RAM-Hand: Robust Acoustic Multi-Hand Pose Reconstruction Using a Microphone Array

Shiyang Wang[1], Henglin Pu[1], Wenjun Jiang[2], Qiming Cao[1], Xingchen Wang[1], Tianci Liu[1],
Zhengxin Jiang[1], Hongfei Xue[3], Lu Su[1*]
[1]Purdue University, West Lafayette, Indiana, USA; [2]Samsung Research America, Mountain View, California, USA;
[3]University of North Carolina at Charlotte, USA
[1]{wang5348, pu36, cao393, wang2930, liu3351, jiang839, lusu}@purdue.edu; [2]wenjunji@buffalo.edu;
[3]hongfei.xue@charlotte.edu

Abstract

Using 3D hand poses as the input of user interfaces can enable many novel human-computer interaction applications. However, conventional solutions for precisely reconstructing the hand poses are either vision-based, which are compute-intensive and may cause privacy issues, or wearable devices-based, which are intrusive to users. In this paper, we propose RAM-Hand, a **R**obust **A**coustic 3D **M**ulti-**Hand** pose reconstruction system built on a microphone array. Our RAM-Hand system can support multiple hands and is designed to be highly adaptable to new scenarios even when training data is limited. Specifically, it should robustly accommodate variations in environment, subject, and hand positions. To achieve this, on one hand, we propose a customized signal processing pipeline to segment multiple hands' reflections and extract the features corresponding to each hand, then feed those features into a transformer-based neural network for precise pose reconstruction. On the other hand, to tackle the challenge that the training data is limited, we propose a series of data augmentation methods to generate virtual training data, and utilize contrastive learning to ensure our model behaves well on new subjects. We conduct extensive experiments on a real-world microphone array testbed to evaluate the performance of the proposed system. The results show that our RAM-Hand system can localize each hand joint with an average error of 10.71 mm, handle multiple hands, and generalize well to the above mentioned new scenarios.

CCS Concepts

• **Human-centered computing** → **Ubiquitous and mobile computing**; **Human computer interaction (HCI)**; **Interaction techniques**.

Keywords

Acoustic Sensing, Hand Pose Reconstruction, Data Augmentation, Transformer, Contrastive learning

SenSys '25, May 6–9, 2025, Irvine, CA, USA

ACM ISBN 979-8-4007-1479-5/2025/05
https://doi.org/10.1145/3715014.3722055

ACM Reference Format:
Shiyang Wang[1], Henglin Pu[1], Wenjun Jiang[2], Qiming Cao[1], Xingchen Wang[1], Tianci Liu[1],, Zhengxin Jiang[1], Hongfei Xue[3], Lu Su[1*] . 2025. RAM-Hand: Robust Acoustic Multi-Hand Pose Reconstruction Using a Microphone Array. In *The 23rd ACM Conference on Embedded Networked Sensor Systems (SenSys '25), May 6–9, 2025, Irvine, CA, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3715014.3722055

1 Introduction

In recent years, interacting with smart devices through hand poses has proven effective in many innovative applications. When 3D hand poses are precisely reconstructed, users can control virtual objects more naturally and accurately. Hand pose recognition can benefit a broad range of applications, including VR/AR games, remote collaboration, and smart home control. Compared to traditional interfaces like keyboards and touchpads, using hands to directly drag and move content in remote meetings offers greater flexibility in organizing presentations. Similarly, grabbing an object in a video game with a hand is more user-friendly than using a mouse. Consequently, there is growing interest in developing hand pose reconstruction systems.

Currently, the primary approaches to hand pose reconstruction are vision-based methods [5, 16, 44, 56–58, 88] and wearable device-based techniques [20, 31, 46, 87]. Vision-based methods require recording video via cameras, which raises privacy concerns in many scenarios and can suffer significant performance degradation under poor lighting conditions or when occlusions occur. Additionally, the heavy computational demands of vision models limit their implementation on edge devices, which typically have restricted processing power. Wearable device-based approaches, on the other hand, require users to wear specialized gloves or wristbands, which may make the users uncomfortable.

To address those limitations, researchers are exploring the use of wireless signals to estimate hand poses [29, 36, 38], offering privacy-preserving, computationally efficient, and device-free sensing that is unaffected by lighting conditions and capable of penetrating certain occlusions. An existing work mmHand [29] proposes to reconstruct the hand pose using a mmWave Radar. However, due to poor elevation resolution and the FFT-based method it uses, it can reconstruct only one hand's poses with limited accuracy.

Recently, researchers have turned to acoustic signals for hand pose reconstruction. Compared to RF signals, acoustic signals are more ubiquitous, cost-effective, flexible to deploy, and offer high resolution. Acoustic sensing systems utilize microphones and speakers,

[*] Lu Su is the corresponding author.

which are not only low-cost but also widely integrated into everyday smart devices, minimizing deployment costs. Moreover, the flexibility of acoustic devices allows them to be easily customized to meet the requirements of diverse applications. Beyond-Voice [36] has demonstrated the potential of estimating hand poses using commercial microphone arrays. However, it also supports only a single hand and relies solely on range features, which limits its applicability and generalizability. Therefore, how to build an acoustic hand pose reconstruction system that can support multiple hands and stay robust across diverse scenarios, including varying environments, subjects, and hand positions, remains an open research question.

To achieve this goal, there are some challenges to address. As collecting acoustic sensing data is labor-intensive, the training data is typically limited and unable to cover all scenarios. Firstly, if the training data is collected from only one hand, the trained model may struggle with multi-hand scenarios, as accurately reconstructing poses with multiple hands requires precise segmentation of each hand's reflections at the feature level. This is not trivial because the hands may be close to each other and difficult to separate. Secondly, it's impractical to collect training data for every possible position and environment, making it challenging to develop a model that generalizes well to new position or environment. Finally, individual users vary in hand motion habits, hand sizes, and shapes, which can significantly impact reflection features. As a result, achieving strong performance for users not included in the training set remains a challenging task.

To address the aforementioned challenges, we propose a signal processing pipeline that effectively extracts range, angle (azimuth/elevation), and velocity features, then segments the reflections of each hand in 3D space, enabling our model to handle multiple hands. We apply the Minimum Variance Distortionless Response (MVDR) beamforming in the 3D space to improve the angular resolution in both the azimuth and elevation dimension to better segment nearby hands. To mitigate performance drops when the system is applied to new hand positions, we propose a series of data augmentation techniques to generate virtual training samples from real collected data, enhancing the system's robustness. Additionally, we apply contrastive learning to our transformer-based neural network to improve cross-subject inference performance, as this approach guides the model to focus on subject-independent features during training.

We implement our RAM-Hand (**R**obust **A**coustic 3D **M**ulti-**Hand** pose reconstruction) system on an L-shaped microphone array with 16 microphones and a speaker, conducting extensive experiments to evaluate its performance. Experimental results show that the proposed RAM-Hand system can achieve an average joint localization error of 10.71 mm when tested on 12 hand gestures. We also perform extensive testing in various scenarios not seen in the training set, with results demonstrating the robustness of our system.

We summarize the main contributions of our work as follows:

- We propose a robust acoustic hand pose reconstruction system, RAM-Hand, capable of supporting multiple hands and generalizing well to data collected across various scenarios, including unseen environments, hand positions, and subjects. To the best of our knowledge, this is the first multi-hand pose reconstruction system using wireless signals.

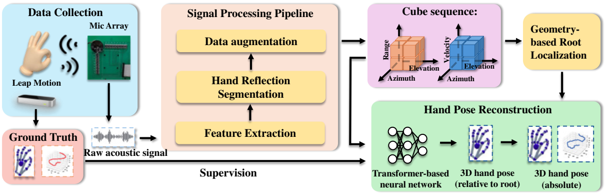


Figure 1: System Overview

- We design a customized signal processing pipeline that can extract range, angle, and velocity features from acoustic signals, and segments the reflections of each hand, enabling simultaneous reconstruction of multi-hand poses.
- To address the limitations of training data, we propose a series of data augmentation techniques to enhance the system's generalizability.
- We incorporate contrastive learning into our transformer-based neural network to overcome the challenges of cross-subject inference.
- We develop a microphone array testbed tailored for multi-hand pose reconstruction, implement our system on it, and conduct extensive experiments on real-world data to evaluate the effectiveness of our proposed system.

The remainder of this paper is organized as follows: Section 2 provides an overview of the whole RAM-Hand system. Section 3 details our signal processing pipeline and neural network design. In Section 4, we present the testbed, experimental settings, and extensive results. Section 5 reviews existing work in related areas. Finally, Section 6 offers our concluding remarks.

2 System Overview

In this paper, our proposed RAM-Hand system aims to precisely reconstruct the poses of multiple hands using acoustic signals, even the training data are collected when there is only a single hand. Our system should also remain robust when applied to new scenarios where the environments, the users, and the hands' relative locations are not seen in the training dataset, which is a common case in real-world applications. Our proposed system contains three main components: Data collection, signal processing pipeline and hand pose reconstruction, as shown in Figure 1.

