LeakyFeeder: In-Air Gesture Control Through Leaky Acoustic Waves

Yongjie Yang[Pitt], Tao Chen[Pitt], Zhenlin An[Pitt], Shirui Cao[UMass], Xiaoran Fan[G], Longfei Shangguan[Pitt]
[Pitt]University of Pittsburgh, [UMass]University of Massachusetts Amherst, [G]Google

ABSTRACT

We present LeakyFeeder, a mobile application that explores the acoustic signals leaked from headphones to reconstruct gesture motions around the ear for fine-grained gesture control. To achieve this goal, LeakyFeeder repurposes the speaker and a single feed-forward microphone on active noise cancellation (ANC) headphones as a SONAR system, using inaudible frequency-modulated continuous-wave (FMCW) signals to track gesture reflections for accurate sensing. Since this single-receiver SONAR system is unable to differentiate reflection angles and further disentangle signal reflections from different gesture parts, we draw on principles of multi-modal learning to frame gesture motion reconstruction as a multi-modal translation task and propose a deep learning-based approach to fill the information gap between low-dimensional FMCW ranging readings and high-dimensional 3D hand movements. We implement LeakyFeeder on a pair of Google Pixel Buds and conduct experiments to examine the efficacy and robustness of LeakyFeeder in various conditions. Experiments based on six gesture types inspired by Apple Vision Pro demonstrate that LeakyFeeder achieves a PCK performance of 89% at 3cm across ten users, with an average MPJPE and MPJRPE error of 2.71cm and 1.88cm, respectively.

CCS CONCEPTS

• **Human-centered computing → Ubiquitous and mobile computing systems and tools**.

KEYWORDS

Wearable Computing, Gesture Recognition, Acoustic Sensing

ACM Reference Format:
Yongjie Yang[Pitt], Tao Chen[Pitt], Zhenlin An[Pitt], Shirui Cao[UMass], Xiaoran Fan[G], Longfei Shangguan[Pitt]. 2025. LeakyFeeder: In-Air Gesture Control Through Leaky Acoustic Waves. 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/10.1145/3715014.3722054

1 INTRODUCTION

In-air gesture plays a crucial role in many emerging applications - from smart home where users can adjust lighting or temperature settings with simple hand gestures or finger motions [36, 68], to virtual reality (VR) scenarios [18], allowing users to interact with virtual objects without physical touch. In-air gestures also enable seamless interaction in healthcare, maintain reliability when wearing gloves, and enhance comfort by eliminating physical contact, making them a versatile complement to traditional touch-based controls.

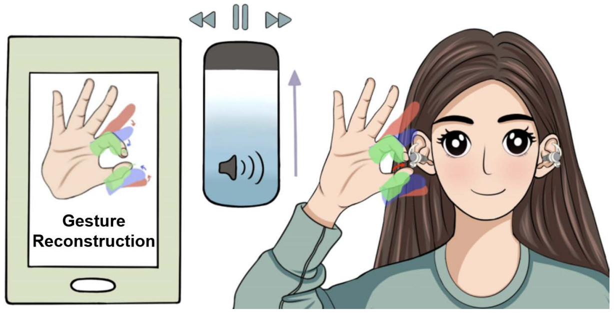

Figure 1: A use case of LeakyFeeder: Alice tunes up her music volume by performing a friction gesture slowly as if she is rotating a physical volume knob.

Most wireless sensing systems designed for in-air gesture control treat gestures as single actions, such as recognizing whether it is a pinch, a sweep, or a friction gesture [68]. They however overlook the critical details of gesture motion such as the speed or intensity variation (*i.e.*, how the user performs gestures), which are crucial for more realistic gesture interactions. For instance, if the music volume is too low, the mobile user often turns the knob rapidly to raise the volume to a reasonable level, then slows down to fine-tune it to the ideal setting. Therefore, obtaining the speed variations of a friction gesture (emulating the way of turning the volume knob) could allow users to control volume dynamics more precisely and realistically. Likewise, in VR environments, tracking the *gesture motion process* (as opposed to recognizing the gesture alone) can enhance virtual object manipulation – if a user is "throwing" an object, the system could use the speed of their gesture to determine the force or distance, making interactions more realistic.

In this paper, we introduce LeakyFeeder, a mobile system that can accurately track hand motions near the user's head in 3D space over time, enabling a full gesture motion reconstruction. We envision that with LeakyFeeder, users can enjoy gesture controls that feel as natural as turning a physical knob to adjust the volume. Additionally, LeakyFeeder could enrich the VR experience by allowing users to see and refine their gestures within the virtual environment, creating a seamless bridge between physical actions and digital responses.

Different from existing gesture tracking systems that rely on either dedicated, bulky, or costly hardware devices such as mmWave RADAR [38], LiDAR [20], or microphone/RF antenna arrays [45], LeakyFeeder achieves fine-grained 3D gesture reconstruction by using barely a pair of speaker transducer and out-ear microphone on headphones, as shown in Figure 1. As speaker transducers are

This work is licensed under a Creative Commons Attribution 4.0 International License.
SenSys '25, May 6–9, 2025, Irvine, CA, USA
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1479-5/2025/05.
https://doi.org/10.1145/3715014.3722054

the basic component of every pair of headphones while out-ear microphones such as the mic for talking or the feed-forward mic for noise cancellation are gradually integrated into headphones, we envision the mobile users can easily access in-air gesture interactions without extra hardware support.

LeakyFeeder is based on a basic observation – if someone nearby is listening to music on her headphone, unintentionally, the sound would escape, allowing others to hear it. When a mobile user performs a gesture near her headphone, leaky acoustic waves reflect off her hand, and the echoes are captured by the out-ear microphones. This essentially creates a SONAR system. Our basic idea is to leverage SONAR principle to reconstruct 3D hand skeleton positions over time. While the high-level idea of using headphones as a SONAR for gesture reconstruction is intuitive, there are still many open research questions. For example, do the minute finger motions such as squeeze offer sufficient SNR for SONAR detection? Are techniques like FMCW adequate with minor modifications? Given the fact that the single-receiver headphone SONAR lacks the capability to differentiate signal reflections from different gesture part, would existing well-studied data-driven approaches be adequate with minor modification? Or are clean-slate models necessary?

With these questions in mind, we first conduct extensive microbenchmarks to de-risk the capability – assessing the feasibility of leveraging leaky acoustic waves for in-air gesture control. Specifically, we start with experiment to validate the ubiquity of leaky acoustic waves with 26 pairs of headphones. We then apply FMCW Radar principle to the ultrasound band to examine headphone SONAR's field of view (FoV), the sensing resolution, and the sensing granularity – three key parameters to the success of 3D gesture tracking over time, in an acoustic chamber. The preliminary results are promising: we find that leaky signals are ubiquitous on all experimental headphones due to both unintentional loss coupling effect and intentional design considerations and this headphone SONAR is capable of detecting the entire hand with sufficient resolution (≥1.5cm) and granularity (≥2.2mm).

However, since most headphones are equipped with only one microphone, the resulting single-receiver SONAR system lacks the ability to map distance variations to different parts of a human hand, let alone tracking the fine-grained gesture motions over time. Inspired by multi-modal learning approaches [37, 67], we transform the problem of reconstructing gestures from superimposed distance measurements into a multi-modal translation task – translating FMCW ranging data from the SONAR domain into 3D hand skeleton movements in the vision domain. We then customize transformer [29] model to capture the spatial and temporal variations in reflection paths from FMCW input, extract fine-grained motion features, and further map these feature representations to the visual domain that accurately aligns with 3D hand skeleton movements.

We implement LeakyFeeder on a pair of Google Pixel Buds and conduct extensive evaluations based on six gestures defined by Apple Vision Pro [42]. Experimental results with ten volunteers performing six gestures demonstrate an overall mean per joint position error of 2.71 cm ± 0.71 cm and mean per joint relative position error of 1.88 cm ± 0.31 cm. For keypoint tracking accuracy, LeakyFeeder achieves a PCK performance of 89% at 3cm error.

