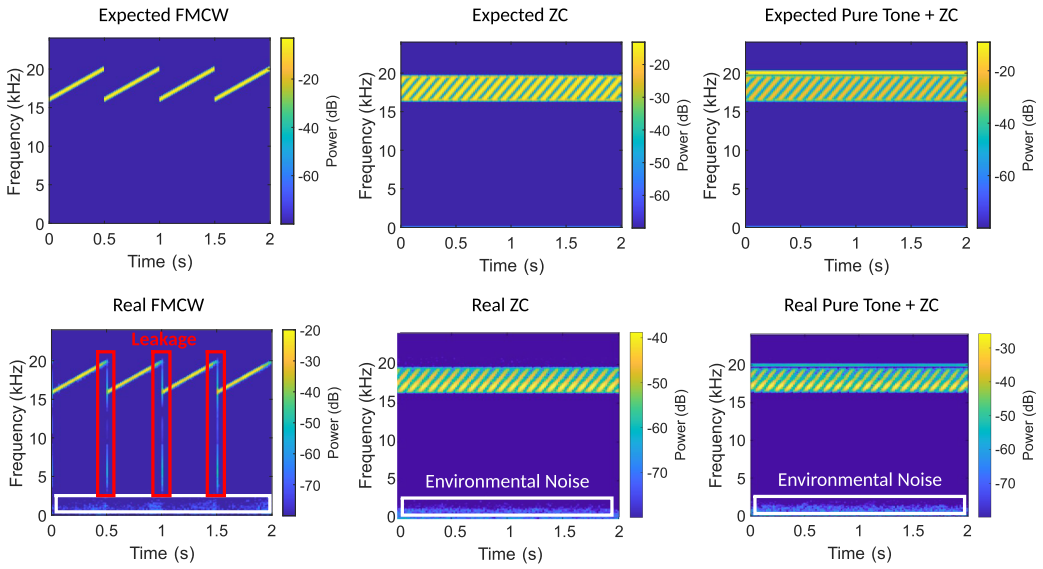


Fig. 5. The audible leakage analysis for three types of sensing signals on a Google Pixel 2.

Hand Tracking. To achieve precise hand tracking, we first use a ZC sequence to estimate the CIR of the acoustic signal and then measure the length of the reflection path. We choose the ZC sequence as our baseband signal because it has conspicuous advantages. First, a ZC sequence has ideal cross-correlation property, so the generated CIR has sharp peaks for reflection paths. This property makes it easier to separate different paths in CIR. Second, some other kinds of waveform that can measure absolute path length like frequency-modulated continuous wave suffer from audible sound leakage [26], which means that even if the acoustic signal we adopt has a frequency above the range that human can hear, it still generates annoying audible noise during the sensing process. In contrast, playing the ZC sequence will not generate any sound in audible frequency range, even when there is another pure tone signal playing. We conduct experiments to prove that as shown in Figure 5.

The N_{zc}-length ZC sequence is

$$zc[n] = exp\left(-j\frac{\pi un(n+1+2q)}{N_{zc}}\right), \tag{5}$$

where $n \in [0, N_{zc})$, q is a constant integer, and $u \in [0, N_{zc})$ is the parameter, which is an integer that satisfies $gcd(N_{zc}, u) = 1$. In our work, we set $N_{zc} = 119$ and $u = 61$. After generating the baseband ZC sequence, we modulate it to inaudible frequency using the OFDM modulation method proposed in the work of Wan et al. [66].

After the OFDM demodulation,we get the CIR frames along the time axis. The LOS signals of the microphone that are on the same side of the speaker will generate a peak with the highest value due to the fact that the LOS signal is much stronger than the reflected signals. We use the index of this peak in CIR as a reference point and circularly shift the CIR frame until this peak becomes the first point in CIR to eliminate the time delay of our system. To highlight the weak reflection path of the hand among all paths in CIR, we take the difference between the consecutive CIR frames so that the reflection from static objects will be removed and only the moving target will generate a peak. We can use the index of the peak to estimate the time of flight of the signal reflected by hand so that we can measure a coarse-grained path length.

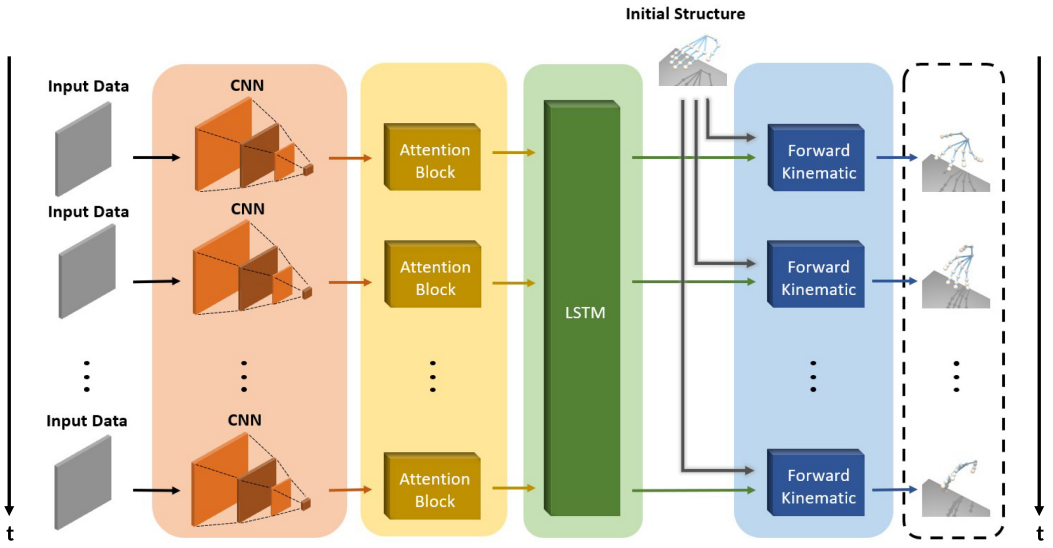


Fig. 6. Model overview.

Due to the fact that the resolution of time-of-flight estimation is limited by the sampling rate of the acoustic signal, we need to use the phase change of the reflection path to measure fine-grained path length change to improve the tracking accuracy. We leverage the curvature-based scheme proposed by Sun et al. [57] to measure the phase change. Combined with the initial absolute path length estimated by the index of the peak in CIR, we can obtain an absolute reflection path length for each microphone. After that, we can fuse the path length information from multiple microphones on the smartphone and utilize the geometric relationship between the speaker and microphones to localize the hand. Considering that most smartphones have two microphones, we track the hand movement in a 2D plane.

3.3 Neural Network

To reconstruct the 3D hand poses accurately, even when there is strong noise and interference in the environment, we develop a deep learning based hand pose reconstruction framework as shown in Figure 6.

Model Design. In the design of our deep learning model, CNN layers are first used to extract spatial information from the DFS profile. Four convolutional layers are piled on top of each other, and each layer's kernel size is decided by the dimension of input data. After each convolutional layer, the output will go through a batch normalization layer, a leaky rectified linear unit, and a dropout layer to normalize the mean and variance, introduce nonlinearity to the model, and prevent the model from overfitting.

To concentrate on dominating spatial features learned from CNN, we add a self-attention block to aggregate high-level spatial feature representation. The self-attention block is designed to learn the relative contributions of each spatial feature to the hand pose we try to reconstruct. We use a linear mapping function followed by a softmax layer to learn the relative weights.

Then we create a succession of feature vectors after the attention block. Since a hand movement often takes place over a period of time, the temporal relationships between successive data samples are highly correlated. We next send feature vectors into an RNN, which is the optimal model for this purpose since it can connect the hidden states of temporally dependent data, to learn the

relationship between successive data samples. Specifically, We use **long short-term memory (LSTM)** [14], an efficient and popular RNN, to capture relatively long movements. On top of the CNNs in our model, we added three-layer LSTMs.

To construct the poses of the hand subject through recursively estimating the rotation of the hand segments, we employ a process called *forward kinematics* [20, 62]. We follow the mathematical definition of the forward kinematics process in the work of Jiang et al. [20]. Considering a hand skeleton tree with N joints, the 3D coordinate of the i-th joint p^i can be obtained given the location of its parent joint $p^{parent(i)}$ and the initial position of p^i, $p^{parent(i)}$:

$$p^i = p^{parent(i)} + R^i \left(p_0^i - p_0^{parent(i)} \right), \tag{6}$$

where R^i is the rotation matrix of the joint p^i with respect to its parent. To improve the computational efficiency and numerical stability of our model, we leverage unit quaternions [77] to represent the 3D rotation group. Then we can use both the features extracted by the LSTM and the prior knowledge of the hand skeleton as the input of forward kinematics layer [62] to generate the 3D joint locations. In this manner, our neural network will concentrate on learning the rotation features, which are skeleton independent. If the bone structure of the hand subject in terms of segment length is not accessible, we can either use a conventional hand bone structure or roughly estimate the hand subject's bone structure using, for instance, a photo of the hand subject. In such case, the subject's hand bone structure is not required in the inference stage, as the hand motion with the precisely estimated rotations will construct the correct hand motion regardless of the slight difference between the hand structure we use and the hand structure of the real hand.

Loss Function. The model loss is summarized as follows:

$$\begin{aligned} Loss = & \frac{1}{T} \sum_{i=1}^{T} \frac{1}{N} \sum_{i=1}^{N} \left\| \hat{p}_t^i - p_t^i \right\|_2 \\ & + \beta \cdot \frac{1}{T-1} \sum_{t=2}^{T} \frac{1}{N} \sum_{i=1}^{N} \left\| (\hat{p}_t^i - \hat{p}_{t-1}^i) - (p_t^i - p_{t-1}^i) \right\|_H \\ & + \gamma \cdot \frac{1}{T} \sum_{t=1}^{T} \frac{1}{N-1} \sum_{i=2}^{N} \left\| \left(\hat{p}_t^i - \hat{p}_t^{parent(i)} \right) - \left(p_t^i - p_t^{parent(i)} \right) \right\|_H, \end{aligned} \tag{7}$$

where β and γ are the hyperparameters to control the weights of different losses. $\| \cdot \|_H$ is the Huber norm. In our case, we utilize *SmoothL1Loss*[1] to realize it in our implementation.

The first term in the loss function captures the *position loss*. This loss is proposed to minimize the difference between the predicted location of each joint i at each time slot t, represented by $\hat{p}_t^i$ and the associated ground truth p_t^i. We suppose that the hand skeleton tree has N joints and that the input DFS sequence has T frames, and we define the position loss as the mean squared error between $\hat{p}_t^i$ and p_t^i. The second term in the loss function represents the *smooth loss*. This loss is required because the posture of the hand at each time point in posture is handled separately by the position loss. Thus, the estimated hand pose will be trembling if we only have a position loss. The last term in the loss function is designed for the *rotation loss*. The localization error on the joint position may also accumulate throughout this procedure since the position of a joint is determined by recursively rotating the joints from the root joint to that joint. If the learned position of a joint's parent joint has already strayed from the true position, the position estimation for that joint may be inaccurate. Therefore, it is required to incorporate a loss to penalize the inaccuracy in a joint's relative location to its parents.