2.1 Data Collection

To record high quality acoustic data for hand pose reconstruction, we build an L-shape microphone array with 16 microphones and a speaker. A Leap Motion Controller (LMC) is employed to collect precise 3D hand joint locations as the ground truth to train our neural network.

2.2 Signal Processing Pipeline

We propose a holistic signal processing pipeline to extract useful features from raw data to train a robust model. First, we employ the Minimum Variance Distortionless Response (MVDR) algorithm, an advanced super-resolution technique with high angular resolution, to extract spatial features. MVDR scans the entire 3D space, amplifies signals from specific directions, and suppresses noise from others, thereby enabling the model to extract fine-grained 3D features for each hand from overlapping signals. To address the challenge of limited training data, which hinders generalizability to new scenarios, we apply data augmentation techniques to generate virtual training samples. The generated virtual training samples expand the training dataset to new scenarios and thus improve the robustness of the trained model.

2.3 Hand Pose Reconstruction

With the collected and augmented training data, we design a transformer-based neural network to reconstruct the 3D hand poses. To address the challenge of cross-subject inference, we leverage contrastive learning to guide our model to only focus on subject-independent features to improve the performance when applied to new users not in the training set. The hand pose predicted by the deep learning model is the joint location relative to the root joint, so we further use geometric relations to localize the root joint and generate the complete hand pose.

3 Methodology

To precisely reconstruct the 3D hand poses when there are multiple hands and make the system remain robust when applied to new data collected in new environments, at new positions, and from new subjects, we first transmit an inaudible FMCW signal and record the echo,as introduced in Section 3.1. Then, we preprocess the recorded signal (Section 3.1) and extract range-azimuth-elevation features (Section 3.3) using MVDR. After that, we can detect and segment reflections corresponding to each target hand (Section 3.4), which enables our system to handle arbitrary number of hands simultaneously and avoids the interference from background reflections. When we know the position of each hand, we can extract the velocity-azimuth-elevation features of each hand (Section 3.5), which is also important for hand pose reconstruction. The above signal processing operations are shown in Figure 2. To handle the hands whose location is not in the training dataset, we apply a series of data augmentation methods (Section 3.6) to the collected training data to improve the generalizability of our system. We use both the real data and the augmented data to train a transformer-based neural network (Section 3.7) to predict the 3D hand poses for each hand. To make our system generalize well on subjects not in the training dataset, we utilize contrastive learning in our neural network design to make the model only focus on subject-independent features. Because the hand pose predicted by our model is each joint's location relative to the root joint, we use geometric relations to localize the hand as the root joint location (Section 3.8).

3.1 FMCW Signal Basics

Frequency Modulated Continuous Wave (FMCW) signal, or called chirp, is a pivotal technology in wireless sensing, renowned for its precise determination of the range, velocity, and angle of objects. We first mix the transmitted signal, $S_T(t)$ and the received signal that is reflected back from an object to the receiver $S_R(t)$ by multiplying them and applying a low-pass filter (LPF) to remove the high-frequency component, we can get the Intermediate Frequency (IF) signal, denoted as $s_{IF}(t)$:

$$s_{IF}(t) = \frac{\alpha}{2}\cos\left(4\pi f_{\min}\frac{d}{c} + \pi K\left(\frac{4dt}{c} - \frac{4d^2}{c^2}\right)\right), \quad (1)$$

where f_{min} is the chirp start frequency and K is the chirp slope, α denotes the path loss and c denotes the speed of wave. d is the distance between the radar and target. The traditional method to process FMCW signal is applying range, Doppler and angle FFT to the IF signal [21]. To differentiate received signal components reflected from various ranges, a Range-FFT operation is applied to the samples of $s_{IF}(t)$ within a chirp for signal separation. This operation transforms the frequency spectrum of $s_{IF}(t)$ into the range spectrum. To additionally determine the target's velocity, we can performing an additional FFT operation, known as Doppler-FFT, along the dimension of different chirps. Except for the range, the angle of the reflection signal relative to the sensing device is also important to localize the object. The angle can be estimated by utilizing signals from multiple receivers in an array. By applying a third FFT operation along the receiver dimension of the signals, known as Angle-FFT, the sensing system can generate a Range-Angle spectrum then detect the object's position.

Using the above mentioned methods, the sensing resolution of FMCW signal in terms of range, velocity, and angle will be $\frac{c}{2B}$, $\frac{\lambda}{2T_f}$, and $\frac{2}{N}$, respectively. We can see that the range resolution is dicided by the bandwidth of the signal and the angular resolution is decided by the number of the microphones in the array. Therefore, our system plays an inaudible 16 - 22 kHz FMCW signal, which is almost inaudible to most people while has enough bandwidth. We apply a hanning window to the FMCW signal to improve the SNR and avoid the audible sound leakage [33]. To enhance the angular estimation, we design an L-shaped microphone array with 16 microphones to record the echo. This microphone array consists of two 8-microphone linear sub-arrays which are perpendicular to each other, so it can capture the angle features in both azimuth and elevation. Unlike a traditional rectangular or square array—which would require a number of microphones that increases quadratically with the resolution in the azimuth and elevation dimensions, the L-shaped design offers a fine-grained angular resolution while keeping the total required number of microphones relatively low, thereby balancing the computational cost.

3.2 Data Pre-processing

To remove noise, calibrate offset caused by system delay, we need to pre-process the data before extracting the features.

3.2.1 System Delay Calibration.

Utilizing FMCW signals to estimate range information requires precise synchronization between the speakers and the microphones. Therefore, it's necessary to calibrate the delay caused by the hardware system. However, the system delay is not a constant and it can vary each time the system restarts. To accurately calibrate this delay, we utilize the following fact: the LoS (line-of-sight) signal has much stronger amplitude than all the reflected signals because the signal goes from the speaker directly to the microphones, and the distance between the speaker and each microphone is fixed. Therefore, we can use the LoS reflection as an anchor to calibrate the system delay. Specifically, we assume that the speaker will start playing the signal slightly later than the time that the microphones begin recording, which means that the estimated ToF (time-of-flight) will contain the system delay. Thus, we can first mix the transmitted and received signal to generate the IF signal to the first few chirps then apply range FFT [21] on them. We localize the frequency bin with the highest energy and derive the corresponding distance. In our microphone array, the distances between the speaker and microphones are fixed, so we can derive the system delay by subtracting the propagation distance of the LoS path from the measured distance and dividing the result by the sound speed. After that, we calculate the number of delayed samples by multiplying the time delay and the sampling rate. Finally, we can shift the raw acoustic data by the delayed samples to calibrate the system delay for the following signals. This calibration process can be done automatically at the beginning of each recording and only takes tens of milliseconds, which can be easily implemented in real-world applications.

3.2.2 Background Reflection Removal. Once the system delay is calibrated, we can apply the mixing and range FFT on the shifted data to extract the distance information. Then we take the differential of each chirp's range FFT results to remove the static background reflection. Specifically, we calculate the difference between the current chirp-length data with the previous chirp to filter out the background reflections that never change. Then we can find a continuous subset of the range bins that includes all the high energy bins as the range of interest, and apply the following feature extraction algorithms only on those bins, which can significantly improve the computational efficiency.

3.3 Range-Azimuth-Elevation Cube Construction

To precisely capture features of the hands, we need to first localize and segment the reflection from each hand in the 3D space. To achieve this, we need to extract the range, azimuth and elevation features of the hand reflection. A simple idea is to apply angle FFT on the horizontal and vertical sub-array, respectively. However, this method has two limitations: First, this method consider the two sub-arrays as independent components and estimate the angle information separately, even if we know the locations of all the microphones. The horizontal sub-array only extracts azimuth features and the vertical sub-array only extracts the elevation. Therefore, we will get a range-azimuth feature map and a range-elevation feature map. Although we can localize and segment the hand reflections in both feature maps, it may be hard to match the reflections segmented from two feature maps when there are multiple hands, especially when the hands are in similar ranges. Second, applying angle FFT on a 8-microphone sub-array results in an angle resolution of about 15°, which is still insufficient to capture fine-grained spatial features. Thus, it is challenging to separate hands that are in close proximity.