In summary, this paper makes the following contributions:

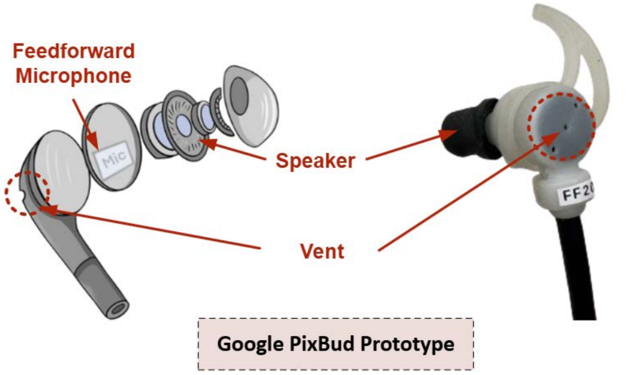

Figure 2: Breakdown of Google ANC Pixel Bud prototype.

- We systematically study the signal leakage phenomenon on different types of headphones and validate the feasibility, derive technical challenges, and observe key opportunities of using such leakage signal for reconstructing gesture motions.
- We formulate the gesture motion tracking in 3D space as a multi-modal translation task and customize a transformer-based neural network that can effectively translate coarse-grained FMCW readings into fine-grained gesture motions.
- We build a prototpye of LeakyFeeder on Google Pixel Buds and conduct extensive experiments involving ten subjects and six gesture types inspired by Apple Vision Pro. The result demonstrate promising performance.

2 LEAKY ACOUSTIC WAVES: PRINCIPLE, OPPORTUNITIES, AND CHALLENGES

In this section, we first introduce the concept of leaky acoustic waves and explain its ubiquity in our daily lives (§2.1). We then experimentally validate the feasibility of leveraging leaky acoustic waves for in-air gesture control (§2.2). Finally, we summarize the challenges on leveraging these leaky acoustic waves for gesture motion reconstruction in 3D space (§2.3).

2.1 The Ubiquity of Leaky Acoustic Waves

Most people have likely experienced the phenomenon where someone nearby is listening to music on her headphone, and unintentionally, the sound would escape, allowing others to hear it. These leaky music, or generally speaking, leaky acoustic waves, are common for the following reasons [11, 19, 23].

Firstly, a large portion of the acoustic signal emitted by earphone speakers may reflect off the ear canal and escape from the earphone. Over-ear and on-ear earphones, which do not completely seal the ear canal, are especially susceptible to signal leakage. Secondly, as shown in Figure 2, although in-ear headphones typically provide a better seal when inserted into the ear, they often feature a small vent on the back of the housing. This vent is designed to allow air to flow in and out, which equalizes pressure, enabling the speaker transducer to move freely for improved sound quality and a more accurate bass response [57]. However, this vent can also allow internal audio

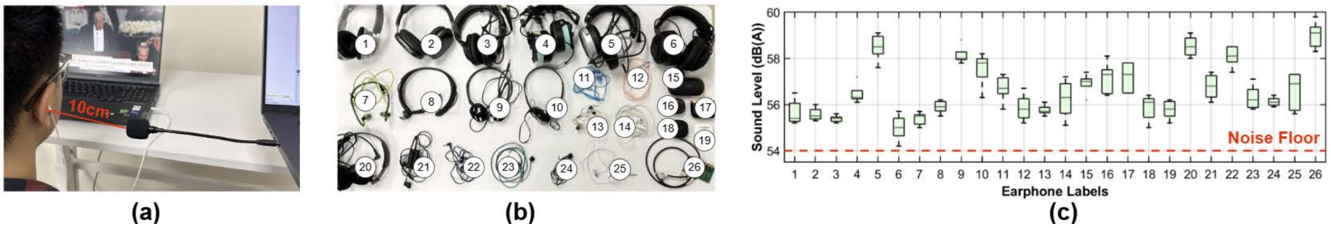

Figure 3: (a) Experimental setup: we play a five-second speech to a mobile user though his headphones. A microphone is placed 10cm away from the headphone to measure the amplitude of the acoustic signal leaked from this headphone.(b) We repeat this experiment on 26 different pairs of headphones. (c) The amplitude (sound level) of acoustic signals leaked from these 26 pairs of headphones.

to escape, resulting in sound leakage. In addition, certain supra-aural earphones, such as the Bang & Olufsen A8 [51] and ATH-EW9 [48], are intentionally designed with sound-leakage features to appeal to users who enjoy a more open and ambient listening experience. As a consequence, the audio played by these earphones also leaks out. Moreover, lower-cost headphones are made with simple plastic casings and lack soundproofing further contributing to acoustic signal leakage in an effort to reduce cost.

To verify whether leaky acoustic waves exist in most headphones, we invited a user to test 26 different kinds of headphones, with prices ranging from $2 to $550, as shown in Figure 3(b), including in-ear, on-ear, and over-ear models. We set the maximum audio volume to 82dB(A) [1] for most headphones to meet the hearing safety regulation [13]. In each test, we play a five-second human speech through the headphones and put a microphone 10cm away from the user to detect the acoustic signals leaked from the headphones, as depicted in Figure 3(a). We repeat this experiment five times for each type of headphone.

Figure 3(c) shows the result. All headphones are found to exhibit noticeable sound leakage, with an average volume level ranging from 54.5 dB(A) to 59.5 dB(A). Another participant stays close to the headphone wearer also confirms that he can hear the leaky signals across all headphone models. Among all 26 pairs of headphones, we find that headphone Model #5, the Hifiman headphones [24], shows the highest sound leakage due to its open-back design. In contrast, the headphone Model #6, the Razer headphones [52], experience less significant signal leakage than all the other types of headphones due to their better sealing condition and thick insulating materials. Nevertheless, the acoustic signal leaked from this headphone is still audible (≥55.4 dB(A)) to the participant nearby.

2.2 Assessing the Feasibility of Using Leaky Acoustic Waves for Gesture Control

Our goal is to harness these leaky signals for in-air gesture control. When a mobile user moves her hand near her headphones, the acoustic waves leaked from the headphones will reflect off her hand, and these echoes will then be captured by the feed-forward microphone positioned on the rear of the headphone, as illustrated in Figure 2. This essentially creates a typical SONAR system. Our basic idea is to leverage SONAR principle to *reconstruct the hand skeletons' 3D position changes over time*.

We validate this idea by analyzing the *field of view* and *sensing granularity* of this headphone SONAR system. Drawing inspiration from RADAR [54] and automotive sensing technologies [50], we employ Frequency-Modulated Continuous-Wave (FMCW) ranging algorithms to measure the change of the distance between each reflection point on the user's hand and the feed-forward microphone on headphones. FMCW signals are chosen because they provide high-precision distance measurements by encoding depth information into frequency shifts. Unlike ambient leaky audio signals, FMCW ensures stable, controlled, and repeatable waveforms that are independent of environmental noise and variations in music or speech content. Also, the use of ultrasonic FMCW signals minimizes interference with human hearing.

Experiment Setups. We plug one Google Pixel Bud into a dummy head and play FMCW signals at an 18-78kHz frequency band for the experiment. The volume of this FMCW signal is set to 82dB(A). The bandwidth and chirp symbol time are set to 20kHz and 42ms, respectively.[2] We wire out audio signals from the Pixel Bud's feed-forward microphone and sample them using a Babyface AD/DA converter [1]. Next, we put a 15cm × 15cm sound-absorbing foam (approximately the size of an adult hand while eliminating unnecessary reflections) in front of the dummy head, as shown in Figure 4(a). The form is further divided into 100 grids (1.5cm × 1.5cm each). We then poke through each grid and move a 1-cm diameter straw with perforated sealed ends back and forth through each grid, emulating the presence of a small reflector at each grid to test whether this reflector can be detected by our headphone SONAR. All experiments are conducted in an acoustic chamber, as shown in Figure 4(a).

2.2.1 *Understanding the Headphone SONAR's Field of View.*

With this experiment setup, we first examines the *field of view (FoV)* of this headphone SONAR system across various hand-to-earphone distance configurations. This helps understand whether this system can detect the entire hand's movement at different distances. The result will also inform us about the *sensing resolution* – the minimum distance between two reflection points (*i.e.*, fingers) where they can be separated by this SONAR system.

We define a successful detection of the straw's back-and-forth movement as the appearance of a signal peak in the FMCW spectrum (Figure 4(c)), demonstrating the existence of a reflector. For reference, Figure 4(b) shows the FMCW spectrum in the absence

[1] Hearing protection is recommended at 85 dB(A) for over 8 hours according to CDC guidelines. No conclusive studies link prolonged ultrasound to hearing loss [58], though it may cause subjective discomfort.

[2] See details about FMCW signal processing in §4.3.