[1] https://pytorch.org/docs/stable/generated/torch.nn.SmoothL1Loss.html

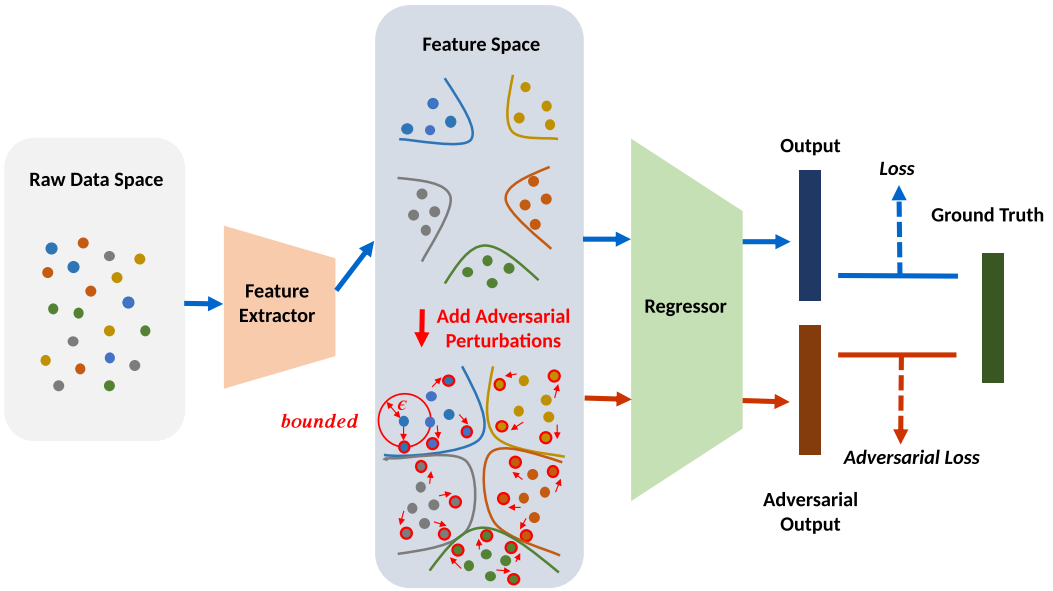
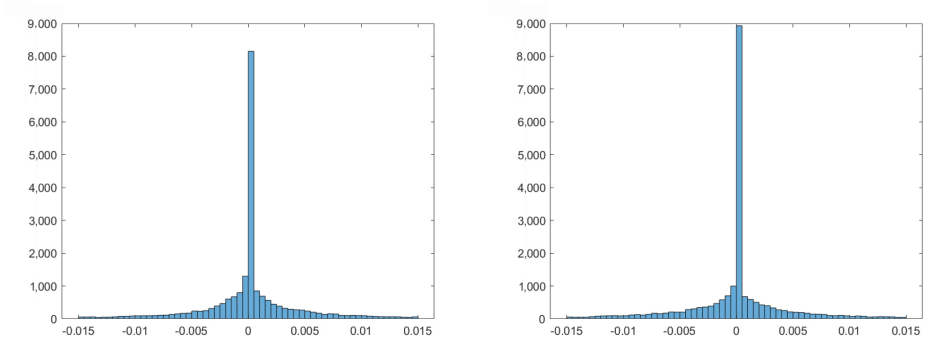


Fig. 7. Adversarial training overview. (By adding bounded perturbations in feature space, there is a great improvement in the generalization capacity of the regressor.)

3.4 Cross-Domain Inference

Here, we consider a practical problem setting: training and testing data are collected from different environments or subjects. In the context of this work, a *domain* is defined as a pair of an environment and a human subject. Although the main feature our system relies on is the velocity caused by the hand pose, which should be invariant to the environment, empirical results show that the performance of cross-environment inference still experiences an unacceptable decline. The possible reason is that acoustic signals can reflect more than once, so there are paths where the signals first are reflected by the moving hand and then are reflected again by the static background. The patterns of those reflections will change in different environments. Although the amplitude of those reflections are much weaker than the paths only reflected by the hand once, the performance of the model may drop because it is hard to learn a model that is robust to those perturbations using a small amount of training data. Additionally, the present work confronts the practical challenge of cross-subject inference. To overcome the difficulty of cross-domain inference, we borrow the idea of adversarial training [35, 64] to enhance the generalization capability of our system.

An overview of our adversarial training method is presented in Figure 7. Our approach partitions the model into a feature extractor (including CNN, attention block, and LSTM architectures) and a regressor (comprising a linear layer as unit quaternions predictor and forward kinematics, which predict the hand poses using learned feature representations). We assume that the feature extracted for the same hand pose is proximal in the feature space. However, with limited data from a seen domain, there is still significant space in the feature space unexplored where the data from an unseen domain may reside. To address this issue, adversarial perturbations are added to the existing training data to simulate the inclusion of data from an unseen domain during training. It is important to note that this approach is only viable if the feature extracted for a given hand pose from different domains is also proximal in the feature space. Our preliminary investigations have revealed that the feature extraction process for a given hand pose collected from different rooms, in



(a) Room 1 (b) Room 2

Fig. 8. The feature histograms exhibit a significant degree of similarity with respect to a given hand pose, across different environments, without adversarial training.

the absence of adversarial training, exhibits noteworthy similarities, as illustrated in Figure 8. This matches our intuition because the feature difference between different environments are caused by weaker secondary reflections, as we mentioned before. Hence, we propose to introduce minor perturbations within the feature space, which enables the development of a more robust model capable of generalizing across various domains, even in cases where training data is limited to a single domain.

Optimization Formulation. With the assumption that the same feature extraction methodology is applicable across multiple domains, the training of the pose feature extraction stage within our neural network pipeline remains unchanged, whereas adversarial training is solely applied to the regressor. Let us consider a hand poses reconstruction task with an underlying data distribution $\mathcal{D}$ over pairs of extracted pose feature representations $z \in \mathbb{R}^{n \times t}$ and corresponding hand poses (ground truth) sequences $p \in \mathbb{R}^{m \times t}$. It is worth noting that both types of data are time series data, where the variable t represents the temporal duration. Our goal is to minimize the discrepancy between predicted hand poses p' and the corresponding ground truth values p. We try to identify the optimal regressor parameters $\theta_r \in \mathbb{R}^d$ that minimize the risk $\mathbb{E}_{(z,p)\sim\mathcal{D}}[L_r(\theta_r, z, p)]$, where the predicted hand poses are generated through the use of the regressor, which leverages pose feature representations directly. The L_r means that it is only related to the regressor instead of the entire neural network.

Based on Figure 8, we can postulate that the pose feature from an unseen domain can be derived by introducing a minor perturbation to the pose feature from a seen domain if the two domains are similar. To add proper perturbations to the pose feature representations, we restrict the perturbation via a minor constant, denoted as $\|z' - z\|_{\mathcal{P}} \leq \epsilon$, where z' denotes a pose feature from an unseen domain, z denotes a pose feature from the seen domain, and ϵ captures the similarity between feature representations. The objective of incorporating perturbations is to maximize the loss of hand pose reconstruction, corresponding to the worst-case scenario. Minimizing the loss under such a perturbation can significantly enhance the model's generalization capacity to the greatest extent. The optimization is formulated as

$$\min_{\theta_r} \mathbb{E}_{(z,p)\sim\mathcal{D}} \left[\max_{\|\delta\|_{\mathcal{P}} \leq \epsilon} L_r(\theta_r, z + \delta, p) \right]. \tag{8}$$

In practice, we use sampled input points to compute the gradients and the expectation in Formula (8). As a result, we can simplify the problem by assuming, without loss of generality, that we

are dealing with an empirical set $\tilde{D}$ over pairs of feature representations and ground truths. The simplified problem is updated as

$$\min_{\theta_r} \sum_{(z,p)\in\tilde{D}} \max_{\|\delta\|_{\mathcal{P}}\leq\epsilon} L_r(\theta_r, z+\delta, p). \tag{9}$$

ALGORITHM 1: Adversarial training

Input: Training data examples X; corresponding ground truths P; training iterations I; balance control β; learning rate γ.

Output: Optimized model parameters θ.

Initialize the model parameters θ (which include the feature extraction parameters θ_f and the regressor parameters θ_r).

for each iteration i = 1 to I **do**

  $x \leftarrow$ Next batch of examples from X

  $p \leftarrow$ Next batch of ground truths from P

  $z \leftarrow$ FeatureExtraction(θ_f,x)

  $\delta \leftarrow$ GetPerturbations (z, p) ▷ Find an adversarial perturbation for a high loss

  $\theta_f = \theta_f - \gamma\nabla_{\theta_f} L(\theta, x, p)$

  $\theta_r = \theta_r - \gamma\nabla_{\theta_r}(L(\theta, x, p) + \beta L_r(\theta_r, z+\delta, p))$

end for

ALGORITHM 2: GetPerturbations with $l_{\mathcal{P}}$-norm

Input: Features z; corresponding ground truth p; optimization rounds T; perturbation limitation ϵ; learning rate α; regressor parameters θ_r.

Output: Optimized perturbation δ.

Randomly initialize δ with $\|\delta\|_{\mathcal{P}} \leq \epsilon$

for each round t = 1 to T **do**

  $\delta = \delta + \alpha\,\text{sign}(\nabla_\delta L_r(\theta_r, z+\delta, p))$

  clamp δ to ensure $\|\delta\|_{\mathcal{P}} \leq \epsilon$

end for

return δ

The overall adversarial training process is depicted in Algorithm 1 while Algorithm 2 is invoked by Algorithm 1 to obtain the optimal perturbation within the feature space. The adversarial training task can be described as a two-step optimization process. The first step involves an inner maximization process, where the objective is to identify an adversarial version of a given hand feature representation z that yields a high loss of hand pose reconstruction (further elaborated in Algorithm 2):

$$\max_{\|\delta\|_{\mathcal{P}}\leq\epsilon} L_r(\theta_r, z+\delta, p). \tag{10}$$

The second step, known as the outer minimization process, aims to identify optimal regressor parameters that minimize the "adversarial loss" obtained in the inner maximization problem:

$$\min_{\theta_r} \sum_{(z,p)\in\tilde{D}} L_r(\theta_r, z+\tilde{\delta}, p). \tag{11}$$

Furthermore, as the generalization ability of our model becomes improved, it is equally important to verify that the adversarially trained model performs well on the original training dataset. To achieve this, we introduce a balance control variable β as a hyperparameter in our adversarial training algorithm, as shown in step 8 of Algorithm 1. During adversarial training, it governs the contribution of the adversarial loss under perturbations to the overall loss for the update of the regressor parameters. This approach allows us to strike a balance between optimizing the model's performance in both seen and unseen domains, thereby achieving superior outcomes overall.

4 Experiments

4.1 Testbeds

4.1.1 Leap Motion Controller. In this work, we use the LMC to obtain the ground truth of 3D hand poses and movements. As a vision-based device, LMC has two **infrared light (IR)** cameras

(a) Leap Motion Controller (b) Pixel 2 (c) Data collection setup

Fig. 9. Testbed setup.

and three IR emitters, which are shown in Figure 9(a). The current application programming interface of LMC provides the positions in Cartesian space. The LMC itself is the center of the coordinate system. The origin is located at the top center of the hardware. In our experiments, the sampling rate of LMC is set to 100 Hz, and this device can estimate the position of hand joints with errors less than 1.2 mm [75]. Since LMC can only stably track the hands at a maximum distance of 60 cm, we place LMC below the hand with a vertical distance of 45 cm and connect it to a laptop to obtain the ground truth data (as shown in Figure 9(c)).