To address the above limitations, we propose to apply the Minimum Variance Distortionless Response (MVDR) beamforming [8] on all the 16 microphones to further enhance angular resolution, thereby facilitating improved segmentation of reflections from closely spaced hands. MVDR is an adaptive spatial filtering technique with the basic idea to maximize the gain in a desired direction while minimizing interference and noise from other directions. Specifically, assume that we know the 3D positions of the M microphones in our array $\mathbf{r}_m = \left[x_m, y_m, z_m\right]$, $m = 1, \ldots, M$, then the MVDR weights for a desired azimuth θ and elevation ϕ can be calculated by:

$$\mathbf{w}_{\mathrm{MVDR}} = \frac{\mathbf{R}^{-1}\mathbf{a}(\theta,\phi)}{\mathbf{a}(\theta,\phi)^{\mathrm{H}}\mathbf{R}^{-1}\mathbf{a}(\theta,\phi)}, \tag{2}$$

Where $\mathbf{R}$ is the covariance matrix of the received signals $\mathbf{x}(t)$, defined as $\mathbf{R} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{x}(t)\mathbf{x}(t)^{\mathrm{H}}$, and H means Hermitian transpose. The steering vector $\mathbf{a}$ for azimuth θ and elevation ϕ is:

$$\mathbf{a}(\theta,\phi) = \left[e^{j\frac{2\pi}{\lambda}\mathbf{r}_1^{\mathrm{T}}\mathbf{d}(\theta,\phi)}, e^{j\frac{2\pi}{\lambda}\mathbf{r}_2^{\mathrm{T}}\mathbf{d}(\theta,\phi)}, \ldots, e^{j\frac{2\pi}{\lambda}\mathbf{r}_M^{\mathrm{T}}\mathbf{d}(\theta,\phi)}\right]^{\mathrm{T}}, \tag{3}$$

Where $\mathbf{d}(\theta,\phi) = \left[\cos(\theta)\cos(\phi), \sin(\theta)\cos(\phi), \sin(\phi)\right]$. Then the output power of the MVDR can be computed as:

$$P_{\mathrm{MVDR}}(\theta,\phi) = \frac{1}{\mathbf{a}(\theta,\phi)^{\mathrm{H}}\mathbf{R}^{-1}\mathbf{a}(\theta,\phi)}, \tag{4}$$

Then we apply MVDR on each range bin of the range FFT results within this frame to generate a range-azimuth-elevation cube, which represents the reflection energy distribution in the whole 3D space. In this cube, reflections from multiple hands can be segmented without any requirement for matching the azimuth reflections and elevation reflections. Furthermore, MVDR is a super-resolution method that can provide higher angle resolution, thanks to its ability to suppress interference and noise more effectively than angle FFT. Thus, it's capable of resolving signals reflected from closely spaced hands.

3.4 Hand Reflection Segmentation

After we generated the range-azimuth-elevation cubes, we need to segment the reflection of each hand. We segment the reflections from the generated cubes by the following methods:

We first use the **CFAR** (Constant False Alarm Rate) algorithm [54] to detect the range, azimuth, elevation bins with high energy, which means there are reflections at those locations. CFAR adjusts the threshold of detection dynamically to maintain a constant rate of false alarms, which enables it to handle varying noise levels and provide stable detection results. Specifically, it applies a sliding window to the cube to estimate the noise level, then the bins that exhibit energy exceeding a noise threshold are preserved and identified as containing potential reflections.

After that, we use **DBSCAN** [12] (Density-Based Spatial Clustering of Applications with Noise) to cluster the areas with high energy and calculate the center of each cluster. DBSCAN can identify clusters of arbitrary shapes by examining the spatial density of data points, which is suitable to cluster the dynamic hand reflections in our system. Then each cluster can be considered as the reflections of one hand and the center represents its location. In case sometimes the body movements also generate clusters that are not expected, we can filter out the clusters that have much larger sizes than a hand. Then we can segment a fixed-sized sub-cube around the center of each cluster for further feature extraction and hand pose reconstruction.

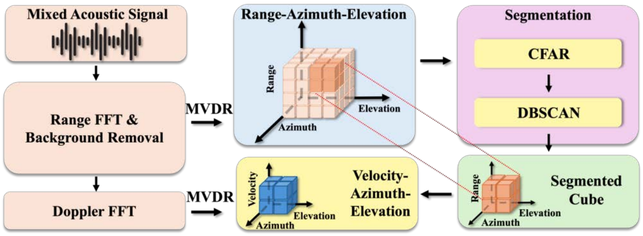


Figure 2: Signal Processing Pipeline. We extract 3D range-azimuth-elevation features then localize the hands and segment a sub-cube around each hand from the whole range-azimuth-elevation cube. Then we also extract each hand's velocity features and concatenate the range-azimuth-elevation and velocity-azimuth-elevation sub-cubes as the training data of the hand pose reconstruction model.

3.5 Velocity-Azimuth-Elevation Cube Construction

After the reflection of each hand is localized and segmented, we can start to extract the features of each hand. Except for the spatial features including range, azimuth, elevation that we have already extracted, velocity features are also important for precisely reconstructing hand poses because the hand will generate specific velocity patterns when making different hand poses. Therefore, we first apply Doppler FFT on the results of the range FFT along the chirp dimension to extract the velocity features. Then for each velocity bin, we search in azimuth and elevation dimension and calculate the MVDR power to build a Doppler-azimuth-elevation cube. In order to reduce the computational complexity, we only apply MVDR in the area of each sub-range-azimuth-elevation cube segmented in Section 3.4. Finally, to combine the spatial features and the velocity features, we concatenate the range-azimuth-elevation and velocity-azimuth-elevation cubes together and feed them to the deep learning model for hand pose reconstruction.

3.6 Data Augmentation

Training a deep learning model with good generalizability requires a large amount of training data. However, collecting acoustic sensing data is inherently complex and labor-intensive, so usually we only have limited amount of data during training, which can result in overfitting and the performance will severely degrade when applied to real-world scenarios where the environment, the user and the hand's relative location to the microphone array are not seen in the training dataset. Therefore, we propose a series of data augmentation techniques designed to synthesize virtual training data from the limited real data, which are introduced in detail below.

Add Random Noise: A straightforward method to enhance the robustness of our trained deep learning model is to add random noise to the recorded acoustic signals. By integrating these noise-augmented data into the training dataset, the model is encouraged to learn more generalized features.

Augmentation for different ranges: To handle hands in unseen positions, we can approximately infer the virtual features in different positions from the feature cubes generated using limited real training data. In the generated cubes, the spatial information is represented by range, azimuth, and elevation. Considering that objects of the same size can occupy different numbers of bins in the azimuth and the elevation dimension, we propose an augmentation method that scales the segmented data cube in both the azimuth and the elevation dimension to simulate the impact of different distances. Specifically, given a segmented data cube generated from real data with N bins in the azimuth-elevation dimension, the distance between the hand and the microphone array is d, and the distance of the virtual hand features that we want to generate is d', we traverse every element in this cube and rewrite its azimuth/elevation index i as $i' = round(\frac{N+1}{2} + (\frac{d}{d'}) \times (i - \frac{N+1}{2}))$. If there are multiple elements assigned to the same index in the augmented data cube, we will take the average of those values. If there are some elements in the augmented data cube not assigned any value, we will fill in those gaps with linear interpolation. This method will be applied to both the range-azimuth-elevation cube and velocity-azimuth-elevation cubes.

Augmentation for different angles: When the hands are at different angles (azimuth/elevation) relative to the microphone array, the measured velocity patterns can vary. The reason is that only the movement along the radial direction can cause the change of a reflection path length, which has been analyzed in previous works [51, 86]. Therefore, we need to apply augmentation on the velocity dimension of the velocity-azimuth-elevation cube to mitigate the challenge when applying our model to scenarios where the hands are at different angles. To this end, we generate virtual features that simulate the impact of angles from the real data collected directly in front of the microphone array. For each angle that we want to generate augmented data, we scale the velocity dimension by calculating the radial projection of each element in the velocity dimension. For an element whose velocity is v, we can calculate its projection on the new radial direction when the azimuth and elevation angle changes to θ and ϕ: $v_{new} = vcos(\phi)cos(\theta)$. Then we assign the value of the element to the corresponding velocity bin in the augmented data cube. For the empty bins, we can fill them with random noise.

Augmentation for inaccurate segmentation: When segmenting the hand reflections from the whole range-azimuth-elevation cube, sometimes the calculated center position can be inaccurate, which means that the center of the segmented features can drift to a nearby position in the data cube. To handle this problem and make our system robust to the segmentation error, we manually shift the position of the center around the estimated one to generate more segmented features.

3.7 Neural Network

The design of our transformer-based neural network is shown in Figure 3. The details are explained below:

3.7.1 Transformer Design. Originally developed for Natural Language Processing tasks, transformers [62] have demonstrated exceptional performance on many other tasks, such as image processing [11], speech recognition [26], video classification [3] and wireless sensing [74]. Since our extracted features are a series of temporal sequences of range-velocity-angle cubes, which are similar to video frames, we build our neural network on TimeSformer [3], which extends the standard transformer architecture by incorporating a

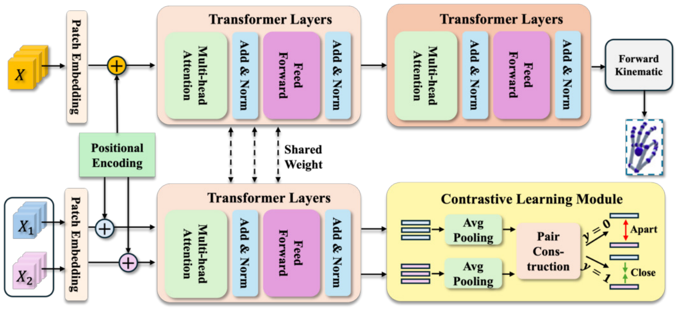


Figure 3: Neural Network Architecture. X is a training sample that includes range/velocity-azimuth-elevation features and is used to train the hand pose reconstruction model. X_1 and X_2 are a pair of training samples randomly selected from the training set, used as the input for the contrastive learning module. Specifically, if X_1 and X_2 are from the same gesture but different subjects, they are considered as a positive pair. And if they are from different gestures, we consider them as a negative pair.

space-time attention mechanism where the self-attention is calculated in spatial dimension and temporal dimension alternately. The spatial attention handles interactions within individual frames, while the temporal attention across different frames captures the dynamics over time. To be specific, in the segmented range/velocity-azimuth-elevation cube, the range/velocity dimension is considered as channels and the azimuth-elevation dimensions are treated as the height and width of an frame of image. The cubes will be segmented into patches following the method in ViT [11] and they will be mapped into embedding vectors $\mathbf{z}^{(0)}$. In each attention layer l, we compute the query/key/value for multiple attention heads from last layer's representation $\mathbf{z}^{(l-1)}$ then apply temporal attention and spatial attention alternately. After that, we can obtain the output encoding of layer l by calculating the weighted sum of value vectors with both the spatial and temporal attention coefficients for each attention head and aggregate the vectors from all heads with an MLP. One difference in our model with the original TimeSformer design is that we didn't use a classification token because we want to predict the hand pose in each frame instead of the class label. Therefore, we directly use a linear layer to project the final layer's output encoding of each frame to the hand pose.