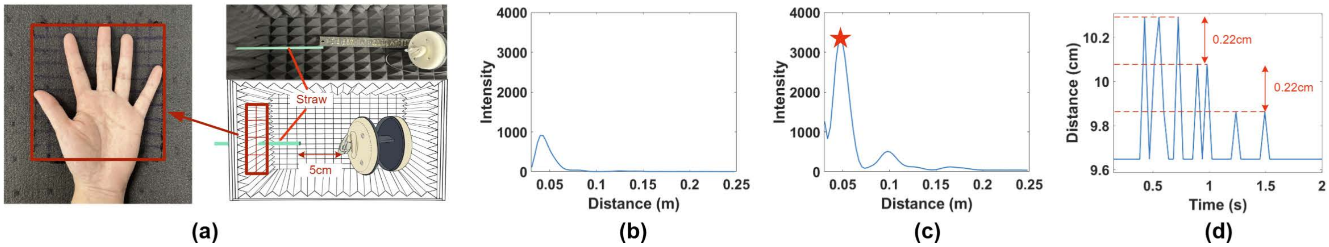

Figure 4: (a) Experimental setup: we test the field of view, sensing range, resolution, and granularity using a hand-crafted testbed. (b) The FMCW spectrum in the absence of any reflectors. (c) The FMCW spectrum in the presence of a reflector 5cm away from the headphone. (d) Examine the sensing granularity of this headphone SONAR system.

of a reflector. We repeat this experiment in each grid 100 times. Figure 5 shows the heatmap of success rate across different grids and in different form-to-earphone distance settings. As shown in Figure 5(a)-(b), the headphone SONAR can reliably detect straw's movement at all 100 grids when the sound-absorbing foam is placed up to 10cm away from the headphone. When the spacing is increased to 15cm, as shown in Figure 5(c), the reflector (*i.e.*, the straw) becomes less detectable at grid points located along the boundary of the foam. When we expand the spacing further to 20cm, as shown in Figure 5(d), the leaky signals experience severe attenuation, which makes their echoes less detectable at the microphone. These experiments demonstrate that this headphone SONAR system is capable of detecting full hand gestures with a resolution of at least 1.5cm (a distance between two fingers) at a distance up to 15cm.

2.2.2 Assessing the Headphone SONAR's Sensing Granularity. Next, we assess this headphone SONAR's sensing granularity, *i.e.*, to understand to what extent the SONAR system can detect the hand's movements. Theoretically, the sensing granularity of a SONAR system is inversely proportional to the central frequency f_c of the probing signal [7, 27]:

$$\Delta d_{\min} \propto \frac{c\Delta\phi}{4\pi f_c},$$

where c represents the speed of sound, and $\Delta\phi$ is the phase difference of the reflected signal. The phase variation $\Delta\phi$ is inversely proportional to the probing signal's wavelength, which is determined by its central frequency. Therefore, the higher of the central frequency f_c, the larger phase variation it will be. As a result, higher central frequencies improve the system's ability to detect small displacements because the phase variation will be more significant.

In this experiment setup shown in Figure 4(a), we use an ultrasound signal with a central frequency of 48kHz as the probing signal, which improves the sensing granularity of our headphone SONAR system. To validate, we use a ruler as a reference and carefully move a straw (the reflector) at varying distances from 9.6cm to 10.3cm. Figure 4(d) presents the FMCW ranging practical results, showing this headphone SONAR can detect displacements as small as 2.2mm. Such a high granularity enables us to detect even tiny hand frictions.

2.3 Challenges in Gesture Reconstruction

The preliminary results are promising so far – we find that this headphone SONAR is capable of detecting the entire hand with sufficient resolution (≥1.5cm) and granularity (≥2.2mm). However, since most headphones are equipped with only a single feed-forward microphone and one speaker to construct the SONAR system, the resulting SONAR system lacks the ability to obtain spatial information from different angles compared to microphone array SONAR systems. This hardware limitation presents fundamental challenges for reconstructing a 3D hand's motion.

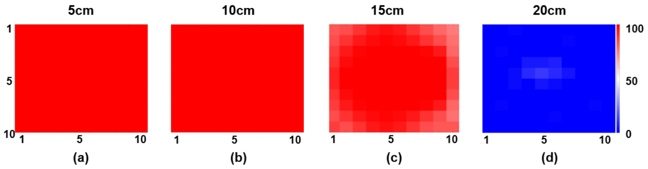

Figure 5: Heatmaps show the results of signal intensity variations across four distances: (a) 5 cm, (b) 10 cm, (c) 15 cm, (d) 20 cm.

- Firstly, the headphone SONAR struggles to identify the exact location of each reflection point at the hand. Although it can detect the distance of a reflection point with respect to the headphone, it cannot pinpoint the exact location of the reflection on the hand. This lack of spatial clarity creates uncertainty in determining the hand's position and understanding its movements.
- Secondly, as the microphone on this SONAR system captures the super-position of multiple reflections, distinguishing individual reflection and further track their variations over time at the 3D plane is challenging to achieve. Consequently, the system struggles to interpret the finer details of joint motion, hindering its ability to reconstruct the hand's motion.

3 DESIGN

In essence, reconstructing gesture motion from headphone FMCW readings is equivalent to generating high-dimensional structured data from low-dimensional data by filling the information gap in-between. Although it is challenging to do so because multiple high-dimensional representations can map to the same low-dimensional data, creating gesture ambiguity, there are still opportunities that can be leveraged to fill this information gap, as outlined below.

Certainty in Uncertainty. While human gesture sets can be highly diverse, gestures performed around the ear by mobile users tend to follow certain fundamental principles. Firstly, as shown in Figure 6(a), upon performing a gesture, the palm typically faces the ear or headphone, with fingers pointing either upward or toward the ear. Secondly, as gestures shown in Figure 6, according to *ergonomic principles* [70], finger joints bend naturally toward the palm when

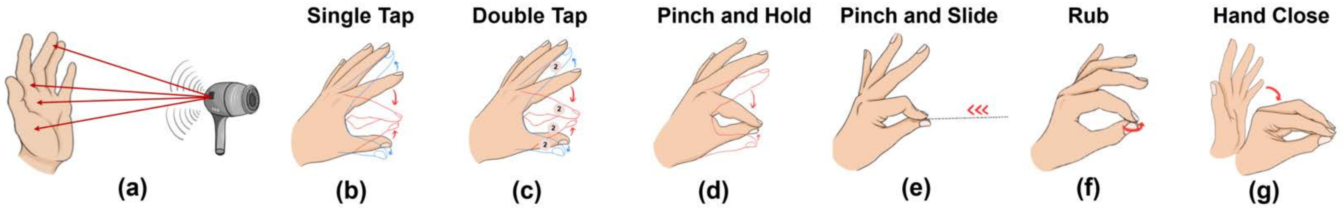

Figure 6: (a) FMCW signal detection of hand positions; (b) single tapping gesture; (c) double tapping gesture; (d) finger pinching and holding gesture; (e) pinching motion with a gentle sliding movement; (f) rubbing motion between thumb and index finger; (g) hand-closing gesture.

performing gestures. Thirdly, *motion continuity* ensures that each finger movement follows previous motions, avoiding abrupt shifts.

Thus, although mapping a single frame of FMCW distance readings to an individual gesture pose can be ambiguous, the continuity of FMCW readings (over a period of time), combined with ergonomic principles and motion continuity, can disambiguate the gesture mapping, providing us a means to bridge the information gap for gesture motion reconstruction.

3.1 Problem Formulation

A Deep Learning-based Approach. These key observations lead us to think of using deep learning techniques for gesture motion reconstruction. We believe that the deep learning approach is promising because it can capture temporal dependencies within the FMCW readings and further extract their spatial features in higher-dimensional latent spaces that are better aligned with the gesture dynamics. This transformation could potentially enable the model to decouple spatial reflectors and access subtle changes in distance and velocity at each reflector that are less evident in raw headphone FMCW readings, simplifying the gesture motion reconstruction. Moreover, deep learning approaches such as generative adversarial network (GAN) [8, 21] and variational autoencoder (VAE) [31] have been shown very effective in producing structured, high-dimensional outputs from low-dimensional data, such as image super-resolution [32, 33, 61], text2image synthesis [44, 53, 66], and 3D image reconstruction [64].