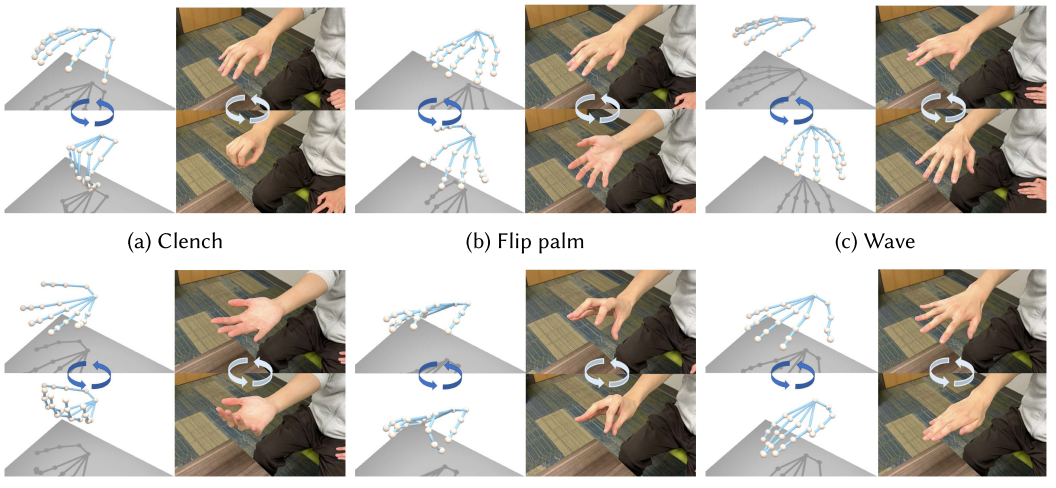
4.1.2 Acoustic Testbed. Our acoustic system can be deployed on most COTS devices with basic acoustic functionality. In our experiments, we use a smartphone (Google Pixel 2 [78]) as the default acoustic testbed. We use one speaker and two microphones on the phone. The speaker is used as the transmitter, and the two microphones are used as receivers. We develop an Android app that can play 20-kHz pure tone as well as a 16.5- to 19.5-kHz ZC sequence. At the same time, it will record stereo audio at 44.1 kHz and log the precise timestamps to synchronize with the ground truths whose timestamp is recorded based on the laptop's clock. To guarantee the synchronization between the ground truth data and the collected acoustic data, we use network time protocol to synchronize the clock between the phone and the laptop. With this approach, we can achieve an average synchronization error less than 10 ms.

4.2 Data Collection and Preprocessing

In our experiments, 10 volunteers (3 female and 7 male) are employed as human subjects. In the basic scenario, we consider six different gestures for the hand pose reconstruction task: clench, flip palm, wave, bend fingers, pinch, and close fingers, which are shown in Figure 10. We extend the number of gestures to 12 in the following gesture inclusiveness experiments. To simplify the data collection, we ask each subject to perform each gesture repetitively for 1 minute in each trial. As shown in Figure 9(c), when we collect data, we fix a single smartphone using a phone tripod. Due to the fact that most users will keep their hands within 50 cm from the phone during daily use, we set the horizontal distance between the smartphone and the subject's hand to 50 cm. The setting for data collection will be adjusted accordingly in the following various experiments.

We also use the LMC to collect the hand pose data for model training and evaluation while collecting the acoustic data using the smartphone. The pose data is sampled at a rate of 100 Hz, and the acoustic data is sampled at a rate of 44.1 kHz. After obtaining raw data, we preprocess the acoustic data and the pose data separately.

We first extract the 20-kHz pure tone signals using a high pass filter from the collected acoustic data and demodulate it with a coherent detector as described in Section 3.2. Then, we apply STFT on the processed data to generate the DFS profiles which contain the velocity information of the hand movements. We adjust the window size so that the generated DFS profile has 10 frames



(a) Clench (b) Flip palm (c) Wave

(d) Bend fingers (e) Pinch (f) Close fingers

Fig. 10. Gestures illustration.

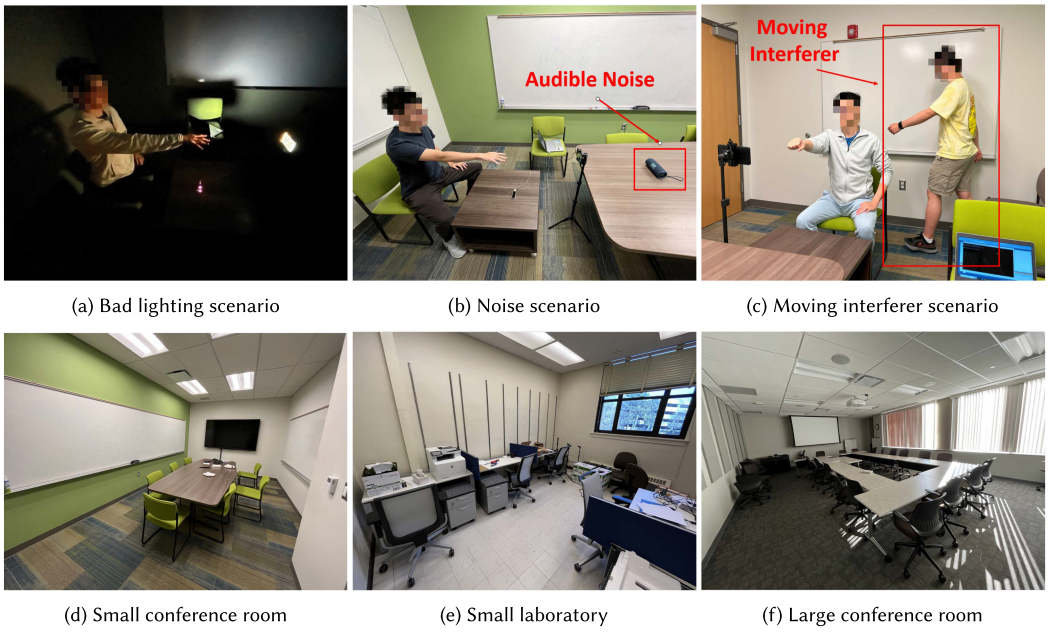
for each second and each frequency bin is nearly 1 Hz. During the whole preprocessing step, we handle the two microphones' data separately and only pick the lower 128 frequency bins given that the frequency shift caused by the hand gestures is usually small. Thus, we can get a $600 \times 2 \times 128$ matrix that can be used as the feature input.

We then downsample the ground truth pose data to a 10-Hz frame rate so that it can be aligned with the DFS profiles. To obtain uniform sampling periods, we further apply nearest-neighbor interpolation to them. Since there are 21 joints in our hand skeleton model and the model is in a 3D-coordinate measuring system, we can get a $600 \times 21 \times 3$ matrix as the ground truth.

As for the hand tracking, the length of the baseband ZC sequence (N_{zc}) and the value of u are set to 119 and 61, respectively. We also modulate the ZC sequence to a central frequency $f_c = 18$ kHz with bandwidth 3 kHz. Therefore, the ZC sequence and the pure tone do not have any overlap in the frequency domain.

4.3 Model Setting and Implementation

For the stacked four-layer CNNs, we choose 2D convolution operation with the numbers of convolutional filters as 64, 128, 64, and 1, respectively. Considering the potential overfitting problem, we add dropout layers [56] to drop out the nodes in our neural network with a dropout rate of 0.5 at each step during training time. However, applying dropout to a neural network typically increases the training time. To decrease the number of training epochs required by the convergence of the deep neural network, we utilize the 2D batch normalization [18] to normalize the contributions to a layer for every mini-batch. In addition, Leaky ReLU [80] is adopted as an activation function with a negative slope rate of 0.02 to introduce nonlinearity to the CNNs. To improve the interpretability of the weight learned by attention block, we also add a softmax as an activation function after the linear mapping layer. For the three-layer LSTMs, we set the number of features in the hidden state to 336 and the dropout rate to 0.1. The predefined hyperparameters in the loss function (i.e., β, γ) are set to 1, and the learning rate of Adam is set to 0.001. For adversarial training, we set the balance control β, optimization rounds T, perturbation limitation ϵ, and learning rate α to 1, 5, 0.01, and 0.001, respectively. By default, we choose l_∞-norm to confine perturbations.



(a) Bad lighting scenario (b) Noise scenario (c) Moving interferer scenario

(d) Small conference room (e) Small laboratory (f) Large conference room

Fig. 11. Experimental setup and different environments.

We segment the training data along time axis and generate 1.5-second-long sequences with overlapping. Then we feed those sequences into our deep learning model. Regarding the testing data, although we use a fixed length for each sample during the evaluation, it is feasible to input collected ultrasonic signals of arbitrary length into our system, which will generate a time series of reconstructed hand poses with an equivalent temporal duration. Finally, the neural network is implemented using PyTorch [44]. We train the neural network with a batch size of 32 and run the program on a desktop with an NVIDIA A6000 GPU and an Intel Xeon Gold 6254 CPU. The inference is conducted on the same machine.

4.4 Experimental Results

We consider different scenarios for the performance evaluation. Specifically, we first consider a basic scenario where the subjects perform gestures with their right hands without any interference. Then we evaluate the robustness of the proposed system to different factors, such as a bad lighting condition, audible noise, and a moving interferer. In addition, we analyze the generalizability of our model by conducting cross-environment/cross-subject experiments and evaluate the performance of our system overtime. We further evaluate the performance when our system is handling a hand that performs gestures while moving as a whole, which is common in real-world applications. The performance of our system when the phone is held in the hand, which is how most people use their phones, and the performance on different smartphones are also studied. Additionally, handling static hand postures is another practical issue we investigate. At the end, we measure the running time of the proposed system.

4.4.1 Basic Scenario. For the basic scenario, we collect data following the description in Section 4.2 in a small conference room, which is shown in Figure 11(d). For each trail with 60 seconds of data, we divide the collected data into two parts: 80% of the data (the first 48 seconds) is used for training and 20% of the data (the remaining 12 seconds) is used for testing. We utilize

Table 1. Average Joint Localization Errors (in Millimeters) for the Single-Hand Scenario

JI	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Overall
BA	13	19	25	30	19	25	31	35	16	22	28	32	15	21	27	32	16	22	27	33	23.25
BL	12	18	24	31	19	24	29	34	17	22	27	31	15	21	26	31	14	20	24	28	22.20
NO	14	20	26	32	19	23	28	32	17	22	26	31	16	21	27	31	16	21	25	30	22.69
MI	14	19	27	34	23	29	35	43	19	26	32	39	16	23	30	37	16	23	28	34	26.12
GI	16	20	25	30	18	21	24	27	17	21	26	30	17	23	30	36	18	23	27	31	22.87

JI, Joint Index; BA, Basic; BL, Bad Lighting; NO, Noise; MI, Moving Interferer; GI, Gesture Inclusiveness.