However, directly predicting the 3D locations of each hand joint independently doesn't utilize the prior knowledge of the hand, which can make the trained model unstable. Because the hand joints can be considered as a tree structure with N joints according to the connection relationship between the joints, we can make use of the prior knowledge of the hand by estimating the rotation of each connected hand segment instead of the location, then applying forward kinematics [24, 63] to generate the locations of hand joints from the root joint to the fingertips.

Specifically, for joint p^i, its 3D coordinate can be derived by: given the location of its parent joint $p^{parent(i)}$ and the initial position of p^i, $p^{parent(i)}$:

$$p^i = p^{parent(i)} + R^i(p_0^i - p_0^{parent(i)}), \tag{5}$$

where $p^{parent(i)}$ is the location of p^i's parent joint, R^i is the rotation matrix of the joint p^i with respect to its parent, p_0^i and $p_0^{parent(i)}$ are the initial position of p^i, $p^{parent(i)}$, respectively. By doing this, we encode the restriction of the hand skeleton knowledge into our model so that the generated hand pose will be more realistic. We integrate this process as a forward kinematic layer in our neural network.

3.7.2 Contrastive learning module. Although we have generated augmented data for different positions and our signal processing makes our model robust to different environment and multiple hands, different subjects still have different habits for hand gestures and their hand shapes also vary. Therefore, it's still challenging to achieve cross-subject inference with high accuracy. To overcome this challenge, we propose to integrate contrastive learning into our model to make our model focus on the subject-independent features during training. More precisely, contrastive learning aims to learn effective representations by contrasting positive pairs (similar or related samples) against negative pairs (dissimilar or unrelated samples). In our case, we can build additional paired training data which consists of two training samples from different recorded acoustic data. If the two samples are from different subjects but the same gesture, we define them as a positive pair. If those samples are from different gestures, we define them as a negative pair. During training, our model will take both the paired data and unpaired data as the input. The unpaired training data X will be used to update the parameters of the whole neural network for hand pose reconstruction, and the two samples X_1 and X_2 in a data pair will be used to guide the neural network focuses on subject-independent features. This is achieved by considering the first few layers of the transformer as a feature extractor, then we feed X_1 and X_2 into those layers separately and get two features F_1 and F_2 respectively. We can calculate contrastive loss on F_1 and F_2 to guide the feature extractor part of the model map the input acoustic data cubes into a features space where different subjects's representations are close to each other. This method is inspired by the siamese network [4], which has been widely used in contrastive learning. But in our design, the unpaired data X and paired data X_1 and X_2 will all be processed by the first few layers in our model with shared weights. Therefore, the representations extracted from X are not obviously affected by different subjects, either. Then the following layers of the transformer will be trained on subject-independent features which makes our model retains good performance in cross-subject inference.

3.7.3 Loss Function. In our paper, the following loss functions are used to train our deep learning model:

First, we use a *pose loss* to minimize the mean square error between the predicted 3D positions of hand joints with the ground truth pose:

$$L_p = \frac{1}{T}\sum_{i=1}^{T}\frac{1}{N}\sum_{i=1}^{N}\|\hat{p}_t^i - p_t^i\|_2, \tag{6}$$

where $\hat{p}_t^i$ is the predicted location of the i-th hand joint at time slot t, and p_t^i is the corresponding ground truth location, N is the total number of hand joints defined in our hand skeleton model, and T is the total number of segmented range-velocity-angle cubes.

Then, a *smooth loss* is applied:

$$L_s = \frac{1}{T-1}\sum_{t=2}^{T}\frac{1}{N}\sum_{i=1}^{N}\|(\hat{p}_t^i - \hat{p}_{t-1}^i) - (p_t^i - p_{t-1}^i)\|_H, \tag{7}$$

where $\|\cdot\|_H$ is the Huber norm. This loss is adopted because we expect that the generated hand poses exhibit consistency across consecutive time slots, which means that sudden changes for the joint positions should be avoided. This is necessary to make the generated poses look realistic.

The third loss function in our system is the contrastive loss:

$$\begin{aligned} L_c = {} & (1-y)\cdot\frac{1}{2}\left(\|F_1 - F_2\|\right)^2 \\ & + y\cdot\frac{1}{2}\left(\max(0, m - \|F_1 - F_2\|)\right)^2, \end{aligned} \tag{8}$$

In this loss function, y is the label of the training sample, defined as 1 for a positive pair and 0 for a negative pair. F_1 and F_2 are the features extracted by the first few layers of the transformer from paired training data X_1 and X_2, respectively.

Then we can summarize the total loss L as:

$$L = L_p + \beta\cdot L_s + \gamma\cdot L_c, \tag{9}$$

where β, and γ are the hyper-parameters to adjust the weight of smooth loss and the contrastive loss.

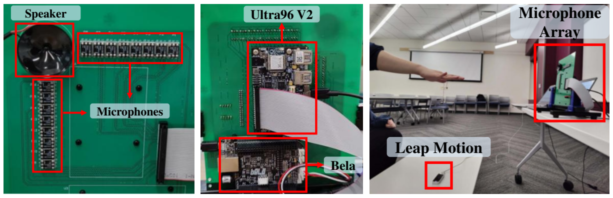
3.8 Root Joint Localization

The hand pose predicted by our deep learning model is each joint's location relative to the root joint (the wrist). However, the location of the root joint is also important in many scenarios, such as grabing an object and move it in a VR game. Therefore, we add another module that predicts the 3D root joint location of each hand. Although it's hard to differentiate the wrist reflection with the reflections from the other part of the hand due to the small size of the hand, we notice that we mainly focus on the translation of the root joint in many applications. Then we can track the location of the strongest hand reflections, which usually appears at the palm and has a relatively stable offset compared to the wrist location, as the root joint location. Thanks to the high quality spatial features extracted by MVDR, we can pick the elements with highest energy in the range-azimuth-elevation cube as the root joint reflection and use the following geometric relations to get 3D coordinates of the root joint: $x = R\cos(\theta)\cos(\phi)$,$y = R\sin(\theta)\cos(\phi)$,$z = R\sin(\phi)$. After that, we use an extended Kalman filter [53] to smooth the root joint trajectory.

4 Experiments

4.1 Testbed

4.1.1 Microphone Array. In order to capture high quality acoustic signals and extract range, velocity, azimuth and elevation information, we design a microphone array with 16 microphones and one speaker as our testbed. As introduced in Section 3.1, the angle resolution is restricted by the number of microphones, which should be considered when designing the testbed. Although we are using a super-resolution method, MVDR, to estimate the angle, the angle estimation will be more accurate as the number of microphones in the array increases. Due to the fact that precisely reconstructing the hand poses requires 3D spatial information, we need to have high accuracy in both azimuth and elevation. Therefore, we design an L-shaped microphone array consisting of 2 linear sub-arrays that are perpendicular to each other, as shown in Figure 4a. The horizontal sub-array is used for estimating the azimuth and the vertical sub-array is for the elevation. In each sub-array, there are 8 microphones [61] with 8.5 mm (half wavelength for 20 kHz acoustic signal) spacing between adjacent microphones. The number of microphones for both azimuth and elevation estimation is larger than the mainstream commercial microphone array such as MiniDSP [43] and ReSpeaker [59], and maintains a balance in terms of data throughput. To transmit the acoustic FMCW signal, a speaker is placed at the intersection of the two sub-arrays.



(a) front **(b) back** **(c) Experiment Setup**

Figure 4: Testbed Setup

We use a Ultra96-V2 [1] FPGA to control the 16 microphones to record acoustic signals simultaneously. However, Ultra96-V2 board doesn't have an audio processing circuit so the quality of the output signal will be limited if we use it to control the speaker playing audio signals. Therefore, we connect the speaker to a Bela platform [2], which is designed for high-performance audio processing, to play the acoustic signal. To synchronize the speaker and the microphones, we can add a control logic between the Ultra96-V2 board and the Bela board by sending a control signal from Ultra96v2 board to Bela when the microphones start recording. Once received this signal, the speaker will start playing the signal. Due to the low latency of both boards, we can make sure that the system delay can be calibrated well using the method in Section 3.2.1. All the aforementioned components are placed on a PCB (printed circuit board) as shown in Figure 4a and 4b. It is worth noting that although our system is implemented on a L-shape array for a better angular resolution, it can also be deployed to any other commercialized microphone arrays, because the MVDR-based 3D spatial feature extraction pipeline in our paper can be applied to any microphone arrays as long as we know the location of each microphone. We also conduct experiments to demonstrate our system can still achieve high performance with less number of microphones in Section 4.4.3, which shows that the proposed method is compatible with commercial microphone array.