We formulate this learning to reconstruct 3D gesture from FMCW readings as a multi-modal translation task [37] – *translating FMCW ranging data from the SONAR domain into 3D hand skeleton movements in the vision domain*. To this end, we build training set by capturing the human gesturing using both the headphone SONAR and a Leap Motion device that captures the fine-grained finger motion at 3D space (represented as 21 joints' locations). Although these two modalities are heterogeneous, they are connected due to the shared information, *i.e.*, both two modalities can obtain the depth information from each reflection point. Therefore, our goal is to design a better representation of the FMCW SONAR readings and learn a translation function that can map this representation into 3D hand gesture skeleton characterized by the 3D position of 21 joints, as shown in Figure 8(a).

3.2 Neural Network

Instead of designing a new deep learning model from scratch, we tend to adapt well-established model architectures to our task by carefully crafting key components such as loss functions. Many of these architectures have proven highly effective for cross-modal translation tasks in diverse domains, such as vision-to-language [60], IMU-to-vision [25], and so on.

Next, we describe the model exploration and adaptation, starting from model input consideration, model architecture selection, down to the design of the loss functions. In particular, we will focus on articulating the rationale for each design choice and explain how we progressively refine loss function designs to enable highly accurate gesture motion reconstruction based on headphone SONAR.

3.2.1 Model Input. Each frame of our FMCW readings describes the likelihood of a reflector presenting at a certain distance with respect to the microphone, as shown in Figure 4(b)–(d). Furthermore, we exclude distances below 2cm to avoid unwanted reflections from the user's auricle. Like video frames, each our FMCW frame barely captures a "single moment of human gesture motion" in a time, which faces ambiguity issues when feeding it into the deep learning model to learn the translation function. Instead, we batch n continuous FMCW frames and feed them together to the deep learning model. As each batch of frames describes the gesture motion in a certain period of time, during training the model can learn a better feature representation of FMCW frames that characterizes hand movement and address specular reflection issues, where single-frame data may miss joint information due to unequal reflection [15, 47].

3.2.2 Model Architecture. Before delving into model architecture selection, let us revisit the primary objectives for our deep learning model. Specifically, we aim to design a model that can learn to (i): effectively capture the spatial and temporal variations in reflection paths from our FMCW input; (ii): extract fine-grained motion features from the super-positioned distance readings on the high-dimensional latent space that closely match the dynamics of gestures; and (iii): map these feature representations to the visual domain in a way that accurately aligns with 3D hand movements.

With these design objectives in mind, we thoroughly review well-studied model architectures and ultimately focus on *Transformer* due to the following reasons.

- **Spatial-temporal correlation**: Transformer is suited for analyzing time-sequenced, multi-frame FMCW readings due to its proven capabilities in capturing the input data's temporal dynamics. This is facilitated primarily through its *positional encoding module* (the green component shown in Figure 7), which explicitly adds the order of the FMCW frame input into their feature embeddings.
- **Feature extraction**: the *embedding component* in the Transformer (the yellow component in Figure 7) is designed to

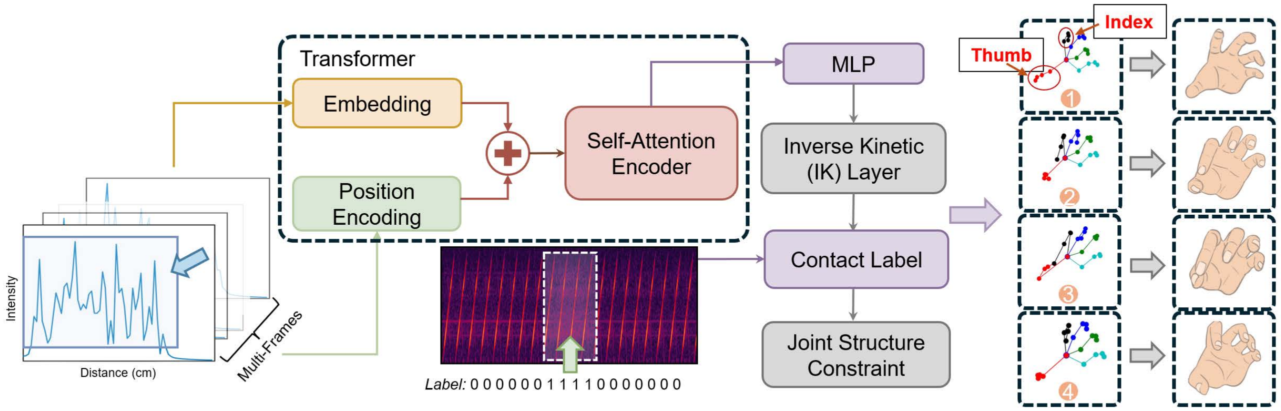

Figure 7: The transformer-based deep learning model architecture effectively translates multi-frame FMCW distance readings into 3D skeletal motions, with the final generated gesture presented in a four-step sequence. Red, black, blue, green, and cyan represent the left-hand joints of the thumb, index finger, middle finger, ring finger, and pinky, respectively.

convert each raw FMCW frame input into vector-based features. This enables the model to process complex patterns more effectively, capturing detailed ranging information and motion features relevant to gesture dynamics.

- **Representation**: the *multi-head attention* mechanism of Transformer (the pink component shown in Figure 7) enables it to understand distance variations across different part of input frames' and at different scales, allowing for effectively FMCW feature presentation.

These three core model components enable Transformer to not only capture the spatial-temporal correlations within each set of FMCW reading input but also to learn motion continuity by processing multiple groups of FMCW samples during training. The understanding of motion continuity is then applied during inference for accurate gesture reconstruction. We use a Multilayer Perceptron (MLP) as the decoder part of this transformer model for the final 3D hand skeleton reconstruction.

3.3 Model Adaptation

Next, we describe how we customize this Transformer model to achieve an accurate gesture motion reconstruction. We start with the loss function design (§3.3.1), and then the integration of inverse kinematic constraint to prevent the model from generating unnatural gestures (§3.3.2). Finally, we explain how the acoustic signals produced by gesture movements are used as cues to further refine the model's gesture motion reconstruction (§3.3.3).

3.3.1 Customize the loss function. In our model shown in Figure 7, the Self-Attention module is connected to a Multi-Layer Perceptron (MLP), which learns a non-linear transformation function to map the feature map representation of FMCW readings to the 4D gesture motion. Like conventional deep learning-based gesture reconstruction methods [6, 40, 47], to train this neural network, the most straight-forward loss function is the mean squared error (MSE) between the ground-truth coordinates and the generated coordinates for each 21 joints of human gesture, expressed as:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{J}\|p_{j,t} - g_{j,t}\|_F^2 \tag{1}$$

where $p_{j,t}$ and $g_{j,t}$ are the predicted and ground truth positions of joint j at time frame t, calculated using the Frobenius norm[3]. T and J are the total number of frames and joints.

However, the testing set shows that this MSE loss function always yields a *static gesture* that does not move over multiple frames, as shown in Figure 8(d). Further investigation reveals that for most of our hand gestures, like finger friction and typing, only a few hand joints move, while most remain relatively static. Because the MSE loss function treats all joints equally, the predominantly static joints begin to dominate the training process. As a result, the model converges into a static gesture with the mostly unmoving joints.

To overcome the limitations of the MSE loss above, we reformulate it to prioritize the small subset of moving joints as follows:

$$\mathcal{L}_{\text{weighted\_MSE}} = \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{J} w_j \cdot \|p_{j,t} - g_{j,t}\|_F^2 \tag{2}$$

where w_j is a dynamic weight function for joint j defined below:

$$\begin{aligned} w_j &= \frac{\sum_{t=2}^{T} d_{j,t}^{\text{gt}}}{\sum_{j=1}^{J}\sum_{t=2}^{T} d_{j,t}^{\text{gt}} + \epsilon} \\ d_{j,t}^{\text{gt}} &= \|g_{j,t} - g_{j,t-1}\|_F \end{aligned} \tag{3}$$

In these equations, $d_{j,t}^{\text{gt}}$ denotes the ground-truth displacement of joint j over frame $t-1$ and t. ϵ is a small constant to prevent division by zero. The weight w_j will grow when the joint j moves significantly, which allows the model to focus more on the moving joints than those relatively static joints during training. To further reinforce the model to learn the dynamics of each joint across multiple frames,

[3] $\|p_{j,t} - g_{j,t}\|_F = \sqrt{(p_{x,j,t} - g_{x,j,t})^2 + (p_{y,j,t} - g_{y,j,t})^2 + (p_{z,j,t} - g_{z,j,t})^2}$