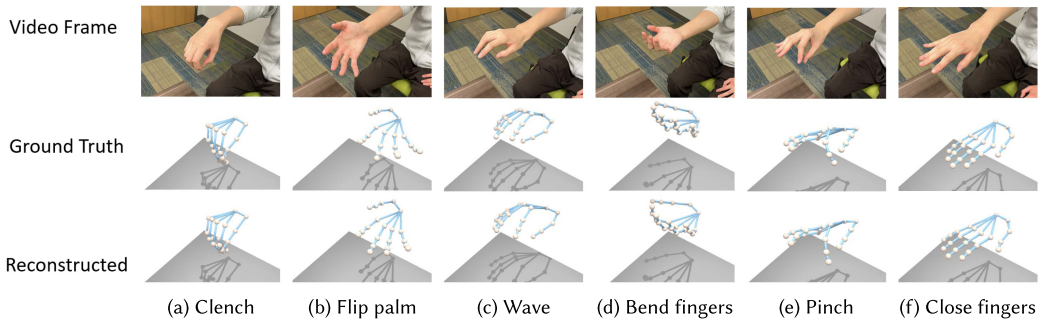


Fig. 12. The examples of the constructed hand poses in the basic scenario.

the average joint localization error as the performance metric, which is the average Euclidean distance between the predicted joint location and the actual joint location over all subjects and hand poses.

The average joint localization errors for the basic scenario are shown in Table 1 (the "BA" row). The error for each joint is the average over all testing data of six different hand poses and 10 participants. We also calculate an overall average error for all joints in the last column of Table 1, which is only 23.25 mm. Considering that we calculate the relative position of each joint with the wrist location (the origin of the coordinate), the error of the wrist joint is always 0 mm. Therefore, we only report the results for the other 20 joints. To get a better understanding of how good our predicted results are, we visualize the reconstructed hand pose and compare them with the visualized ground truths and the corresponding video frames. As shown in Figure 12, we pick one example frame for each hand pose. The first row is the video frame captured by the camera on a smartphone during the data collection process. The second row is visualized by PyOpenGL[2] based on the ground truth data collected by the LMC. Our predicted hand poses are visualized in the same way and shown in the last row. We can observe that the reconstructed hand poses are realistic looking and almost the same as the ground truths.

Gesture Smoothness Analysis. Besides the average joint localization error, we also evaluate the smoothness of the reconstruction. Figure 13 shows some consecutive frames of the reconstructed hand poses for gesture 2 (flip palm) in the basic scenario. The video version[3] of visualization is provided as well with a frame rate of 10 Hz. The 12 gestures include 6 in the basic scenario and the other 6 from the following analysis of gesture inclusiveness. We can see that the reconstructed poses are close to the ground truth and smooth. This is mainly because we incorporate smooth loss

[2]https://pyopengl.sourceforge.net/
[3]https://drive.google.com/file/d/1uFFkrnuD2mbf3RoeUo5HYkYdqw0ILBjO/view?usp=sharing

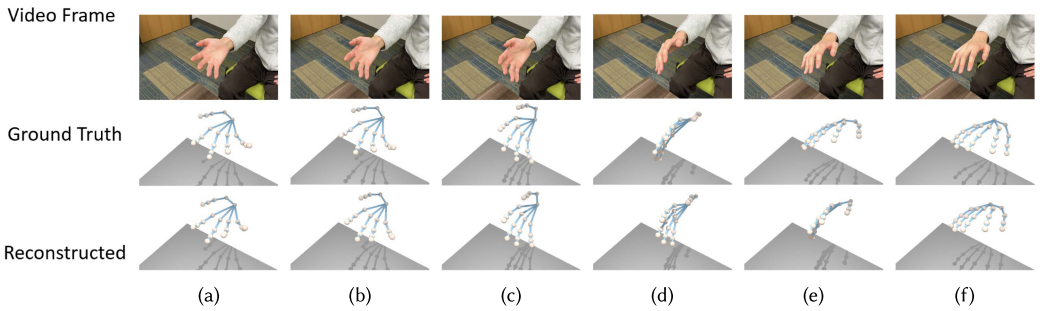


Fig. 13. The consecutive examples of the constructed hand poses for gesture 2 (flip palm) in the basic scenario.

and LSTM into our neural network. The temporal relationships are extracted precisely to exhibit the correlation between successive data samples. To quantitatively evaluate the effectiveness of the smooth loss, we further conduct experiments wherein the smooth loss was applied and omitted. We measure the average displacement of each joint between consecutive frames, serving as an indicator of smoothness. Specifically, using only the position loss and the rotation loss results in an average displacement of 5.01 mm. With the help of the smooth loss, this metric drops to 4.45 mm. Concurrently, the average joint localization error exhibited a decline from 24.02 mm to 23.25 mm. Therefore, we can conclude that our loss design can effectively reconstruct hand poses with a blend of smoothness and accuracy.

STFT Window Size Analysis. We select a proper window size for the STFT to generate DFS with the expected frame rate. However, that does not mean our model is sensitive to the STFT window size. To investigate the performance of our system under different window size, we train our model five times using DFS generated with different STFT window sizes. By modifying the overlap between consecutive windows, we can still get 10 frames of DFS per second when the window size changes. The results illustrated in Figure 15(a) shows that the hand pose reconstruction accuracy is stable, which proves that our system is not sensitive to the STFT window size.

Inference Length Analysis. Although the data length we collect for each subject is fixed, our model can also deal with flexible lengths of inference data in daily use. To prove this, we conduct experiments where we segment sequences with different lengths from the inference data and feed them to our model. The average joint errors of different inference data lengths are shown in Figure 15(b). We can find that the performance of our model remains stable when the length of inference data changes. Therefore, our system can handle real-world cases where the inference data has an arbitrary length.

Gesture Inclusiveness Analysis. To demonstrate the inclusiveness of our system in recognizing various gestures, we expand our experiments by considering more gestures. The new gestures include gestures with minor variations to those already present in our system (pinch middle, pinch ring, pinch pinky), as well as more intricate gestures (rocker, point, peace) as shown in the video frame part of Figure 14. Sequentially, we introduce the additional gestures one by one into the training set, followed by an evaluation of the system's performance on the corresponding testing set. The empirical results, depicted in Figure 15(c), substantiate the system's consistent stability while accommodating an increasing number of diverse gestures. We further demonstrate the effectiveness of our approach by visualizing the reconstructed poses in Figure 14. We can observe that the reconstructed hand poses are close to the ground truth while capturing the subtle differences between similar pinch gestures that use different fingers as shown in Figure 14(a) through 14(c).

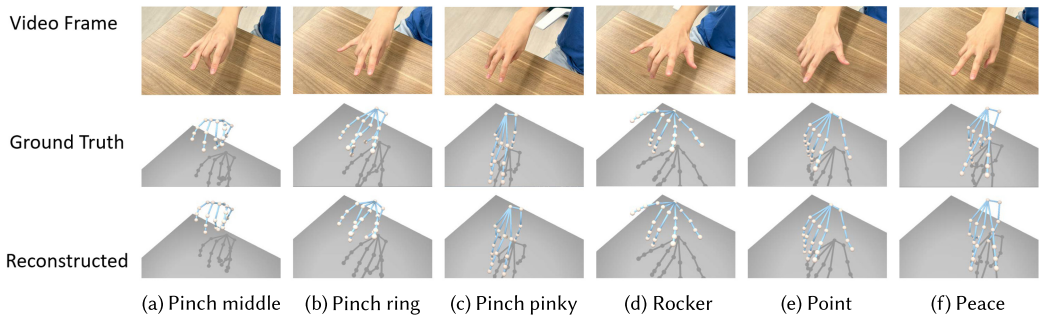
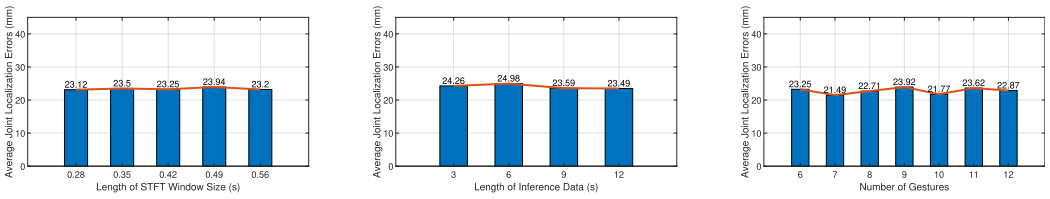


Fig. 14. The examples of the constructed hand poses for additional gestures.



(a) Different STFT window size (b) Different length of inference data (c) Different number of gestures

Fig. 15. Evaluation plots of different lengths of inference data and different number of gestures.

4.4.2 Robustness in Different Scenarios. In real-world applications, acoustic sensing systems would work in more complicated environments, where many interference factors may produce a negative impact on the performance. To demonstrate the robustness of our system in practice, we evaluate the performance of our model in different scenarios. Specifically, we investigate scenarios involving challenging lighting conditions, audible background noise, and the presence of moving interferers. In each scenario, we test the inference using the model trained in the basic scenario directly, which means that collecting new training data for those scenarios is not required.

Robustness to Bad Lighting Conditions. Another advantage of the proposed system over vision-based methods is that it is robust to bad lighting conditions. To demonstrate this point, we collect data using the same process as that in the basic scenario after turning off all the lights in the conference room. As shown in Figure 11(a), there is only some brightness from the smartphone and the laptop (driving the LMC to collect ground truth). Based on the results shown in the "BL" row of Table 1, the average joint localization error under bad lighting conditions is similar to that in the basic scenario. The results show that the bad lighting condition almost has no effect on the performance of the proposed system.

Robustness to Audible Noise. As an acoustic system, one potential issue is how to deal with the surrounding noise in real life. Owing to the meticulous choice of the ultrasonic signal, we can easily filter out all of the audible noise. As shown in Figure 11(b), we play music as background noise to test the system's robustness. The intensity level of the background noise is around 60 dB, which is comparable to the noise level generated by a vacuum cleaner.[4] The average joint localization errors in this scenario are shown in the "NO" row of Table 1. As we can see, the system can achieve an overall error of 22.69 mm, which demonstrates the robustness of the system to audible noise.

[4]https://pulsarinstruments.com/news/decibel-chart-noise-level/

Table 2. Average Joint Localization Errors (in Millimeters) for the Cross-Environment Scenarios

JI	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Overall
BA	13	19	25	30	19	25	31	35	16	22	28	32	15	21	27	32	16	22	27	33	23.25
CE1	17	25	32	38	23	32	40	47	19	27	34	41	19	26	34	41	20	28	35	42	29.58
CE2	16	23	30	35	22	30	38	45	19	26	34	40	18	25	33	40	19	27	33	40	28.31

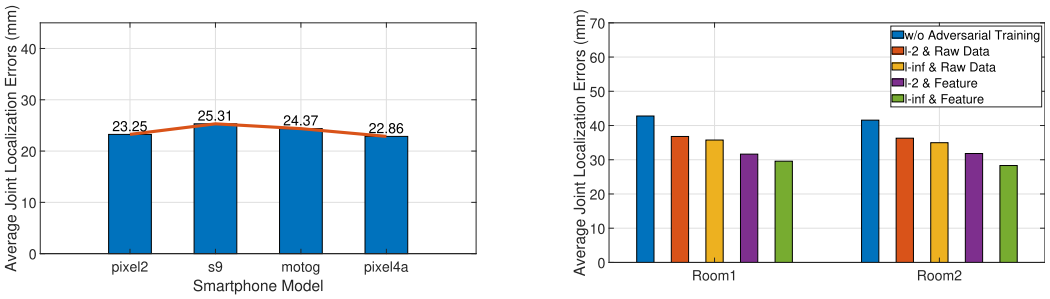
In the cross-environment experiments, we use the model trained in the basic scenario to infer the data collected in another room. JI, Joint Index; BA, Basic; CE1, Room 1 under a cross-environment setting; CE2, Room 2 under a cross-environment setting.