4.1.2 Leap Motion Controller. We use a Leap Motion Controller (LMC) to collect the ground truth of the 3D hand poses. The LMC is equipped with two Infrared Light (IR) cameras and three IR emitters to obtain the precise 3D locations of each joint of the hands in Cartesian space. According to [72], the LMC can estimate the position of hand joints with errors less than 1.2 mm. During the experiments, we connect it to a laptop and obtain the ground truth poses through the official Application Programming Interface (API) of LMC. The sampling rate is set to 100 Hz.

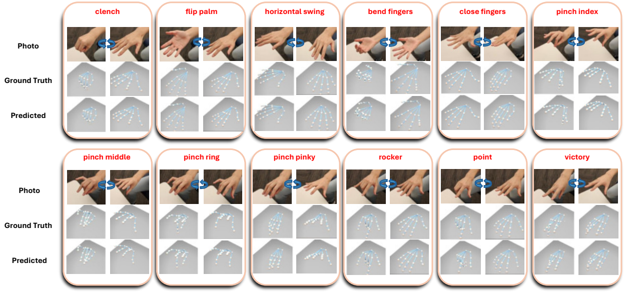


Figure 5: Hand Gestures Visualization.

4.2 Data Collection and Prepossessing

We recruited 8 volunteers as human subjects in our experiments. We designed 12 hand gestures to evaluate the hand pose reconstruction performance of our RAM-Hand system, including: clench, flip palm, horizontal swing, bend fingers, close fingers, pinch index, pinch middle, pinch ring, pinch pinky, rocker, point, and victory. The hand gestures are visualized in Figure 5. Each subject was requested to perform every hand gesture for 60 seconds with their right and left hands, respectively. They also provide 60 seconds of random hand gesture combination data. We collected the subjects' data in 3 rooms, 4 different distances and 7 different angles (azimuth or elevation) to evaluate the robustness of our system. To prove that the proposed system can handle multiple simultaneous existing hands even if there is single-hand data in the training set, we also asked the subjects to perform each gesture for an additional 60 seconds with their left hands and both hands, respectively. All the experiments have been proved by our institution's IRB. The setting of the data collection is shown in Figure 4c. The microphone array is fixed by a stand on a table to play and record acoustic signals, and a leap motion controller is placed under the subjects' hands to collect the ground truth of the hand poses. The transmitted FMCW signal has a frequency rate of $16 - 22$ kHz and a sampling rate of 44.1 kHz. The length of a single chirp is 20 ms and there is no gap between adjacent chirps. The frame rate of the ground truth collection is set to 100 Hz but down-sampled to 10 Hz later.

We first calibrate the system delay as mentioned in Section 3.2.1, which can be done automatically using the first few chirps within tens of milliseconds. Then we dechirp the transmitted signal and the received signal by multiplying them together and applying a range FFT to extract the range features. After that, we calculate the difference between the range FFT results of adjacent chirps to remove the static background reflections. Because the chirp length is 20 ms, we have 3000 chirps in a 60-second recording. We then apply MVDR on each range bin of the range FFT results to construct the range-azimuth-elevation cube, with the search scope from 40° to 140° for azimuth and from $-50°$ to 50° for elevation. The search step is set to 2° to balance the accuracy and the computational cost, which results in 51 bins in both azimuth and elevation dimensions.

To segment the reflections of each hand from the cubes, we first apply the CFAR algorithm to the range-azimuth-elevation cube to detect the elements with high energy as the positions where there are reflections. We can use a threshold to further filter out points with low energy after CFAR. Then we use DBSCAN to cluster those detected points into clusters to distinguish different hands. The eps value is set to 2, and the minimum number of points is set to 10. In each frame, we filter out the clusters unlikely to be a hand by checking their sizes and then extract a sub-cube around each cluster center with a fixed size from the cube. The size of the segmented cube is set to $(7, 7, 7)$ in range, azimuth, and elevation dimensions, respectively.

We then apply Doppler FFT on the range FFT results with motions to extract the velocity features. The window size for one Doppler FFT is set to 25 chirps and the step size is 5. Therefore, the generated range-Doppler profiles has a frame rate of 10 Hz. We construct a Doppler-azimuth-elevation cube by applying MVDR on the Doppler FFT result and set the search scope to $(C_a - 12°, C_a + 12°)$ in azimuth dimension and $(C_e - 12°, C_e + 12°)$ in the elevation dimension, where C_a and C_e are the azimuth and elevation of the hand reflection's center obtained in the segmentation process. Due to the small search scope, we can use a finer-grained search step value, 1°, in this step, which creates a Doppler-azimuth-elevation cube with shape $(25, 25, 25)$ for each hand. After that, we can regenerate the range-azimuth-elevation cube for each hand using the same search scope and step and concatenate both cube in the range/velocity dimension as the feature fed into the deep learning model for hand pose reconstruction.

We propose a series of data augmentation methods to generate virtual training samples from the real collected training data, as introduced in Section 3.6. We add random noise with SNR $10dB$, $15dB$, $20dB$, $25dB$ and $30dB$, respectively, which generates 5× virtual data. In addition, augmented data for distance in 30 cm, 70 cm, 90 cm, 110 cm, 130 cm, and 150 cm, is generated from the data collected in 50 cm and results in 6× more data. We creates virtual features for azimuth within the scope $(45°, 135°)$ and the scope for elevation is $(-45°, 45°)$. The step is set to 15°, so there is 36× virtual data. Finally, we will shift the location of each hand with offset $(-1, 1)$ bins in range, azimuth and elevation dimension, which generates 26× augmented data. Overall, the data augmentation can provide over 70× virtual data.

4.3 Model Setting and Model Training

We build our transformer model with the space-time attention layers proposed in [3]. In each layer, we take a sequence with 16 frames of segmented cubes as the input, and the patch size is set to 5, the number of attention heads is 10, the dropout rate for attention computation is 0.15 and the activation function is Softmax. We have 12 attention layers in total and the first 4 layers are also trained with contrastive loss to extract subject-independent features. The weight hyper-parameters β, γ in the loss function are both set to 0.1, and we use an Adam optimizer with learning rate 0.0001 to update the parameters of our model. The batch size is set to 32. For the training data in each recording session, the input cubes are segmented into fixed-length sequences using a overlapped sliding windows with size 16 and step 1 as the final input of the neural network. For the testing data with an arbitrary length of cube sequence, we can segment it into non-overlapped windows with length 16 during inference. The neural network is implemented in Pytorch [48].

4.4 Experiment Results

To comprehensively evaluate the performance of our proposed system, we conduct extensive experiments in different scenarios

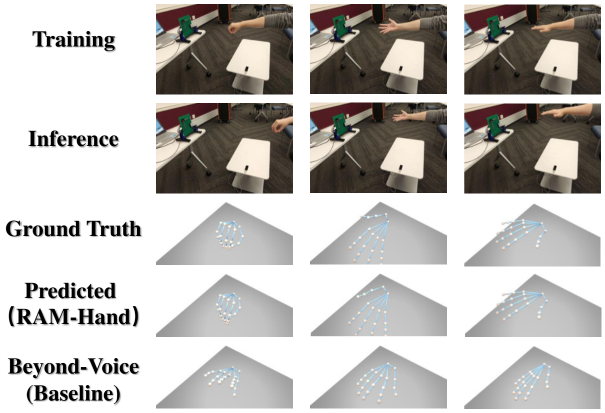


Figure 6: Examples of the Reconstructed Hand Poses in Unseen Location.

and compare the performance with the baseline method. We first evaluate the hand pose reconstruction accuracy in the basic scenario where the testing data is collected in the same setting as the training data. Then we show that our system is robust to testing scenarios where the hand positions, environments, subjects, and number of hands are new and not seen in the training set. We conduct additional experiments to measure the root joint tracking accuracy. We also evaluate the running time of our system in this section.

4.4.1 *Baseline.* We choose Beyond-Voice [36] as the baseline method. It extracts range profiles by calculating the cross-correlation between the transmitted and received signal, then feed those features into a CNN-LSTM model to reconstruct the hand poses. We implement it on our dataset and compare its performance with our system in the following experiments.

4.4.2 *Metrics.* Since we are using a learning-based method to reconstruct the hand pose and a physical model to localize the root joint, we measure their performance separately. For hand pose reconstruction, we first align the root joint in the predicted hand pose and the ground truth, then use the **Average Joint Localization Error**, which is defined as the average Euclidean distance between the 3D locations of each hand joint of the predicted hand poses and the ground truths as the quantitative metric. For root joint localization, we only measure the Euclidean distance between the predicted root joint location with the ground truth.