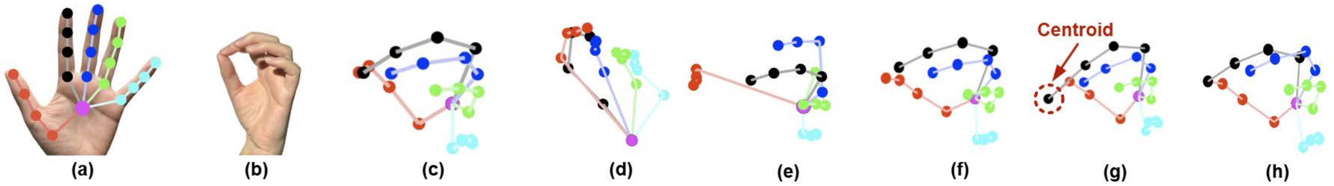

Figure 8: (a): the skeleton of the left hand represented by 21 joints. (b): a photo of a human gesture. (c): the ground-truth joint positions of gesture shown in (b) captured by leap motion. (d): the static gesture recovered by using the plain MSE loss $\mathcal{L}_{\text{MSE}}$. (e): the gesture motion generated by the model with combined weighted MSE loss term and the displacement term (Equation 6). (f): the gesture motion corrected by the IK layer. (g): the fingertips corrected using centroid calculation. (h): the final output gesture recovered by the model.

we introduce another loss term $\mathcal{L}_{\text{displacement}}$ defined below:

$$\mathcal{L}_{\text{displacement}} = \frac{1}{J(T-1)} \sum_{j=1}^{J} \sum_{t=2}^{T} (d_{j,t}^{\text{pred}} - d_{j,t}^{\text{gt}})^2 \tag{4}$$

where $d_{j,t}^{\text{pred}}$ denotes the predicted displacement of joint j over frame $t-1$ and t, which is defined as follows:

$$d_{j,t}^{\text{pred}} = \|p_{j,t+1} - p_{j,t}\|_F \tag{5}$$

This new displacement loss function $\mathcal{L}_{\text{displacement}}$ allows the model to learn the motion of each finger joint based on the FMCW frame input. The loss function $\mathcal{L}_{\text{final}}$ is a weighted combination of these two loss functions defined below:

$$\mathcal{L}_{\text{final}} = w_1 \cdot \mathcal{L}_{\text{weighted\_MSE}} + w_2 \cdot \mathcal{L}_{\text{displacement}} \tag{6}$$

where w_1=0.9 and w_2=0.1 are two empirical hyper-parameters that balance the position and motion recovery accuracy during training. These two loss terms are crucial for accurate gesture reconstruction.

3.3.2 Embrace the inverse kinematic constraint. While the customized loss function enables the model to generate gesture motions, we found that the produced gesture often moves in unnatural ways. For instance, as shown in Figure 8(e), the model can now effectively learn the relative position of each fingertip; however, the rotation of certain joints relative to the palm and the bone length still remain incorrect. To address this issue, we further propose to add an inverse kinematics (IK) layer to correct the model output.

The IK layer utilizes singular value decomposition and matrix operations to derive the position and orientation differences between connected joints, and then adjust each joint's rotation to adhere to human kinematic constraints. Besides, the configuration of IK layer's skeletal structure is defined to align with Leap Motion's 21 joints. The adjusted joint positions are then compared with the ground-truth coordinates to calculate the loss. However, incorporating the IK layer to the early stages of training can make it difficult for the model to learn natural hand motion trends, leading to slow or poor convergence. To address this, we propose to integrate the IK layer to the later stages of training, *i.e.*, after the MLP computes initial joint positions. We gradually add weights to the IK layer so that the influence of the IK constraints increases progressively with each epoch of training, allowing the model to first learn gesture patterns without strict ergonomic constraints and then refine joint movements as the training progresses. Figure 8 (f) shows the recovered hand skeleton after applying IK layer. We can see that by effectively constraining the rotation angle physically, our metacarpal joints align naturally relative to the palm (taking Figure 8(e) as a reference), enabling an inward wrapping gesture.

3.3.3 Gesture motion refinement using acoustic clues. Most gesture motions can be successfully reconstructed with the proposed model and IK layer so far. However, for certain fine-grained gestures with subtle movements, like finger friction for volume control shown in 8(b), we observed that LeakyFeeder often reconstructs the fingertips of the thumb and index finger without contacting each other. This issue arises because the model predicts joint positions rather than actual fingertip skins, making it challenging to determine the precise distance between fingertips across individuals. Such inaccuracies in gesture motion reconstruction can negatively impact user experience when displayed on the VR screen.

To refine these inaccurate gesture motions, we are inspired by an acoustic clue – the natural contact of fingertips in these gesture motions typically produces a sound, which will be captured by the feed-forward microphone. More importantly, the natural sound produced by human gestures (*e.g.*, friction) is a broad-band acoustic signal reaching both the audible and inaudible band [65, 74], as shown in Figure 9(a). The Helmholtz resonance effect of the microphone [28] further amplifies the higher frequency component of this finger-generated sound, making it more detectable. In contrast, since headphone speakers typically compress music to frequencies below 15 kHz [17], they will not interfere with the ultrasonic components of finger touch or friction sounds. This allows us to precisely identify the interval at which these sounds occur.

We further emulate four different application scenarios by playing sound clips collected in a lab, a cafe, an outdoor street, and in a chatting scenario with a loudspeaker to understand their impact on the reception of this fingertip sound. As shown in Figure 9(b–e), we observe that the noise does not overlap with the high-frequency component of the fingertip sound, indicating that that LeakyFeeder can reliably take the fingertip sound to refine the gesture construction.

Based on this observation, we propose a gesture motion refinement design that works in two steps. In this first step, we aim to detect finger friction sound or touching sound reliably. Due to the short period, rapid oscillation, and relatively low intensity of high-frequency friction signals compared to FMCW probing signals, traditional envelope detection often smooths out these quick fluctuations, making it challenging to capture small amplitude changes accurately. To address this issue, we use spectral entropy to detect hand friction signals. This approach is effective because the FMCW signal is a

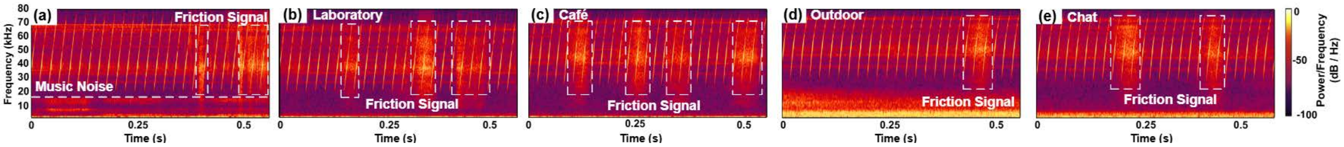

Figure 9: (a) Spectrogram of friction sound, music playback, and FMCW probing signals (18–78 kHz). (b–e) Robustness of fingertip contact sounds in four real-world environments: (b) Lab, (c) Cafe, (d) Outdoor, and (e) Chat.

linear chirp with concentrated spectral organization, relatively uniform spectral energy distribution in the 34-36kHz frequency band, and thus, low spectral entropy. In contrast, the friction signal is a broadband frequency range that exhibits dispersed and random characteristics, resulting in a significantly higher spectral entropy. Following this, we perform spectral entropy analysis using a sliding window of 44ms over this frequency band. Specifically, we first compute the power spectral density (PSD) of the filtered signal as $PSD(f) = |X(f)|^2$. Then, we normalize the power spectral density $p(f) = \frac{PSD(f)}{\sum_f PSD(f)}$ within each frequency band, and finally, we calculate the spectral entropy $H = -\sum_f p(f) \log_2(p(f))$. To determine friction events, we set a threshold $H_{\text{threshold}} = 5.25$ bits, empirically derived from training data. When $H > H_{\text{threshold}}$, we label that frame as 1, indicating friction; otherwise, we label it as 0.