Robustness to the Moving Interferer. Since static interference exhibits consistent behavior over time, we address its effect by choosing DFS as the extracted feature that exclusively leverages changes of the sensing subject. However, the presence of dynamic interference in the environment may be a confounding factor for the system. In light of this concern, we conduct an experiment to study the effect of a moving interferer from the surrounding environment as shown in Figure 11(c). When the user is interacting with the phone, usually the hand is the closest moving object to the phone. Therefore, we only study the scenario where there is a subject walking behind the subject who is performing gestures. The results in Table 1 show that the average joint localization error remains acceptably low, with a recorded value of 26.12 mm. These findings suggest that the moving interferer that is not very close to the phone does not significantly affect the system's overall performance.

4.4.3 Model Generalization. In daily use, our system may be used at different rooms for different users. It is unfavorable if the users need to collect new training data once they move to a new room or add a new user. We also want the performance of our system to be stable as time goes on so that the model does not need to be retrained frequently. Therefore, in this section, we study whether it is possible to train the system once and then apply it to "anywhere" (cross-environment inference) for "anybody" (cross-subject inference) at "any time" (stable performance over time).

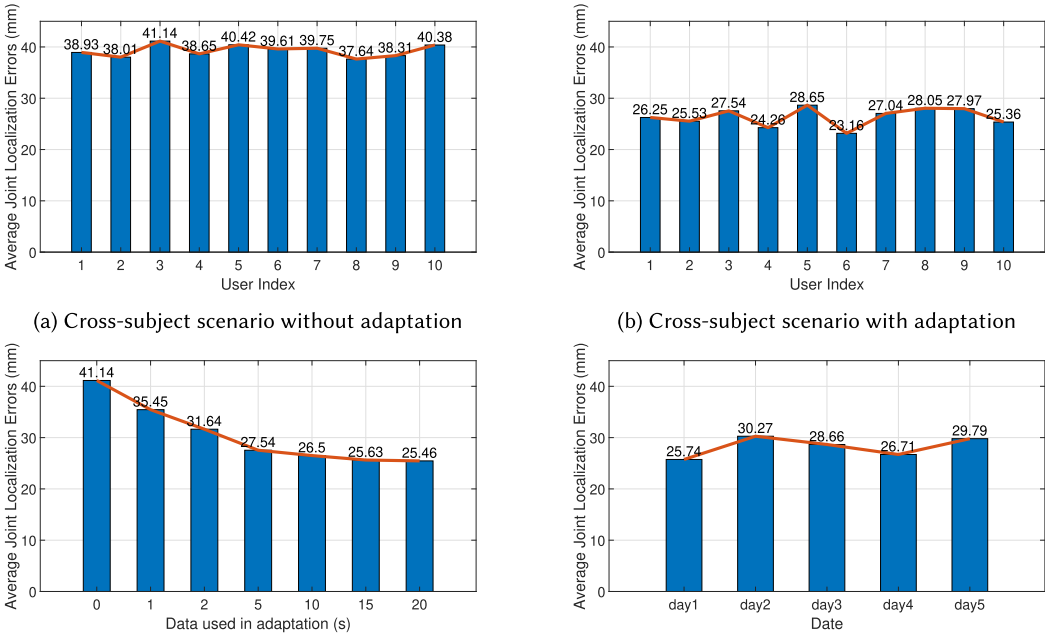
Cross-Environment Performance. To explore the possibility of applying the trained model anywhere, we evaluate the performance of the proposed system for cross-environment pose construction (Table 2). Specifically, we train the model using the data collected in one room (the small conference room used in the basic scenario as shown in Figure 11(d)) and test it using the data collected in other rooms (a small laboratory and a large conference room as shown in Figure 11(e) and Figure 11(f), respectively). We call the small laboratory Room 1 and the large conference room Room 2 for convenience.

By directly applying the trained model, we find that the cross-environment setting degrades the performance of the system. To address this challenge, we use adversarial training (as discussed in Section 3.4) to improve the performance of the system. Specifically, we add minor perturbations during training to enhance the generalization ability of our model. As mentioned in the work of Madry et al. [35], besides dealing with the feature representation, perturbations can be introduced to the raw data as well, whereas the different norms (like l_∞ norm or l_2 norm) can be utilized to confine perturbations. Thus, we evaluate the efficacy of adversarial training with different choices of where to add the perturbation and how to bound it. As shown in Figure 16(b), empirical evidence demonstrates that the incorporation of l_∞ bounded perturbations in feature space yields the most effective performance in cross-environment inference. With such an optimized configuration of adversarial training, the performance in cross-environment settings becomes significantly improved compared to the case where there is no adversarial training applied, which leads to a reduction in the average joint localization error from 42.78 mm to 29.58 mm for Room 1 and from 41.57 mm to



(a) Different smartphone models (b) Cross-environment scenario

Fig. 16. Evaluation plots of different smartphone models and different configurations of adversarial training for the cross-environment.



(a) Cross-subject scenario without adaptation (b) Cross-subject scenario with adaptation

(c) Data required for adaptation (d) Errors over time

Fig. 17. Evaluation plots of cross-subject scenarios and performance over time.

28.31 mm for Room 2 as shown in Figure 16(b). These findings suggest that the system with adversarial training has acceptable generalization ability and can be deployed in a new environment.

Cross-Subject Performance. To further assess whether our model can generalize to new users, we conduct leave-one-user-out experiments, which means that we train the model using the data collected from nine subjects and subsequently test it using the data collected from the remaining one. These experiments also incorporate adversarial training. However, the result in Figure 17(a) reveals that the average error becomes 39.28 mm despite the help of adversarial training, which is unsatisfactory. The reason is that different users have unique hand shapes and their own habits of performing gestures, which makes DFS profiles extracted vary across different subjects even for the same gesture. Thus, appropriate adaptation is necessary for cross-subject

scenarios. Specifically, we add few data from the new user into the training process to enhance the performance. To investigate how much data is required, we pick the user with the worst performance in Figure 17(a) as the target user in the adaptation experiment. We gradually add data of the new user into the training process and evaluate the model with the same testing data. Note that the training data and testing data are never overlapped. From Figure 17(c), we can see that the performance is improved significantly until the data length used in adaptation reaches 5 seconds. This observation substantiates our contention that 5 seconds of data from each new user can ensure acceptable performance in the cross-subject scenario. We also repeat the leave-one-out experiments with 5-second adaption data. As shown in Figure 17(b), the average joint localization errors are decent, even compared with the one in the basic scenario.

Performance over Time. For the "any time" perspective, we collect data on consecutive days to evaluate the generalization ability of the model in the temporal dimension. We train the model using data collected on day 0 and test the model using the data collected on the following 5 days. As shown in Figure 17(d), the average joint localization error is pretty stable over time. Most of the average joint localization errors are within 30 mm, which are comparable to the results in the basic scenario. Thus, our system can be trained once and then used over a long period of time.

4.4.4 Wrist-Free Scenario. When the user's hand is performing gestures while moving as a whole, we need to track the location of the hand as the coordinate of the wrist. In this section, we evaluate the performance of our system in a wrist-free scenario. In this work, we assume that people use their hand poses to interact with their phones in a restricted area near the phones so that the patterns of the DFS for each gesture will not change a lot. This setting is reasonable because it can meet the requirements of many daily applications on smartphones. Therefore, we let the subjects simultaneously perform gestures and move their hands in an area of 25×25 cm in front of the phone, which is enough to cover the typical area where the users perform their hand gestures.

Hand Tracking Performance. We first evaluate the hand tracking accuracy of our system. For each gesture, the subjects will move their hands along six different trajectories in the 25×25 cm area. To obtain a stable reflection for initial location estimation, the subjects turn their palms toward the phone at the beginning of each trajectory. We evaluate the tracking accuracy in a 2D plane because most smartphones have two available microphones, which are able to support 2D tracking.

At the beginning, we evaluate the tracking error of our method on a moving hand without any other gestures as a reference. In such cases, our system can achieve an average 2D tracking error of 9.0 mm, which is comparable to the state-of-the-art smartphone-based tracking system. Then the subjects are asked to move their hands and perform gestures. Most of the gestures have average tracking errors less than 20 mm, as shown in Figure 18. We can observe that hand gestures will degrade the tracking accuracy, and the impact of gestures on the tracking result depends on gesture types. For gestures that do not move the palm a lot, such as gesture 6 (close fingers), the tracking performance can still be good. However, for gesture 2 (flip palm) that rapidly changes the orientation of the whole palm, our method has limited tracking accuracy. The reason is that the hand has its own size and can cause a multipath effect. If the shape of the reflection surface that faces the phone has a significant change, then the multipath effect caused by hand also changes, which makes the tracking unstable. However, even with different gestures, the hand tracking accuracy of our system is still enough to satisfy the needs of many daily applications that do not require extremely precise location, such as mobile games.

Wrist-Free Pose Reconstruction. Although the location of the hand can be tracked, the DFS caused by the hand gesture may also be affected with hand movement. Therefore, we next evaluate

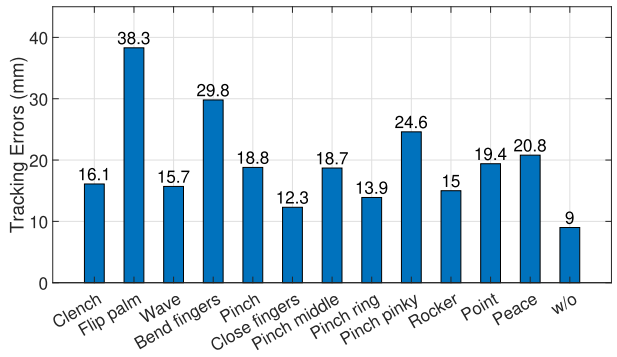
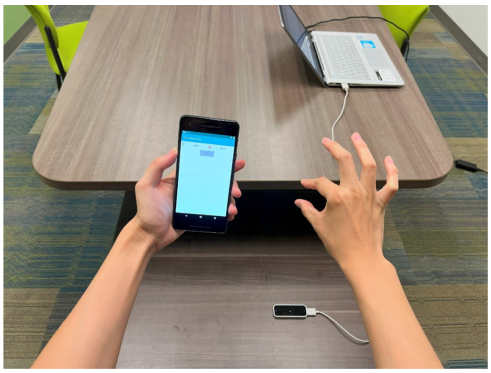


Fig. 18. Hand tracking performance when performing different gestures (w/o represents without any gesture while the hand is tracked).

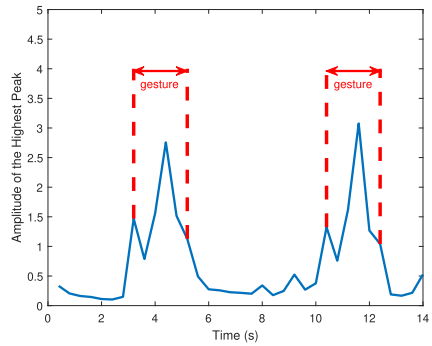
whether our system can be trained at a fixed location and tested when the hand is moving around the training location. We ask each subject to perform each gesture at a fixed location for 60 seconds as the training set. Then they will perform the same gestures when their hands are moving in the 25×25 cm area around this location for another 60 seconds as the testing set. Our study shows that the average joint localization error of the wrist-free scenario is 32.08 mm, which does not degrade severely compared to the wrist-fixed pose reconstruction. Therefore, for the users who interact with their phones with hands in front of their phones, they can directly use the previous model trained at a location near the phone without the requirement for additional training data collection.