4.4.3 *Basic Scenario.* We define the setting of our basic scenario as: following the description in Section 4.2, the subjects perform each of the 12 hand gestures with their right hands and left hands, respectively in a conference room. The distance between the hands and the microphone array is set to 50 cm and the azimuth and the elevation angle of the hands relative to the array are both 0 degree. For each subject and each gesture, 80% (the first 48 seconds) of the data are used for training and the remaining 20% of the data are used for testing.

In the basic scenario, the average joint localization error is 10.71 mm, which is low enough to accurately capture the hand poses. We also plot the predicted hand poses and the corresponding ground truth poses in Figure 5 as qualitative results. In this figure, we can see that the reconstructed poses looks very similar to the ground truth for all the gestures. We also evaluate the accuracy of the model trained with 12 gestures' data on the random gesture combination data, which gives an error of 17.25 mm. Note that we didn't include any data from the random combination session in the training set, so the degradation in performance is very slight for this setting. As a comparison, the results of the baseline method, Beyond-Voice, in the above two experiments are 16.2 mm and 21.72 mm, respectively. Our proposed method has better performance because we extract extensive features including range, azimuth, elevation and velocity while Beyond-Voice utilizes only the range information.

Table 1: Sensitivity Analysis Results

Parameter	Value	Result (mm)
MVDR Step Size	**1**	10.71
	2	12.04
	3	12.91
Transformer Layer Number	**8**	11.47
	9	11.13
	10	10.84
	11	10.69
	12	10.71
Microphone Number	**10**	12.77
	12	12.39
	14	11.85
	16	10.71

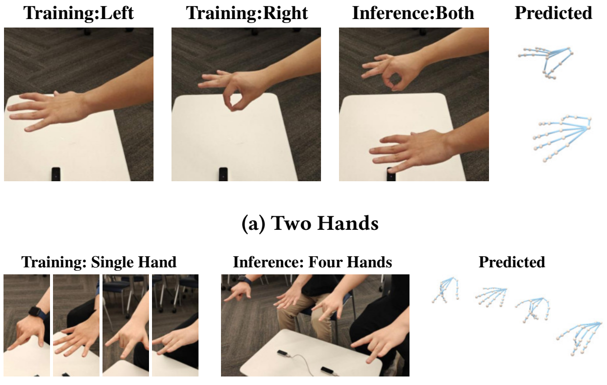
Sensitivity Analysis: We analyze if our proposed system is sensitive to different parameters and hardware configuration, such as the searching step size of MVDR, the number of layers of the transformer model, and the number of microphones. The results are shown in Table 1.

For the MVDR searching step, we directly use the sub-cube segmented from spatial features generated with step size 2° and 3° to train our model, and compare the results with the default step size, 1°. The joint localization error only increases slightly from 10.71 mm to 12.91, even if we use 3° as the step size. Thus, our method can work well with varying MVDR searching step sizes.

We also evaluate whether our deep learning model is sensitive to the number of transformer layers. During the experiments, we fix the number of layers used for contrastive learning as 4, but use less subsequent layers to change the total layer number. The results show that the accuracy remains high even if we only use 8 transformer layers, which means that our method has the potential to further reduce the required computational resources.

Finally, considering that there are many commercial microphone arrays with different number of microphones, we conduct experiments to test if our method is sensitive to different microphone numbers, by applying MVDR on data from only part of the microphones. Due to the L-shaped design of our testbed, we measure the results of using 5, 6, and 7 microphones in each sub-array, so the total numbers of microphones are 10, 12, and 14. From the results shown in Table 1, our method maintains excellent performance even the number of microphones reduces to 10. Thence, our proposed system also has the potential to be implemented on other commercial microphone arrays.

Based on the above sensitivity analysis, we can conclude that our method maintains high performance in a broad parameter range and different hardware settings, which improves its practicality in different real-world scenarios.



(a) Two Hands

(b) Four Hands

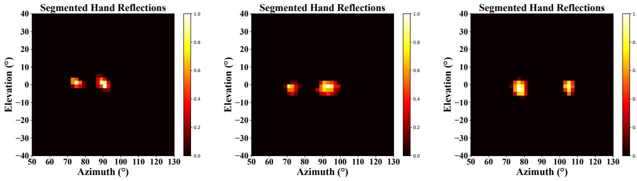
Figure 7: Example of Multi-Hand Pose Reconstruction

4.4.4 Multi-Hand Pose Reconstruction. When there are multiple hands performing gestures at the same time, our system can also work well even there is only single-hand data in the training set because it can segment reflections from each hand. We conduct experiments to demonstrate our system's effectiveness to handle multiple hands.

We collect 12 sessions of data from each subject, and the subjects are asked to make gestures with both hands for 60 seconds in each session. Their left hands and right hands can perform the same gestures or different gestures. The two hands are separated about 30° in azimuth angle (the distance is about 25 cm) to avoid mutual occlusion. Then we train our model with data collected when the subjects are using a single left hand and a single right hand only. We evaluate the accuracy of this model on the two-hands data. The result shows that accuracy of our system can achieve 19.48 mm for the right hands and 20.83 mm for the left hands. An example of the qualitative result is shown in Figure 7a.

To evaluate the minimum required hand separation distance, the subjects use both hands to perform gestures with varying hand separation distances. We plot the features after segmentation in Figure 8. The hand separation distance is defined as the distance between the centers of the two palms. We can see that even when the hands are as close as 15 cm, their features can still be segmented into different clusters well.

We also conduct experiments to evaluate the hand reconstruction performance when there are more hands, which involves the second subject. We ask two subjects to sit in front of the microphone array and perform hand gestures with their hands and make sure that every two hands are at least 20 cm away from each other. The average accuracies for each hand are 23.01 mm, 21.56 mm, and 24.26 mm in the three-hands scenario, from left to right. In the four-hands scenario, the accuracies are 22.81 mm, 21.62 mm, 25.42 mm, and 28.37 mm. And it's visualized in Figure 7b. According to those results, we can conclude that when there are more hands, the average accuracy drops slightly due to the stronger interference between those hands, but the performance is still acceptable. Although our current method can't handle the occluded hands, the users' hands are usually occluded when two hands are crossing in real-world scenarios, which only causes short-time occlusion. Therefore, it's possible to leverage information from both preceding and subsequent frames to infer the occluded hand poses.



(a) 15cm

(b) 20cm

(c) 25cm

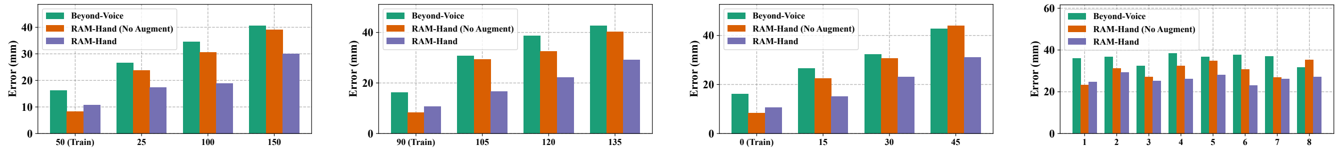
Figure 8: Hand Separation at Different Distances.

From the above experimental results, we can say that our system can work well when there are multiple hands at the same time, which can't be achieved by the existing wireless-based hand pose reconstruction approaches.

4.4.5 Robustness Analysis. In this section, we prove the robustness of our model when applied to new scenarios not seen in the training dataset. The impact of different distances, different angles, different environments, different subjects and multiple simultaneous existing hands are further evaluated in the following experiments. In all these experiments, the training data used for our model is collected in the basic scenario with 50 cm distance, 90° azimuth, and 0° elevation, then augmented by the methods described in Section 3.6.

Robustness to Different Distances To evaluate if our model can keep good performance when the hands are in distances not seen in the training dataset, we conduct experiments that apply the model trained with 50 cm data to the acoustic data collected in the distances 25 cm, 100 cm, and 150 cm, respectively. In those experiments, the azimuth and elevation angle are fixed to 0 degree to only analyze the impact of different distances. The results are shown in Figure 9a. We can see that when the distance changes to 25 cm and 100 cm, the accuracy is still lower than 20 mm, which is a satisfying result considering that we never use any training data collected from those distances. When the distance increases to 150 cm, the performance degrades to 30.08 mm, due to the fast attenuation of the acoustic signal. However, the current results already support the users to interact with their smart devices in a large area, and our method is better than the baseline, BeyondVoice [36], in all the distances. The first column of figures in Figure 6 shows the reconstructed hand pose when the inference is on data collected at 100 cm. We can see that the pose reconstructed by our system looks very similar to the ground truth while the baseline method fails in this case. To evaluate how effective the data augmentation methods are, we also report the results when our model is trained without any augmented data in the same figure. The average errors increase obviously in all the distances that are not in the training set.