In the second step, we use the contact label from the first step to identify friction events and refine joint positions based on thumb-to-fingertip distances. After obtaining the final joint outputs, we calculate the distance between the thumb and the other four fingertips. Given the thumb's role in stability and grip, friction events typically occur between its distal joint and nearby fingertips [5]. For fingertips close to the thumb, we compute a centroid coordinate and update their distal positions to this shared location, ensuring realistic finger alignment. However, we observe that the hand motion primarily involves movements of the finger's intermediate and metacarpal joints. The newly generated coordinates for these joints resulted in asymmetrical bone length ratios, as shown in Figure 8(g). To address this, we calculate the proportional variance in the joint coordinates before and after the update, adjusting the 3D coordinates of the corresponding finger joints accordingly. This adjustment ensures that all fingers exhibit natural bending behavior, achieving a realistic hand shape, as depicted in the final hand skeleton of Figure 8(h).

4 IMPLEMENTATION

This section introduces hardware setup (§4.1), training process (§4.2), probing signal (§4.3), and data collection (§4.4).

4.1 Hardware Setups

Figure 10 depicts the hardware setup. The Pixel Bud's speaker and feed-forward microphone are connected to an external Babyface AD/DA converter, which samples the audio signals at 192kHz[4]. Prior work [7, 65] has demonstrated that the commodity mobile device can support up to 192 kHz audio sampling rate with firmware modifications [7]. A Leap Motion [26] infrared vision sensor is placed below the hand to capture gesture motions at sub-millimeter accuracy. Although Leap Motion and our headphone SONAR system captures gesture motion from different angles, the forward linear layers in our model can effectively learn this coordinate system transformation, allowing it to output results in the Leap Motion coordinate system. Model training and testing are conducted on a server equipped with an NVIDIA A100 GPU.

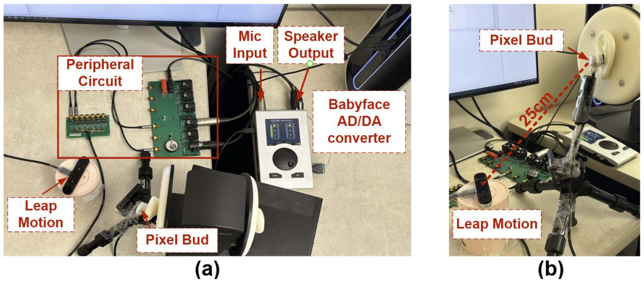

Figure 10: Hardware setup: (a) top view, (b) left-side view.

4.2 Software Implementation

We call MATLAB recording function for capturing FMCW audio data, while the Leap Motion joint data, including 3D coordinates of 21 hand joints as defined by our model, is simultaneously collected via the official Python API [59]. All joint coordinates are then normalized for consistency. We then employ a dynamic down-sampling strategy to capture 225 frames, prioritizing high-variance hand movements based on ground-truth data while reducing the frames for static hand positions. We do not use a segmentation algorithm to separate our different gesture ranging signals; instead, we use a 5-second synchronized audio signal alongside data collected by Leap Motion. We implemented the model in PyTorch, training it with the Adam optimizer at a learning rate of 0.0001, a batch size of 64, and over 100 epochs. The Transformer model is configured with 8 attention heads, 6 encoder layers, and a dropout rate of 0.1 to capture spatial-temporal dependencies while ensuring generalization.

4.3 FMCW Probing Signals

Similar to prior works [34, 35], we add fade-in and fade-out effects to FMCW signals to eliminate perceptible artifacts at each chirp transition, ensuring that the probing signal remains unnoticeable to users during operation. In our settings, we define the duration of one chirp as 4096 sampling points:

$$\text{single\_chirp\_length} = 4096 = F_s \times T$$

where $F_s = 192\,\text{kHz}$ is the sampling frequency; $T = \frac{4096}{F_s}$ is the chirp duration. The joint-to-earphone distance R can be directly

[4] Although LeakyFeeder is compatible with all ANC earphones, API limitations currently restrict real-world testing to the Google Pixel Buds prototype.

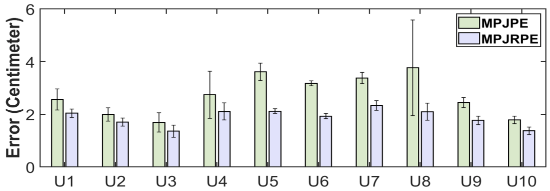

Figure 11: Overall performance across ten users.

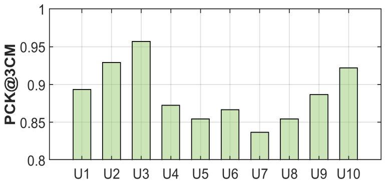

Figure 12: PCK across ten users.

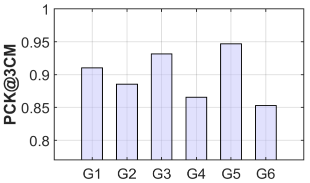

Figure 13: PCK across six different gestures.

determined from the beat frequency f_b, obtained as the primary result of the first FFT performed on the Intermediate Frequency (IF) signal, which represents the frequency difference between the transmitted and reflected signals:

$$R = \frac{V_s \times T \times f_b}{2B}$$

where $V_s = 343\,\text{m/s}$ is the speed of sound; $B = 60\,\text{kHz}$ is the chirp bandwidth, with the central frequency of 48 kHz. The FMCW ranging algorithm enables the measurement of both target-to-transmitter distances and displacement changes.

4.4 Data Collection

We adopt the widely used gesture definitions from the Apple Vision Pro standard [42] to define six gestures for system evaluation, as shown in Figure 6. These six gestures cover both fine-grained (*e.g.*, rubbing) and coarse-grained (*e.g.*, tapping, pinching, and sliding) gesture motions. Besides, the users are free to define their own gestures as described in Section 5.5. For data collection, we use the dummy head testbed (Figure 10) to record each of the six gestures, performed 100 times by a volunteer for training. Then, following the same training strategy principle in [30], we randomly divided the dataset, allocating 90% for training and 10% for validation. For evaluation, we invite ten participants to wear the Pixel Buds and perform each gesture ten times around their ears. We set the speaker volume to 82dB(A) and inform participants to keep the hand-to-headphone distance below 10cm while performing the gestures.

5 EVALUATION

We employ the following three metrics for evaluation:

- **Mean Per Joint Position Error (MPJPE)**: MPJPE is the average Euclidean distance between estimated key points p_j and ground-truth points g_j:

$$\text{MPJPE} = \frac{1}{J}\sum_{j=1}^{21}\left\|p_j - g_j\right\|_2 \tag{7}$$

- **Mean Per Joint Relative Position Error (MPJRPE)**: MPJRPE is the average Euclidean distance between estimated and ground-truth joint coordinates after aligning the palm root (0-th joint):

$$\text{MPJRPE} = \frac{1}{J}\sum_{j=1}^{21}\left\|(p_j - p_0) - (g_j - g_0)\right\|_2 \tag{8}$$

  where p_0 and g_0 represent the central palm position for measuring the relative structure of the hand pose.

- **Percentage of Correct Keypoints (PCK)**: PCK [73] is a gold-standard metric for evaluating keypoint tracking accuracy. A predicted joint is considered correct if the Euclidean distance between it and the ground-truth joint is within a specified threshold:

$$\text{PCK@a} = \frac{1}{J}\sum_{j=1}^{21}\delta\left(\left\|(p_j - p_0) - (g_j - g_0)\right\|_2 \le \alpha D\right) \tag{9}$$

  where δ is a logical operation that outputs 1 if true and 0 if false. α represents a threshold fraction, typically set at 0.2 by default[5], and D represents the maximum side length of the bounding box of the hand, which is 15cm in our experiments. The higher value of PCK is better.

5.1 Overall Performance

We first conduct field studies to evaluate LeakyFeeder.

Performance Across Different Users. We recruited ten participants (2 female and 8 male) under IRB approval, aged between 20 and 30 years to evaluate our LeakyFeeder. Participants are instructed to wear the Google Pixel Buds, positioned identically to those on the dummy head setup shown in Figure 10. Figure 11 presents the MPJPE and MPJRPE of LeakyFeeder across ten users, averaging 2.71cm ± 0.71cm (MPJPE) and 1.88cm ± 0.31cm (MPJRPE) over all joint positions. However, some results indicate higher tail errors and increased variability for certain users. For instance, as shown in Figure 12, substantial variance is observed for subject 8, likely due to her longer fingernails, which partially occlude fingertip movements during the experiment. Notably, user 7 (U7) achieves a PCK of 83%, as his unusual finger rotations, particularly abnormal bending of the pinky finger during movements, contribute to increased error rates. Overall, these results reaffirm the overall effectiveness of LeakyFeeder across various users.