4.4.5 Practical Issue Analysis. In the deployment of our system within real-world applications, some practical issues need to be considered. For example, the users may not want to put the phone on a table or in a holder. Instead, they can hold the phone in one hand while performing gestures with another hand. Another challenge arises from the fact that many users do not have a leap motion, leading to difficulties in collecting precise ground truth. In addition, the users may keep their hands static for a while between gestures, which requires further discussion. Last, whether different models of smartphones with varying speaker and microphone layouts will affect the performance is also important when our system is applied in practical use. In this section, we will discuss those concerns and conduct experiments to prove that our system is able to handle those practical issues well.

In-Hand Scenario. When the user is interacting with a smartphone, it is natural to hold the phone in one hand and control the application using the other hand. Therefore, we evaluate our system in a more realistic scenario, where the subjects hold the phone in their left hand and use their right hand to perform the gestures defined in the basic scenario. Other settings are also the same as in the basic scenario. Due to the fact that most of the users feel that it is convenient to put their hand on the side of the phone during the interaction, the subjects are asked to perform gestures on the right side as shown in Figure 19(a). We train a model using 80% of the in-hand data and test it with the remaining 20% data. The results are listed in Table 3. We notice that our system even achieves better performance in the in-hand experiments than in the basic scenario because the distance between the hand and the phone is closer in this setting, which results in a better sensing signal to noise ratio. Those results also prove that omnidirectional acoustic sensing has a broader sensing angle compared to the camera-based solution, which makes our system suitable to handle the case where the user performs hand gestures on the side of the phone. Consequently, we believe that our system is able to be integrated into real-world smartphone applications.



(a) In-hand experimental setup



(b) Result of gesture segmentation using differential CIR

Fig. 19. Practical issue analysis.

Table 3. Average Joint Localization Errors (in Millimeters) for the In-Hand Scenario

JI	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Overall
BA	13	19	25	30	19	25	31	35	16	22	28	32	15	21	27	32	16	22	27	33	23.25
IH	13	15	20	26	15	18	24	29	13	17	22	25	13	17	22	26	14	17	21	25	18.64

JI, Joint Index; BA, Basic; IH, In-hand scenario.

Collecting Data without a Leap Motion. Due to the fact that not all users have access to leap motion, acquiring training data could be challenging for them. Hence, we propose an alternative solution for those users. Specifically, we can employ predefined unified hand poses as the established ground truth used for training. To synchronize the collected acoustic data with this ground truth, the smartphone can showcase a video demonstrating hand gestures while playing acoustic signals, and the user is asked to perform the gestures following the video guidance. To show the effectiveness of this method, we conduct experiments where our model is trained using a unified ground truth. The experimental setup remains the same with the basic scenario, except for a difference that we choose one subject's hand poses as the unified ground truth. The model is trained using the acoustic data from each subject alongside the unified ground truth. Then we assess the model's performance using testing data and ground truth from all subjects. In such cases, the average joint localization error is 27.68 mm, indicating only a minor degradation. This result shows that our system has the potential to be applied even in the absence of a leap motion.

Handling Static Hand Postures. Previous experiments have demonstrated that our model accurately reconstructs hand poses when users perform gestures. However, the users' behaviors may become more intricate in real-world scenarios. At times, the user may perform a gesture and maintain the resulting static posture for a while. In such cases, it is expected that our model will preserve this static posture as the output until the user executes the next gesture. An intuitive approach to accomplish this goal is to include some static hand postures in our training data, enabling the model to generate correct hand poses even when the hand is not moving. Nevertheless, this presents a challenge, as a static hand will not contribute to the DFS profile, which is used to train our model. Consequently, noise will dominate the profile, making it quite difficult to train a robust model using such data. Therefore, we design a threshold-based hand movement segmentation method that utilizes the CIR estimated in the hand tracking module to precisely segment hand movement. To elaborate, we calculate the differential of CIR [31] to eliminate the static reflections. Then we measure the amplitude of the highest peak as an indicator

of hand movement and apply a threshold to segment the discontinuous gestures. The DFS will be fed into the model to predict the new hand poses only when the hand is engaged in gestures. Otherwise, if the hand is static, the estimated hand pose will remain unchanged. We ask a subject to perform a hand gesture, then stay static for a few seconds, and then perform another gesture, and the result is shown in Figure 19(b). We can clearly see that the amplitude of the highest peak is obviously larger when there is a hand gesture. Therefore, we can easily set a threshold to segment the hand gestures to handle the scenario where the user maintains a static hand gesture.

Performance on Different Smartphones. In practical applications, our system will be deployed to different models of smartphones with different layouts, which raises concerns about whether the phone model will affect the pose reconstruction accuracy. To evaluate the performance of our system on different models of smartphones, we conduct experiments using different models of smartphones to collect data. Besides the Google Pixel 2 used in the basic scenario, we repeat the experiment in the basic scenario using Samsung Galaxy S9, Motorola moto g pure, and Google Pixel 4a. For each model of smartphone, 80% of the data is used for training and 20% of the data is used for testing. According to the results in Figure 16(a), our system achieves the average joint localization errors of 23.25 mm, 25.31 mm, 24.37 mm, and 22.86 mm, respectively, which shows that our system can work well on different models of smartphones.

4.4.6 Running Time Analysis on the Smartphone. To demonstrate the efficiency of our pose reconstruction system for deployment on mobile devices, we conduct experiments aimed at evaluating the runtime performance of the proposed methodology. These experiments were conducted on an OnePlus 9 smartphone using PyTorch Mobile.[5] Concretely, we separate the process after we collect the raw data into two steps. First, we measure the time required to extract the DFS feature from the raw data. Then, the DFS feature will be fed to our deep learning model to assess the inference time. The result shows that the DFS generation step takes 0.038 seconds while the deep learning model inference takes 0.019 second for one frame. Therefore, it is short enough to support a frame rate of 17.5 fps, and it can be further accelerated when deployed on more advanced smartphones. Thus, our approach has the potential to be implemented as a real-time system.

5 Related Work

5.1 Acoustic Human Sensing

Gesture Recognition. Recently, acoustic signals have widely been used in gesture and activity recognition applications. Some previous works [5, 15, 21, 46, 51] propose to leverage the Doppler effect caused by the movement of the hand to recognize different hand gestures. To further improve the performance, some researchers [57, 70, 74] estimate the CIR of the acoustic signal to achieve gesture recognition. By estimating CIR using the ZC sequence, VSkin [57] is able to separate the structure-borne sounds and the air-borne sounds. Then both of them can be used to recognize hand gestures on the back of mobile devices. Furthermore, Wang et al. [70] propose a system that only extracts the CIR of the structure-borne component to avoid being interfered by other movements in the air. Xu et al. [82] use a wristband with a microphone and optic motion sensors to detect micro finger gestures. Nonetheless, the preceding approaches can only classify coarse-grained hand gestures and are deficient in the ability to estimate fine-grained hand poses. In our work, we take advantage of the prior knowledge of the hand skeleton by exploiting forward kinematics [62] so that our system has the ability to reconstruct the 3D location of every joint of the user's hand.

Tracking. A large number of researchers [4, 13, 36, 45, 67, 85, 90] focus on how to track a smart device, such as the smartphone. AAMouse [85] and CAT [36] can track a mobile device based on

[5] https://pytorch.org/mobile/android/

the DFS caused by the movement of the device. MilliSonic [67] reaches submillimeter 1D tracking accuracy using the frequency-modulated continuous wave phase. EarphoneTrack [4] localizes the user by tracking the earphone's motion. Ge et al. [13] model the sound field when there are two speakers playing sound simultaneously to localize a smartphone. But those methods are designed to track a device that can record acoustic signals. Therefore, their methods cannot be directly used for hand pose reconstruction.

To make the usage more natural, researchers have started to study how to use acoustic signals to build device-free tracking systems [27, 32, 38, 41, 73, 86], which means that the users do not have to attach any device or sensors to their body. FingerIO [41] sends OFDM pulses and performs correlations between the received and transmitted signals to measure the distance. LLAP [73] and Strata [86] track hand motion using the phase change of the reflected acoustic signals. RTrack [38] and FM-Track [27] further increase the tracking range and are able to track multiple targets with the help of microphone arrays. Although many of the preceding approaches have good tracking accuracy, they consider the whole hand as a point and are unable to localize all joints of the hand, which means that they are unable to estimate the hand pose well. Our system overcomes this problem by predicting the rotation of each hand joint and then generating the hand pose with the help of forward kinematics [62].

Imaging. The existing work that is most similar to ours is acoustic imaging. Traditional acoustic imaging methods such as ultrasonography require a large microphone array or a microphone with precisely controlled motion. In addition, many of them use ultrasonic signals with very high frequencies that cannot be generated by daily smart devices. The requirement of dedicated devices makes these methods difficult to implement in daily life. Therefore, some researchers expend their efforts to achieve acoustic imaging using COTS devices [37, 71]. AIM [37] moves a smartphone along a predefined trajectory as a virtual microphone array based on a synthetic aperture radar technique to achieve 2D imaging. However, it requires that the target remain static during the scanning. Thus, it cannot be used to image a hand that performs gestures. Amaging [71] focuses on hand shape imaging and is able to handle a moving hand. But it can only estimate a coarse 2D hand shape, so it cannot be used as a good interface to interact with smart devices. Our system can precisely reconstruct 3D hand poses of the user, which enables a broader application scenario than the previous 2D imaging methods.

5.2 RF Signal Based Pose Reconstruction

In recent years, estimating human pose using RF signals has become a popular research area. A few authors [1, 91, 92] first proposed RF signal-based systems to estimate 2D or 3D human skeletons. Those works are superior to the vision-based methods when the lighting condition is not good or there is occlusion between the subject and sensors. However, they all require expensive specialized hardware, such as a T-shaped antenna array built and synchronized using USRP. To overcome this limitation, researchers have started to make efforts in using WiFi signals to estimate human pose. Wang et al. [69] propose a 2D human pose reconstruction system based on channel state information extracted from WiFi signals. Thereafter, WiPose [20] and GoPose [50] were proposed to reconstruct 3D human skeletons from WiFi signals. There are also some works that focus on using RF signals to extract more information than the skeleton of human subjects. With a mmWave radar, mmMesh [83] is able to estimate both the body shape and pose of the subjects and then construct the mesh of the body. However, the preceding RF signal based methods can only reconstruct coarse-grained human poses like the skeleton of the whole body. To reconstruct fine-grained hand pose, mm4Arm [34] is an indirect method that uses mmWave radar to sense the vibration of the user's forearm, which is caused by finger movements. Nonetheless, mmWave signals are highly directional, which means that the radar needs to face the forearm. This property strictly limits its

application ranges. On the contrary, the speakers and microphones on the smartphones are omni-directional, which supports the users while interacting with the phone from anywhere they feel comfortable. To the best of our knowledge, the system proposed in our work is the first one to use acoustic signals generated by daily smart devices to estimate fine-grained hand poses.