Robustness to Different Azimuth/Elevation We also assess the impact of angles not seen during training to the performance of our model. We collected additional data when the azimuth/elevation angle between the hands and the microphone array are 105°/15°, 120°/30°, 135°/45° while keeping the distance as 50 cm, which provides data at 6 new locations. The results are shown in Figure 9b and 9c. As the azimuth/elevation between the hand and the microphone array increases, the error increases slightly due to the imperfection of the augmentation. However, the results are still



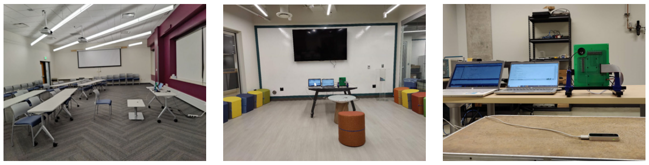
(a) Distance (cm)

(b) Azimuth (°)

(c) Elevation (°)

(d) Cross-Subject User Index

Figure 9: Impacts of different factors on average joint localization errors (mm).



(a) Room 1

(b) Room 2

(c) Room 3

Figure 10: Different Environments.

acceptable and using augmented data will significantly improve the performance when the hands are in new angles. Considering that the users usually interact with their smart devices within restricted angles, our system can still work well when the users are changing their positions. The second column of Figure 6 shows the results on data collected at 120° azimuth and the third column shows the results on 30° elevation’s data.

Robustness to Different Environments In order to prove that the performance of our RAM-Hand system is not affected when applied to environments that are not in the training set, we conduct experiments in other two rooms shown in Figure 10b and 10c using the model trained in room 1 (Figure10a). We also keep the distance as 50 cm and the angles as 0 degree. The average joint localization errors in room 2 and room 3 are 17.42 mm and 19.65 mm, respectively. Therefore, we can conclude that the proposed system is robust to the environment change because its accuracy doesn’t decrease much when applied to new environments.

Robustness to Different Subjects We conduct leave-one-user-out experiments to evaluate our system’s generalization ability to new users. Specifically, our model is trained with the data collected from seven subjects then tested on the other subject. We test each of the subjects in turn with and without using contrastive learning in our model to test if this design helps cross-subject inference. From the results shown in Figure 9d, we can see that there is a significant improvement on the performance when contrastive learning is applied, which reduces the average leave-one-user-out result from 30.23 mm to 26.19 mm, and it also has a better performance than the baseline method. Therefore, our RAM-Hand system also behaves well when applied to new users’ data, which greatly simplifies the training procedure as it won’t need to fine-tune for every new user.

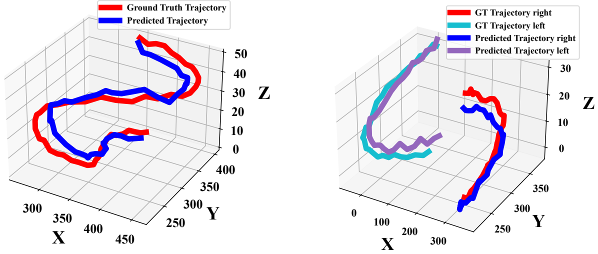
Robustness in Real-world Scenario Finally, we consider a scenario where the subjects can move their hands freely in 3D space while performing hand gestures. That means the distance, azimuth and elevation of the hand can change randomly at the same time, which is very similar to the real-world scenario. There is no occlusion between the subjects’ hands and the microphone array. We also collect 60 seconds data for each of the 12 gestures and evaluate the performance of the model trained only using data collected in the basic scenario and tested on data collected in this scenario. During the data collection, we also played randomly selected YouTube videos as background music to simulate the noise in daily environments. In addition, there is another person in the same room acting as an interferer, who randomly walks around at least one meter away from the subject in the same room. In the meanwhile, the interferer can also talk to make some noise. Our system achieves an accuracy of 24.91 mm in the real-world scenario, while the accuracy of the baseline method, Beyond-Voice, is 30.39 mm. Thus, we can conclude that our proposed method can still behave well in the real-world scenario where the distance, azimuth and elevation of the hand vary simultaneously.

4.4.6 *Root Joint Localization.*

The previous experiments focus on measuring the hand pose reconstruction accuracy, but the location of the whole hand, or the root joint, is also important in many applications. Therefore, we ask the subjects to move their right hands freely in the 3D space for 120 seconds, while performing random hand gestures. We also conduct experiments where the subjects use both hands to draw random shapes to show that our method can localize multiple hands simultaneously. Because the predicted hand location is in the microphone array’s coordinate system while the leap motion’s coordinate system, we calibrate the two coordinate systems before the experiments by first let the subject draw a circle then aligning the center of the of the predicted circle and the ground truth circle. The accuracy of single-hand localization achieves 29.3 mm in the 3D space and the accuracy of two-hands localization is 29.58 mm for the left hand and 28.42 mm for the right hand. The qualitative results for both single-hand and two-hand trajectories are visualized in Figure 11. From the above results, we can conclude that our system is able to precisely localize the root joint location of the hands. Without using the extended Kalman filter, the single-hand localization accuracy is 36.45 mm, and the two-hands localization accuracies are 37.41 mm and 36.82 mm for left hand and right hand, respectively. Those results demonstrate that using the extended Kalman filter is helpful to improve the root joint localization accuracy.

4.4.7 *Running Time Analysis.*

We evaluate the running time of our proposed system to show its practicality in real-world applications. We implement both the signal processing and the deep learning using Pytorch on a server with NVIDIA A6000 GPU. The results show that the average time required to extract features from one frame of data (0.1 seconds) is 51.5 ms and the inference time of the deep learning model for one frame feature is 1.96 ms. Thus, the proposed system could achieve a frame rate around 18.7 fps, which can support real-time interaction between the user and smart devices.



(a) Single Hand **(b) Two Hands**

Figure 11: Root Joint Localization Results.

5 Related Work

Acoustic Sensing Nowadays, leveraging acoustic signals to sense the surrounding objects, especially human activities has succeeded in many applications, such as gesture recognition [18, 25, 35, 49, 55, 60, 66, 70, 77], device tracking [6, 17, 39, 65, 82], hand tracking [34, 37, 41, 45, 69, 83], facial expression recognition [13, 75] and imaging [40, 67]. However, the above systems can only predict a label of the gesture, track the hand as a single point, or image a very coarse 2D shape, while our system can precisely estimate the 3D locations of each hand joint.

Acoustic signal has also been used in pose estimation. EchoWrist [31] can estimate hand poses using acoustic signals transmitted by a wristband. However, it only works for the hand wearing the wristband, whereas our system can support device-free pose reconstruction for multiple hands. SonicHand [68] builds a hand pose reconstruction system on a smartphone. But it can't handle the hand in unseen locations or multiple hands, due to the poor sensing ability of the smartphone. The existing work that is most similar to our system is Beyond-Voice [36], which also reconstructs hand pose using a microphone array. However, they can only estimate the pose of one hand while our system has the ability to handle multiple simultaneous existing hands when the model is only trained with one hand's data, due to the design of segmenting each hand's reflection in the extracted features. And Beyond-Voice only takes the range information as features, so the range-velocity-angle features used in our system can provide more comprehensive information and enhance the robustness. To the best of our knowledge, our RAM-Hand system is the first system that can reconstruct poses of multiple hands using wireless signals only.

RF Signal-based Pose Reconstruction Recently, many researchers have spent their efforts on using RF signals for human pose reconstruction, instead of focusing solely on traditional gesture/activity recognition tasks [7, 10, 23, 74]. Those systems are built on specialized hardware [84, 85], Wi-Fi [24, 52, 71, 81] and mmWave Radar [30, 78–80]. However, all the above systems can only reconstruct the coarse-grained human pose. With the help of our customized signal processing pipeline and transformer model, our RAM-Hand system is able to precisely reconstruct the fine-grained hand poses and won't suffer from obvious performance degradation when applied to new scenarios. [22] utilizes Wi-Fi for hand pose reconstruction, but it has a limited accuracy and can't handle multiple hands due to the low resolution of Wi-Fi. The authors of mm4Arm [38] propose to sense the vibration of the user's forearm using a mmWave Radar and then reconstruct the hand pose indirectly. However, it requires the radar to face the user's forearm, which is unrealistic in practical use. MmHand [29] can directly sense the hand and reconstruct the pose with a mmWave Radar. Nonetheless, it can't handle multiple hands and has poor resolution in the elevation dimension due to the hardware design, which limits its performance. Our system enables the users to reconstruct poses of multiple hands in a large area without obvious performance degradation, which can be used in a broader range of applications.

Vision-based Hand Pose Reconstruction Reconstructing hand poses is a well-studied problem in the computer vision area. Either depth camera [14, 15, 27, 47, 50, 64, 76] and RGB camera [5, 16, 44, 56–58, 88] can be used to reconstruct accurate hand poses. Nonetheless, the vision-based methods require good lightning conditions and are vulnerable to occlusion. Recording videos may also raise privacy concerns, which narrows the usage scenarios of vision-based methods. Because our RAM-Hand system only uses acoustic signals, it can address all the above issues while maintaining good performance. Therefore, our RAM-Hand system will be a good complement to the vision-based approaches.