Performance Across Different Gestures. Figure 13 shows PCK@3CM performance of six gestures. G1: Single Tap; G2: Double Tap; G3: Pinch and Hold; G4: Pinch and Slide; G5: Rub; and G6: Whole Hand Close. We observe that the PCK results for all six gestures exceed 85%, with an average PCK of 90%. Specifically, we find that when users perform G5, the friction sound generated during this gesture contributes to the formation of the contact label, resulting in more precise gesture reconstruction (95%). However, for G4, we find that some participants drag and drop too far, which their gestures are beyond the sensing range of our headphone SONAR system. Additionally, fingers merging in G6 is relatively complex; for example, the thumb and pinky finger joints may become occluded, resulting in a decreasing PCK to 85%.

We show a few snapshots of reconstructed rubbing gesture in Figure 14. The rubbing gesture starts with the thumb and index finger making contact, then moving back and forth with light pressure to create friction, allowing for tactile feedback and precise control. This motion is often used for actions like scrolling or adjusting

[5] PCK@0.2 (α = 0.2), which strikes a balance between strictness and practical usability, making it suitable for most gesture estimation tasks [73].

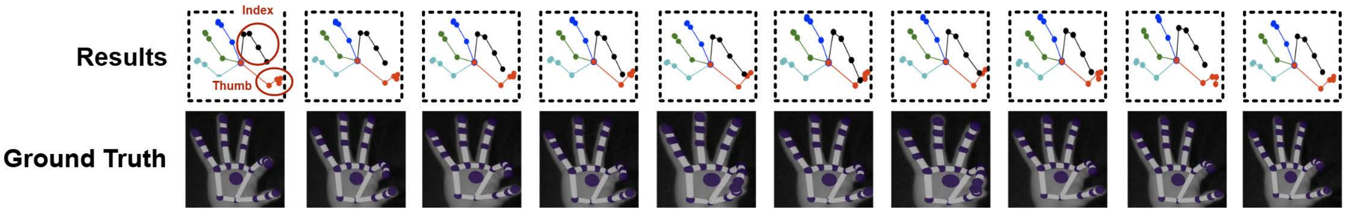

Figure 14: Example of a generated rubbing gesture (Figure 6(f)) compared to its ground truth across 10 frames. We provide more results to illustrate reconstructed gestures at: https://leekyfeeder.github.io/results

Table 1: Ablation Study. We evaluate the contribution of each design component and list it in this table.

	MPJPE (cm)	**MPJRPE (cm)**	**PCK@3cm**
mm4ARM [41] + $\mathcal{L}_{\text{weighted\_MSE}}$ + $\mathcal{L}_{\text{displacement}}$ + IK layer + audio clue	3.88	2.68	0.74
Transformer + $\mathcal{L}_{\text{weighted\_MSE}}$	4.75	4.13	0.67
Transformer + $\mathcal{L}_{\text{weighted\_MSE}}$ + $\mathcal{L}_{\text{displacement}}$	3.52	2.42	0.76
Transformer + $\mathcal{L}_{\text{weighted\_MSE}}$ + $\mathcal{L}_{\text{displacement}}$ + IK layer	2.97	2.02	0.86
Transformer + $\mathcal{L}_{\text{weighted\_MSE}}$ + $\mathcal{L}_{\text{displacement}}$ + IK layer + audio clue (LeakyFeeder)	**2.71**	**1.88**	**0.89**

volume. it is the most fine-grained gesture among all six gestures that LeakyFeeder reconstructed. Compared to the ground truth obtained through Leap Motion, we observe that LeakyFeeder can accurately recover the gesture motion, with most fingers remain in fixed positions, while thumb and the index finger gradually touch each other, moving back and forth, and then open. Videos can be found at the link: https://leekyfeeder.github.io/results.

5.2 Ablation Study

To quantify the individual contributions of our design components within the model, we conduct an ablation study exploring the following configurations:

(1) **Compared to a baseline mm4ARM [41]**: We replace our transformer backbone proposed by a SOTA solution mm4ARM, retaining all other LeakyFeeder's components.
(2) **Weighted Position Loss Only**: The model is trained with only $\mathcal{L}_{\text{weighted\_MSE}}$. This setup tests the model's performance when only minimizing the weighted position error, without accounting for motion continuity.
(3) **Final Loss $\mathcal{L}_{\text{final}}$**: The model is trained by adding both the weighted loss $\mathcal{L}_{\text{weighted\_MSE}}$ and displacement loss factor $\mathcal{L}_{\text{displacement}}$. This variant assesses performance when prioritizing motion consistency.
(4) **Final Loss + IK Layer Constraint**: We combine final loss and the IK layer to evaluate reconstruction performance.
(5) **Final Loss + IK Layer Constraint + Audio Clues**: We add motion sound to generate a contact label input.

For each baseline listed in Table 1, we conduct five training runs and run inference on ten users' data to evaluate the model's performance across average MPJPE, MPJRPE, and PCK metrics. The mm4ARM framework employs a widely adopted encoder-decoder architecture with ResNet layers. We modify its input to process raw FMCW acoustic data. Comparing mm4ARM and LeakyFeeder, we can see that using transformer improves the MPJPE, MPJRPE, and PCK@3cm from 3.88cm, 2.68cm, and 74% to 2.71cm, 1.88cm, and 89%, respectively. This improvement is attributed to the Transformer's ability to effectively process temporal information.

In evaluating our design components, we analyze the impact of each loss function, the IK layer, and friction-based acoustic cues. The results indicate that applying only the $\mathcal{L}_{\text{weighted\_MSE}}$ yields PCK@3CM about 67%, as the model primarily learns basic hand positions with temporal variance. By adding $\mathcal{L}_{\text{displacement}}$ to smooth the gesture performance over time, accuracy improves to 76%. Meanwhile, the MPJPE and MPJRPE also decreases from 4.75cm and 4.13cm to 3.52cm and 2.42cm, respectively. These results confirm that calculating dynamic displacement variance greatly aids the model in learning variations in hand joint positions across preceding and following frames. To ensure natural hand motion and realistic joint constraints, we further incorporate IK layer, which increases PCK@3cm to from 76% to 86%. The MPJPE and MPJRPE also decreases further to 2.97cm and 2.02cm, respectively. Although we witness marginal gain (e.g., PCK@3cm grows from 86% to 89% and MPJPE and MPJRPE delines to 2.71cm and 1.88cm) when adding the audio clues, this part is still indispensible because it helps refine those fine-grained gestures with subtle movements like finger friction, enforcing fingers to touch each other. Compared to the recent SOTA WiFi-based hand tracking method [30], LeakyFeeder performs slightly worse due to its single SONAR hardware constraint.

5.3 Benchmark

We also conduct four benchmarks to understand the impact of various factors on LeakyFeeder's performance.

① **Impact of Speaker Volume:** The intensity of the speaker volume determines the level of leaky signal emitted. We invite a volunteer to conduct this experiment in a controlled laboratory environment. As shown in Figure 10, the volunteer performed six gestures near the dummy head ear under four different sound levels: 25%, 50%, 75%, and a maximum level of 100% (82dB(A)). The results, clearly depicted in Figure 15, show consistently small errors (less than 3cm) at volume levels above 75%. Conversely, at a volume level of 50%,

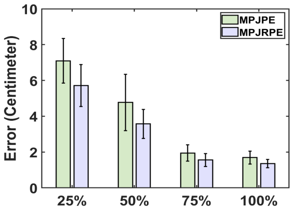

Figure 15: Impact of speaker volume.

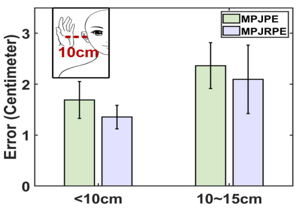

Figure 16: Impact of hand-to-headphone distance.

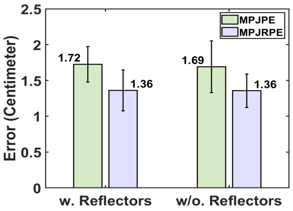

Figure 17: Impact of background reflection.

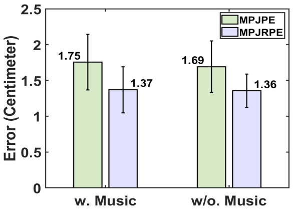

Figure 18: Impact of music playback.

Table 2: Model Overhead measured on a Google Pixel 9.