5.3 3D Hand Pose Reconstruction

Computer Vision Based Methods. Hand pose reconstruction has been studied for a long time in the CV area. Existing vision-based approaches can be categorized into depth camera based methods [10, 11, 23, 43, 48, 65, 81] and RGB camera based methods [3, 12, 40, 53–55, 96]. Nonetheless, vision-based approaches raise the risk of causing privacy issues, as they require recording videos. Furthermore, the performance of current vision-based methods degrades obviously when the lighting condition is poor. In the cases where the hand is occluded, it is hard to retain good performance in the preceding methods as well. In contrast, our system is based on acoustic signals, which are not restricted by those limitations.

Wearable Device Based Methods. Wearing specialized devices on the hands is another way to precisely estimate 3D hand pose. To reconstruct the hand skeleton or mesh of the user, researchers have developed a variety of systems based on glove-like devices with different kinds of sensors, such as those described in various works [7, 16, 25, 39, 42, 94]. Digits [24], DorsalNet [79], and FingerTrak [17] use wrist-mounted cameras or thermal cameras for hand pose reconstruction. All of the preceding methods require additional costs for the wearable device. Moreover, wearing such kinds of devices may be troublesome for the users, which narrows the application scenarios of wearable device based systems. Compared to these works, our system can be easily deployed with only one smartphone without any other expensive hardware. Additionally, our system is completely device-free and inaudible, which makes it user-friendly.

5.4 Cross-Domain Inference

Domain Adaptation. Several works [19, 28, 29, 33, 52] aim to extract domain-independent features leveraging the knowledge from the samples of different domains. Chen et al. [6] and Zhang et al. [87] utilize data augmentation to increase the capability of cross-domain inference. Feng et al. [8] and Zhou et al. [95] incorporate meta-learning to address the challenges faced in domain adaptation. Zhang et al. [88] and Zhang et al. [89] construct domain-transferable mapping to eliminate the gap between different domains. The scenario we consider suffers from the absence of target domain data in the training process. As a consequence, conventional domain adaptation techniques are not readily applicable to our problem.

Domain Generalization. Wang et al. [68] minimize the maximum mean discrepancy between each pair of source domains to make the feature codes able to be generalized to an unseen domain. Zheng et al. [93] extract body-coordinate velocity profiles, which are domain independent and act as unique indicators of gestures. Virmani and Shahzad [63] present a translation function to automatically generate virtual samples for the target domain under all possible domain configurations. Jiang et al.t [20] extend the body-coordinate velocity profile to a 3D velocity profile that can capture the movements of the 3D space. Our system is enhanced with the inclusion of physical domain-independent features (DFS) in conjunction with adversarial training. This approach serves to augment the system's domain generalization capacity.

6 Conclusion

In this article, we studied whether inaudible acoustic signals generated by a smartphone have the ability to accurately estimate fine-grained hand poses. Specifically, we aimed to infer the locations of all joints in the hand. To this end, we proposed an ultrasonic-based 3D hand pose reconstruction

system SonicHand, which relies on a single speaker and two microphones of a smartphone to interact with the user. With the help of forward kinematics and a deep learning model, our system is able to leverage the topological information of the hand skeleton as prior knowledge and precisely reconstruct high-quality and realistic-looking hand poses. At the same time, our system has the generalization ability for cross-environment and cross-subject inference due to the adoption of adversarial training. We also integrated a CIR estimation based hand tracking method into our system to obtain the location of the hand pose when the hand is moving. We implemented our system on a smartphone testbed and demonstrated its superior performance through extensive experiments.

References

[1] Fadel Adib, Chen-Yu Hsu, Hongzi Mao, Dina Katabi, and Frédo Durand. 2015. Capturing the human figure through a wall. *ACM Transactions on Graphics* 34, 6 (2015), 1–13.

[2] Ralph Beebe Blackman and John Wilder Tukey. 1958. The measurement of power spectra from the point of view of communications engineering—Part I. *Bell System Technical Journal* 37, 1 (1958), 185–282.

[3] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. 2018. Weakly-supervised 3D hand pose estimation from monocular RGB images. In *Proceedings of the European Conference on Computer Vision (ECCV'18)*. 666–682.

[4] Gaoshuai Cao, Kuang Yuan, Jie Xiong, Panlong Yang, Yubo Yan, Hao Zhou, and Xiang-Yang Li. 2020. EarphoneTrack: Involving earphones into the ecosystem of acoustic motion tracking. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 95–108.

[5] Ke-Yu Chen, Daniel Ashbrook, Mayank Goel, Sung-Hyuck Lee, and Shwetak Patel. 2014. AirLink: Sharing files between multiple devices using in-air gestures. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 565–569.

[6] Xi Chen, Hang Li, Chenyi Zhou, Xue Liu, Di Wu, and Gregory Dudek. 2020. Fido: Ubiquitous fine-grained WiFi-based localization for unlabelled users via domain adaptation. In *Proceedings of The Web Conference 2020*. 23–33.

[7] Jean-Baptiste Chossat, Yiwei Tao, Vincent Duchaine, and Yong-Lae Park. 2015. Wearable soft artificial skin for hand motion detection with embedded microfluidic strain sensing. In *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA'15)*. IEEE, 2568–2573.

[8] Chao Feng, Nan Wang, Yicheng Jiang, Xia Zheng, Kang Li, Zheng Wang, and Xiaojiang Chen. 2022. Wi-Learner: Towards one-shot learning for cross-domain Wi-Fi based gesture recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 3 (2022), 1–27.

[9] Liuhao Ge, Yujun Cai, Junwu Weng, and Junsong Yuan. 2018. Hand PointNet: 3D hand pose estimation using point sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8417–8426.

[10] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. 2016. Robust 3D hand pose estimation in single depth images: From single-view CNN to multi-view CNNs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3593–3601.

[11] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. 2017. 3D convolutional neural networks for efficient and robust hand pose estimation from single depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1991–2000.

[12] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 2019. 3D hand shape and pose estimation from a single RGB image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10833–10842.

[13] Linfei Ge, Qian Zhang, Jin Zhang, and Qianyi Huang. 2020. Acoustic strength-based motion tracking. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 4 (2020), 1–19.

[14] Klaus Greff, Rupesh K. Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. 2016. LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems* 28, 10 (2016), 2222–2232.

[15] Sidhant Gupta, Daniel Morris, Shwetak Patel, and Desney Tan. 2012. SoundWave: Using the Doppler effect to sense gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1911–1914.

[16] Pei-Chi Hsiao, Shu-Yu Yang, Bor-Shing Lin, I.-Jung Lee, and Willy Chou. 2015. Data glove embedded with 9-axis IMU and force sensing sensors for evaluation of hand function. In *Proceedings of the 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'15)*. IEEE, 4631–4634.

[17] Fang Hu, Peng He, Songlin Xu, Yin Li, and Cheng Zhang. 2020. FingerTrak: Continuous 3D hand pose tracking by deep learning hand silhouettes captured by miniature thermal cameras on wrist. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 2 (2020), 1–24.

[18] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning*. 448–456.

[19] Wenjun Jiang, Chenglin Miao, Fenglong Ma, Shuochao Yao, Yaqing Wang, Ye Yuan, Hongfei Xue, Chen Song, Xin Ma, Dimitrios Koutsonikolas, Wenyao Xu, and Lu Su. 2018. Towards environment independent device free human activity recognition. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. 289–304.

[20] Wenjun Jiang, Hongfei Xue, Chenglin Miao, Shiyang Wang, Sen Lin, Chong Tian, Srinivasan Murali, Haochen Hu, Zhi Sun, and Lu Su. 2020. Towards 3D human pose construction using WiFi. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 1–14.

[21] Yincheng Jin, Yang Gao, Yanjun Zhu, Wei Wang, Jiyang Li, Seokmin Choi, Zhangyu Li, Jagmohan Chauhan, Anind K. Dey, and Zhanpeng Jin. 2021. SonicASL: An acoustic-based sign language gesture recognizer using earphones. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 2 (2021), 1–30.

[22] Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. 2012. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *Computer Vision—ECCV 2012*. Lecture Notes in Computer Science, Vol. 7577. Springer, 852–863.

[23] Sameh Khamis, Jonathan Taylor, Jamie Shotton, Cem Keskin, Shahram Izadi, and Andrew Fitzgibbon. 2015. Learning an efficient model of hand shape variation from depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2540–2548.

[24] David Kim, Otmar Hilliges, Shahram Izadi, Alex D. Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. 2012. 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*. 167–176.

[25] Jaehwan Lee, Sanghyeok Kim, Jinjae Lee, Daejong Yang, Byong Chon Park, Seunghwa Ryu, and Inkyu Park. 2014. A stretchable strain sensor based on a metal nanoparticle thin film for human motion detection. *Nanoscale* 6, 20 (2014), 11932–11939.

[26] Dong Li, Shirui Cao, Sunghoon Ivan Lee, and Jie Xiong. 2022. Experience: Practical problems for acoustic sensing. In *Proceedings of the 28th Annual International Conference on Mobile Computing and Networking*. 381–390.

[27] Dong Li, Jialin Liu, Sunghoon Ivan Lee, and Jie Xiong. 2020. FM-Track: Pushing the limits of contactless multi-target tracking using acoustic signals. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 150–163.

[28] Danyang Li, Jingao Xu, Zheng Yang, Yumeng Lu, Qian Zhang, and Xinglin Zhang. 2021. Train once, locate anytime for anyone: Adversarial learning based wireless localization. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM'21)*. IEEE, 1–10.

[29] Hang Li, Xi Chen, Ju Wang, Di Wu, and Xue Liu. 2021. DAFI: WiFi-based device-free indoor localization via domain adaptation. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 4 (2021), 1–21.

[30] Hui Liang, Junsong Yuan, and Daniel Thalmann. 2014. Parsing the hand in depth images. *IEEE Transactions on Multimedia* 16, 5 (2014), 1241–1253.

[31] Kang Ling, Haipeng Dai, Yuntang Liu, Alex X. Liu, Wei Wang, and Qing Gu. 2020. UltraGesture: Fine-grained gesture sensing and recognition. *IEEE Transactions on Mobile Computing* 21, 7 (2020), 2620–2636.

[32] Chao Liu, Penghao Wang, Ruobing Jiang, and Yanmin Zhu. 2021. AMT: Acoustic multi-target tracking with smartphone MIMO system. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM'21)*. IEEE, 1–10.

[33] Jianwei Liu, Jinsong Han, Feng Lin, and Kui Ren. 2020. Adversary helps: Gradient-based device-free domain-independent gesture recognition. *arXiv preprint arXiv:2004.03961* (2020).

[34] 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. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 6, 3 (2022), 1–28.

[35] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* (2017).

[36] Wenguang Mao, Jian He, and Lili Qiu. 2016. CAT: High-precision acoustic motion tracking. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. 69–81.

[37] Wenguang Mao, Mei Wang, and Lili Qiu. 2018. AIM: Acoustic imaging on a mobile. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. 468–481.

[38] Wenguang Mao, Mei Wang, Wei Sun, Lili Qiu, Swadhin Pradhan, and Yi-Chao Chen. 2019. RNN-based room scale hand motion tracking. In *Proceedings of the 25th Annual International Conference on Mobile Computing and Networking*. 1–16.