Wearable Device-based Hand Pose Reconstruction To precisely reconstruct the hand poses, wearing some devices equipped with sensors on the users' hands is another popular research area. Existing approaches can be classified into glove-based methods [9, 19, 32, 42, 46], ring-based methods [87] and wrist-mounted sensor-based methods [20, 28, 31, 73]. Nevertheless, all the above methods may make the users feel troublesome, as they all require the users to wear additional devices on their hands. They also have the limitation that only the pose of the hand wearing the devices can be reconstructed. Therefore, as a device-free solution that supports multiple hands, our RAM-Hand system can be used in a wider range of applications and offer improved user-friendliness.

6 Conclusion

In this paper, we propose an acoustic multi-hand pose reconstruction system, RAM-Hand, which is able to precisely predict the 3D location of every joint of multiple hands and stay robust in new scenarios where the hand positions, environments, and subjects are not seen in the training dataset. To this end, we design a customized signal processing pipeline to extract the range, angle, velocity features and segment the reflections of each hand, then we use a transformer-based neural network to reconstruct the hand poses. To address the challenge that it's hard to train a model with good generalizability using limited data, we propose a series of data augmentation methods to generate virtual training samples from the real collected data. To further enhance the robustness for cross-subject inference, we utilize contrastive learning to guide our model to only focus on the features independent of different subjects. We implement our system on a microphone array and conduct extensive experiments to demonstrate the effectiveness and robustness of our model.

Acknowledgments

This work was supported by the U.S. National Science Foundation under Grants CNS-2154059 and IIS-2348427.

References

[1] AVNET. [n. d.]. Ultra96-V2. https://www.avnet.com/wps/portal/us/products/avnet-boards/avnet-board-families/ultra96-v2/

[2] BELA.io. [n. d.]. BELA/BELA MINI. https://bela.io/products/bela-and-bela-mini/

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. 2021. Is space-time attention all you need for video understanding?. In *ICML*, Vol. 2. 4.

[4] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems* 6 (1993).

[5] 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)*. 666–682.

[6] 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.

[7] Qiming Cao, Hongfei Xue, Tianci Liu, Xingchen Wang, Haoyu Wang, Xincheng Zhang, and Lu Su. 2024. mmCLIP: Boosting mmWave-based Zero-shot HAR via Signal-Text Alignment. In *Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems*. 184–197.

[8] Jack Capon. 1969. High-resolution frequency-wavenumber spectrum analysis. *Proc. IEEE* 57, 8 (1969), 1408–1418.

[9] 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 *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2568–2573.

[10] Shuya Ding, Zhe Chen, Tianyue Zheng, and Jun Luo. 2020. RF-net: A unified meta-learning framework for RF-enabled one-shot human activity recognition. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 517–530.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[12] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise *(KDD'96)*. AAAI Press.

[13] Yang Gao, Yincheng Jin, Seokmin Choi, Jiyang Li, Junjie Pan, Lin Shu, Chi Zhou, and Zhanpeng Jin. 2021. SonicFace: Tracking Facial Expressions Using a Commodity Microphone Array. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 4 (2021), 1–33.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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 *2015 37th annual international conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 4631–4634.

[20] 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.

[21] Texas Instruments. 2020. The fundamentals of millimeter wave radar sensors.

[22] Sijie Ji, Xuanye Zhang, Yuanqing Zheng, and Mo Li. 2023. Construct 3D Hand Skeleton with Commercial WiFi. In *Proceedings of the 21st ACM Conference on Embedded Networked Sensor Systems*. 322–334.

[23] Wenjun Jiang, Chenglin Miao, Fenglong Ma, Shuochao Yao, Yaqing Wang, Ye Yuan, Hongfei Xue, Chen Song, Xin Ma, Dimitrios Koutsonikolas, et al. 2018. Towards environment independent device free human activity recognition. In *Proceedings of the 24th annual international conference on mobile computing and networking*. 289–304.

[24] 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.

[25] 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.

[26] Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, et al. 2019. A comparative study on transformer vs rnn in speech applications. In *2019 IEEE automatic speech recognition and understanding workshop (ASRU)*. IEEE, 449–456.

[27] 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.

[28] 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.

[29] Hao Kong, Haoxin Lyu, Jiadi Yu, Linghe Kong, Junlin Yang, Yanzhi Ren, Hongbo Liu, and Yi-Chao Chen. 2024. mmHand: 3D Hand Pose Estimation Leveraging mmWave Signals. In *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 1062–1073.

[30] Hao Kong, Xiangyu Xu, Jiadi Yu, Qilin Chen, Chenguang Ma, Yingying Chen, Yi-Chao Chen, and Linghe Kong. 2022. m3track: mmwave-based multi-user 3d posture tracking. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*. 491–503.

[31] Chi-Jung Lee, Ruidong Zhang, Devansh Agarwal, Tianhong Catherine Yu, Vipin Gunda, Oliver Lopez, James Kim, Sicheng Yin, Boao Dong, Ke Li, et al. 2024. Echowrist: Continuous hand pose tracking and hand-object interaction recognition using low-power active acoustic sensing on a wristband. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–21.

[32] 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.

[33] 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.

[34] 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.

[35] Dong Li, Jialin Liu, Sunghoon Ivan Lee, and Jie Xiong. 2022. Room-scale hand gesture recognition using smart speakers. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*. 462–475.

[36] Yin Li, Rohan Reddy, Cheng Zhang, and Rajalakshmi Nandakumar. 2023. Beyond-Voice: Towards Continuous 3D Hand Pose Tracking on Commercial Home Assistant Devices. *arXiv preprint arXiv:2306.17477* (2023).

[37] Chao Liu, Penghao Wang, Ruobing Jiang, and Yanmin Zhu. 2021. AMT: Acoustic multi-target tracking with smartphone mimo system. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 1–10.

[38] 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.

[39] 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.

[40] 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.

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

[42] 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 *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 3186–3191.

[43] miniDSP. [n. d.]. UMA-16 v2 USB mic array. https://www.minidsp.com/products/usb-audio-interface/uma-16-microphone-array

[44] 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.

[45] 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.

[46] 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 *The Sixth International Conference on Sensor Device Technologies and Applications*. International Academy, Research, and Industry Association.

[47] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. 2011. Efficient model-based 3D tracking of hand articulations using Kinect.. In *BmVC*, Vol. 1. 3.

[48] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] Maria Isabel Ribeiro. 2004. Kalman and extended kalman filters: Concept, derivation and properties. *Institute for Systems and Robotics* 43, 46 (2004), 3736–3741.

[54] Hermann Rohling. 1983. Radar CFAR Thresholding in Clutter and Multiple Target Situations. *IEEE Trans. Aerospace Electron. Systems* AES-19, 4 (1983), 608–621.

[55] 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.

[56] 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.

[57] 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 (TOG)* 39, 6 (2020), 1–14.

[58] Adrian Spurr, Umar Iqbal, Pavlo Molchanov, Otmar Hilliges, and Jan Kautz. 2020. Weakly supervised 3d hand pose estimation via biomechanical constraints. In *European conference on computer vision*. Springer, 211–228.

[59] Seeed Studio. [n. d.]. ReSpeaker Mic Array v2.0. https://www.seeedstudio.com/ReSpeaker-Mic-Array-v2-0.html

[60] 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.

[61] tindle. [n. d.]. ICS4135X Digital PDM Microphone Breakout. https://www.tindie.com/products/onehorse/ics4135x-digital-pdm-microphone-breakout/

[62] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[63] 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.

[64] 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.

[65] 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.

[66] 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.

[67] Penghao Wang, Ruobing Jiang, and Chao Liu. 2022. Amaging: Acoustic hand imaging for self-adaptive gesture recognition. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 80–89.

[68] 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 Transactions on Sensor Networks* 20, 5 (2024), 1–32.

[69] 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.

[70] 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.

[71] Yichao Wang and Jie Yang. 2022. 3D Human Mesh Construction Leveraging Wi-Fi. *arXiv preprint arXiv:2210.10957* (2022).

[72] 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.

[73] 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.

[74] Rui Xiao, Jianwei Liu, Jinsong Han, and Kui Ren. 2021. Onefi: One-shot recognition for unseen gesture via cots wifi. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*. 206–219.

[75] Wentao Xie, Qian Zhang, and Jin Zhang. 2021. Acoustic-based Upper Facial Action Recognition for Smart Eyewear. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 2 (2021), 1–28.

[76] 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.

[77] 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.

[78] Hongfei Xue, Qiming Cao, Yan Ju, Haochen Hu, Haoyu Wang, Aidong Zhang, and Lu Su. 2022. M4esh: mmwave-based 3d human mesh construction for multiple subjects. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*. 391–406.

[79] Hongfei Xue, Qiming Cao, Chenglin Miao, Yan Ju, Haochen Hu, Aidong Zhang, and Lu Su. 2023. Towards generalized mmwave-based human pose estimation through signal augmentation. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. 1–15.

[80] 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.

[81] Kangwei Yan, Fei Wang, Bo Qian, Han Ding, Jinsong Han, and Xing Wei. 2024. Person-in-WiFi 3D: End-to-End Multi-Person 3D Pose Estimation with Wi-Fi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 969–978.

[82] 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.

[83] 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.

[84] 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)*.

[85] 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.

[86] 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.

[87] 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.

[88] 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.