[39] Hadrien O. Michaud, Laurent Dejace, Séverine De Mulatier, and Stéphanie P. Lacour. 2016. Design and functional evaluation of an epidermal strain sensing system for hand tracking. In *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'16)*. IEEE, 3186–3191.

[40] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. 2018. GANerated hands for real-time 3D hand tracking from monocular RGB. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 49–59.

[41] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. 2016. FingerIO: Using active sonar for fine-grained finger tracking. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 1515–1525.

[42] Brendan O'Flynn, J. Torres Sanchez, James Connolly, Joan Condell, Kevin Curran, Philip Gardiner, and Barry Downes. 2015. Integrated smart glove for hand motion monitoring. In *Proceedings of the 6th International Conference on Sensor Device Technologies and Applications*.

[43] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A. Argyros. 2011. Efficient model-based 3D tracking of hand articulations using Kinect. In *Proceedings of the British Machine Vision Conference (BMVC'11)*, Vol. 1. 3.

[44] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NIPS'19)*. 8026–8037.

[45] Chunyi Peng, Guobin Shen, Yongguang Zhang, Yanlin Li, and Kun Tan. 2007. BeepBeep: A high accuracy acoustic ranging system using COTS mobile devices. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*. 1–14.

[46] Corey Pittman, Pamela Wisniewski, Conner Brooks, and Joseph J. LaViola Jr. 2016. Multiwave: Doppler effect based gesture recognition in multiple dimensions. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. 1729–1736.

[47] Branislav M. Popovic. 1992. Generalized chirp-like polyphase sequences with optimum correlation properties. *IEEE Transactions on Information Theory* 38, 4 (1992), 1406–1409.

[48] Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. 2014. Realtime and robust hand tracking from depth. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1106–1113.

[49] Kun Qian, Chenshu Wu, Zheng Yang, Yunhao Liu, and Kyle Jamieson. 2017. Widar: Decimeter-level passive tracking via velocity monitoring with commodity Wi-Fi. In *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. 1–10.

[50] Yili Ren, Zi Wang, Yichao Wang, Sheng Tan, Yingying Chen, and Jie Yang. 2022. GoPose: 3D human pose estimation using WiFi. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 2 (2022), 1–25.

[51] Wenjie Ruan, Quan Z. Sheng, Lei Yang, Tao Gu, Peipei Xu, and Longfei Shangguan. 2016. AudioGest: Enabling fine-grained hand gesture detection by decoding echo signal. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 474–485.

[52] Rui Shu, Hung H. Bui, Hirokazu Narui, and Stefano Ermon. 2018. A DIRT-T approach to unsupervised domain adaptation. *arXiv preprint arXiv:1802.08735* (2018).

[53] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. 2017. Hand keypoint detection in single images using multiview bootstrapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1145–1153.

[54] Breannan Smith, Chenglei Wu, He Wen, Patrick Peluse, Yaser Sheikh, Jessica K. Hodgins, and Takaaki Shiratori. 2020. Constraining dense hand surface tracking with elasticity. *ACM Transactions on Graphics* 39, 6 (2020), 1–14.

[55] Adrian Spurr, Umar Iqbal, Pavlo Molchanov, Otmar Hilliges, and Jan Kautz. 2020. Weakly supervised 3D hand pose estimation via biomechanical constraints. In *Proceedings of the European Conference on Computer Vision*. 211–228.

[56] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[57] Ke Sun, Ting Zhao, Wei Wang, and Lei Xie. 2018. VSkin: Sensing touch gestures on surfaces of mobile devices using acoustic signals. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. 591–605.

[58] Danhang Tang, Hyung Jin Chang, Alykhan Tejani, and Tae-Kyun Kim. 2014. Latent regression forest: Structured estimation of 3D articulated hand posture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3786–3793.

[59] Danhang Tang, Jonathan Taylor, Pushmeet Kohli, Cem Keskin, Tae-Kyun Kim, and Jamie Shotton. 2015. Opening the black box: Hierarchical sampling optimization for estimating human hand pose. In *Proceedings of the IEEE International Conference on Computer Vision*. 3325–3333.

[60] Jonathan Taylor, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Toby Sharp, Eduardo Soto, David Sweeney, Julien Valentin, Benjamin Luff, Arran Topalian, Erroll Wood, Sameh Khamis, Pushmeet Kohli, Shahram Izadi, Richard Banks, Andrew Fitzgibbon, and Jamie Shotton. 2016. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Transactions on Graphics* 35, 4 (2016), 1–12.

[61] David Tse and Pramod Viswanath. 2005. *Fundamentals of Wireless Communication*. Cambridge University Press.
[62] Ruben Villegas, Jimei Yang, Duygu Ceylan, and Honglak Lee. 2018. Neural kinematic networks for unsupervised motion retargetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8639–8648.
[63] Aditya Virmani and Muhammad Shahzad. 2017. Position and orientation agnostic gesture recognition using WiFi. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. 252–264.
[64] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C. Duchi, Vittorio Murino, and Silvio Savarese. 2018. Generalizing to unseen domains via adversarial data augmentation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18)*. 5339–5349.
[65] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. 2019. Self-supervised 3D hand pose estimation through training by fitting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10853–10862.
[66] Haoran Wan, Lei Wang, Ting Zhao, Ke Sun, Shuyu Shi, Haipeng Dai, Guihai Chen, Haodong Liu, and Wei Wang. 2022. VECTOR: Velocity based temperature-field monitoring with distributed acoustic devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 3 (2022), 1–28.
[67] Anran Wang and Shyamnath Gollakota. 2019. MilliSonic: Pushing the limits of acoustic motion tracking. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–11.
[68] Dazhuo Wang, Jianfei Yang, Wei Cui, Lihua Xie, and Sumei Sun. 2022. AirFi: Empowering WiFi-based passive human gesture recognition to unseen environment via domain generalization. *IEEE Transactions on Mobile Computing* 23, 2 (2022), 1156–1168.
[69] Fei Wang, Sanping Zhou, Stanislav Panev, Jinsong Han, and Dong Huang. 2019. Person-in-WiFi: Fine-grained person perception using WiFi. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5452–5461.
[70] Lei Wang, Xiang Zhang, Yuanshuang Jiang, Yong Zhang, Chenren Xu, Ruiyang Gao, and Daqing Zhang. 2021. Watching your phone's back: Gesture recognition by sensing acoustical structure-borne propagation. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 2 (2021), 1–26.
[71] Penghao Wang, Ruobing Jiang, and Chao Liu. 2022. Amaging: Acoustic hand imaging for self-adaptive gesture recognition. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM'22)*. IEEE, 80–89.
[72] Wei Wang, Alex X. Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. 2015. Understanding and modeling of WiFi signal based human activity recognition. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. 65–76.
[73] Wei Wang, Alex X. Liu, and Ke Sun. 2016. Device-free gesture tracking using acoustic signals. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. 82–94.
[74] Yanwen Wang, Jiaxing Shen, and Yuanqing Zheng. 2020. Push the limit of acoustic gesture recognition. *IEEE Transactions on Mobile Computing* 21, 5 (2020), 1798–1811.
[75] Frank Weichert, Daniel Bachmann, Bartholomäus Rudak, and Denis Fisseler. 2013. Analysis of the accuracy and robustness of the leap motion controller. *Sensors* 13, 5 (2013), 6380–6393.
[76] Peter Welch. 1967. The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics* 15, 2 (1967), 70–73.
[77] Wikipedia. 2022. Quaternions and Spatial Rotation. Retrieved October 29, 2022 from http://en.wikipedia.org/w/index.php?title=Quaternions%20and%20spatial%20rotation&oldid=1117495750
[78] Wikipedia. 2023. Pixel 2. Retrieved January 17, 2023 from http://en.wikipedia.org/w/index.php?title=Pixel%202&oldid=1132819321
[79] Erwin Wu, Ye Yuan, Hui-Shyong Yeo, Aaron Quigley, Hideki Koike, and Kris M. Kitani. 2020. 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*. 1147–1160.
[80] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. 2015. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853* (2015).
[81] Chi Xu and Li Cheng. 2013. Efficient hand pose estimation from a single depth image. In *Proceedings of the IEEE International Conference on Computer Vision*. 3456–3462.
[82] Chenhan Xu, Bing Zhou, Gurunandan Krishnan, and Shree Nayar. 2023. AO-Finger: Hands-free fine-grained finger gesture recognition via acoustic-optic sensor fusing. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–14.
[83] Hongfei Xue, Yan Ju, Chenglin Miao, Yijiang Wang, Shiyang Wang, Aidong Zhang, and Lu Su. 2021. 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*. 269–282.
[84] Linlin Yang and Angela Yao. 2019. Disentangling latent hands for image synthesis and pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9877–9886.

[85] Sangki Yun, Yi-Chao Chen, and Lili Qiu. 2015. Turning a mobile device into a mouse in the air. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. 15–29.

[86] Sangki Yun, Yi-Chao Chen, Huihuang Zheng, Lili Qiu, and Wenguang Mao. 2017. Strata: Fine-grained acoustic-based device-free tracking. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. 15–28.

[87] Bin-Bin Zhang, Dongheng Zhang, Yadong Li, Yang Hu, and Yan Chen. 2021. Unsupervised domain adaptation for device-free gesture recognition. *arXiv preprint arXiv:2111.10602* (2021).

[88] Jie Zhang, Zhanyong Tang, Meng Li, Dingyi Fang, Petteri Nurmi, and Zheng Wang. 2018. CrossSense: Towards cross-site and large-scale WiFi sensing. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. 305–320.

[89] Xie Zhang, Chengpei Tang, Kang Yin, and Qingqian Ni. 2021. WiFi-based cross-domain gesture recognition via modified prototypical networks. *IEEE Internet of Things Journal* 9, 11 (2021), 8584–8596.

[90] Zengbin Zhang, David Chu, Xiaomeng Chen, and Thomas Moscibroda. 2012. SwordFight: Enabling a new class of phone-to-phone action games on commodity phones. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*. 1–14.

[91] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. 2018. Through-wall human pose estimation using radio signals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*.

[92] Mingmin Zhao, Yonglong Tian, Hang Zhao, Mohammad Abu Alsheikh, Tianhong Li, Rumen Hristov, Zachary Kabelac, Dina Katabi, and Antonio Torralba. 2018. RF-based 3D skeletons. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. 267–281.

[93] Yue Zheng, Yi Zhang, Kun Qian, Guidong Zhang, Yunhao Liu, Chenshu Wu, and Zheng Yang. 2019. Zero-effort cross-domain gesture recognition with Wi-Fi. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*. 313–325.

[94] Hao Zhou, Taiting Lu, Yilin Liu, Shijia Zhang, and Mahanth Gowda. 2022. Learning on the rings: Self-supervised 3D finger motion tracking using wearable sensors. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 2 (2022), 1–31.

[95] Zhipeng Zhou, Feng Wang, Jihong Yu, Ju Ren, Zhi Wang, and Wei Gong. 2022. Target-oriented semi-supervised domain adaptation for WiFi-based HAR. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM'22)*. IEEE, 420–429.

[96] Christian Zimmermann and Thomas Brox. 2017. Learning to estimate 3D hand pose from single RGB images. In *Proceedings of the IEEE International Conference on Computer Vision*. 4903–4911.

Received 26 August 2023; revised 19 March 2024; accepted 2 July 2024