Memory Usage	Latency	Power Consumption
95MB	21.62ms	1504mAh

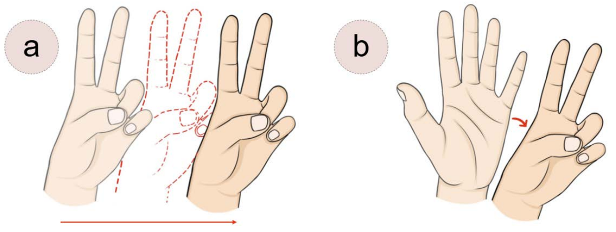

Figure 19: Two other user defined gestures. (a) a two-finger sliding gesture; (b) a V sign gesture.

the MPJPE and MPJRPE respectively show errors of 4.7cm and 3.6cm, as accuracy declines due to reduced leaky signal strength, diminishing the SNR. At 25% volume level, the MPJRPE increases to 7cm, since only a limited portion of the fingertips can be reliably detected under the conditions of a very weak leaky signal. In conclusion, maintaining the speaker volume at or above 75% level would have an optimal system performance.

② **Impact of Hand-to-headphone Distance:** Next, we set the sound volume to 82dB(A) to evaluate the gesture motion reconstruction performance at different hand-to-headphone distance settings. We categorize the sensing range into two intervals: *<10cm* and *10-15cm*. Measurements beyond 15cm are not taken into account because we observe that it's more natural to perform gestures close to the ear rather than stay far away from ears. Figure 16 shows the result. We observe that the reconstruction error declines with the increasing hand-to-headphone distance. Specifically, MPJPE rises from 1.7cm at *<10cm* distance setup to 2.4cm in the *10-15cm* distance setup. Likewise, MPJRPE also grows from 1.3cm to 2.1cm as we expand the hand-to-headphone distance to above 15cm. This degradation is expected due to the attenuation of the leaky signal over a larger hand-to-headphone distance. Overall, an effective sensing range of up to 15cm is achieved, covering the human hand and aligning with natural user behavior.

③ **Impact of Background Reflection:** To examine the impact of complex real-world environments with dynamic reflections, a second individual is invited to pass by the first participant while she is performing gestures. The sound volume is set to 82dB(A). As shown in Figure 17, the system maintains consistent errors below 2cm for MPJPE and 1.5cm for MPJRPE, regardless of the presence and absence of this passing by participant. This resilience demonstrates LeakyFeeder's reliable performance, effectively minimizing the impact of weaker multi-path reflections in the complex environment and reducing the likelihood of false positives.

Table 3: MPJPE, MPJRPE, and PCK@3cm results for two user-defined gestures.

Gestures	MPJPE (cm)	MPJRPE (cm)	PCK@3cm
Two-finger sliding	2.56	1.50	0.95
V sign	1.79	1.48	0.92

④ **Impact of Music Playback:** To assess the impact of music, we invite a volunteer to perform six gestures while listening to music at the same sound level as the FMCW probing signal. The speaker volume is 82 dB(A) and the hand-to-headphone detection distance is below 10cm. As shown in Figure 18, the errors with and without music are similar, with mean MPJPE 1.7cm and MPJRPE 1.35cm. The probing signal operates in the ultrasound range, while the artificial music played through the speakers is limited to frequencies below 15kHz. These results confirm LeakyFeeder's ability to reconstruct gestures in the presence of music.

5.4 Model Overhead

Our Transformer model comprises 51 million parameters. We convert it to ONNX and deployment on a Google Pixel 9 [2] to measure its system overhead. As detailed in Table 2, the model requires 95MB of memory for inference. The average inference delay (on gesture reconstruction) is 21.62ms. Running the transformer-based gesture reconstruction model, on the other hand, would consume around 1504mAh power, nearly one-third of the total battery of the testing phone. Hence edge-based deployment might be a more promising solution and we leave it for future explorations.

5.5 Extend to Other User-Defined Gestures

LeakyFeeder also supports user-defined gestures for recognition. To demonstrate the feasibility, we select one gesture that received the highest ratings in the EarHover's [55] user study, as shown in Figure 19(a), and another widely recognized gesture from real-world interactions [43], illustrated in Figure 19(b). A volunteer is asked to perform each gesture 100 times. We then use 90% of the data for training and 10% of data for testing. The results are shown in Table 3. LeakyFeeder achieves PCK@3cm scores of 95% and 92%, MPJRPE of 1.50cm and 1.48cm, and MPJPE of 2.56cm and

1.79cm, on these two gestures, respectively. These promising results clearly demonstrate the remarkable potential of our system in future user-defined gesture controls.

6 RELATED WORK

Our work relates to a broad spectrum of research in hand gesture reconstruction and leaky signal applications. In this section, we review related works in both categories.

Hand Gesture Reconstruction: Hand gesture reconstruction has significantly improved through various sensing technologies. Guo *et al.* [22] summarize these works. Below we divide the latest works in this domain into three categories.

(1) Vision-based: Recent advancements in deep learning have significantly improved the processing of image data for hand gesture sensing, enabling methods like as seen in Hand3D [12] and hand image understanding (HIU) [72]. While these approaches utilize the rich semantic information in images for precise tracking, they raise privacy concerns for on-body applications. Moreover, EarAuthCam [46] discusses a method for user authentication that utilizes a small camera attached to earphones. Although this setup could potentially be adapted for gesture reconstruction, the cameras on earphones are highly sensitive to individual anatomy, hair coverage, positioning, and lighting conditions. These factors limit reliability, especially in low light or glare. Given these inherent constraints, vision-based sensing on earbuds remains fundamentally limited in both robustness and practicality.

(2) RF-based: RF-based hand gesture reconstruction has emerged as a promising alternative, demonstrating promising results based on various sensing modalities such as mm-wave radar [39, 41] and WiFi [30, 56, 63, 71]. However, these systems often require complex, high-cost hardware and specific device setups, reducing their flexibility for on-body wearable applications. Unlike these methods, LeakyFeeder uses a compact ultrasound system on commercial headphones, enabling 3D hand reconstruction with a minimal setup, without the constraints of specialized hardware.

(3)Acoustic-based: Some ultrasound-based systems have been developed for gesture recognition [55, 62, 68, 69]. Among these, [55] is the most similar to our work; however, it only performs coarse gesture classification rather than fine-grained reconstruction. In contrast, LeakyFeeder advances beyond classification to achieve detailed 3D hand pose reconstruction.

Leaky Signal Applications. The sound leakage in earable devices [9, 10] has become an active area of research, with recent studies demonstrating that earphones and other wearables can capture leakage signals for various applications, including interaction [55, 68], user authentication [4], and expression recognition [3, 14, 16, 49]. While previous works have leveraged leakage signals from earables for gesture recognition, such as EarHover [55] and MAF [68], these studies capture only Doppler shifts for high-level gesture classification and lack the capability for fine-grained hand reconstruction. To the best of our knowledge, no prior work has explored whether leakage signals can achieve the detailed granularity of traditional acoustic sensing for interactive tasks. We are the first to investigate this potential, aiming to determine whether these signals can support detailed hand keypoint reconstruction comparable to those sensing systems that rely on dedicated, costly, or bulky hardware platforms.

7 DISCUSSION

Single-pair of speaker and microphone: Our earphone-based SONAR system contains only a single pair of speaker and microphone, which poses challenges in obtaining angular information of gesture reflections. In the future, we plan to investigate the feasibility of integrating both the feed-forward microphone and the talking microphone to develop a multi-receiver SONAR system. This design would enhance spatial diversity, allowing us to better isolate reflections from different hand parts.

Impact of human motions: Our current evaluations are conducted in static scenarios where the mobile user performs gestures while staying in a fixed position. This fixed positioning allows the model to learn the translation within a consistent coordinate system during training. LeakyFeeder encounters challenges when the coordinate system shifts due to head movements during walking. A potential solution is to leverage IMU readings from earphones to estimate the head movement and further mitigate its impact on hand gesture recognition. We leave it as other future work.

Potential model over-fitting: As presented in Section 5, our model design yields promising results. However, the Transformer model faces a potential risk of overfitting due to dataset limitations. First, the limited number of gesture samples restricts the model's ability to generalize, especially for fine-grained finger gestures, where occlusions from crossed fingers add complexity. Second, the dataset lacks demographic diversity, including variations in hand structures and the absence of data from users with disabilities (e.g., missing fingers), which may impact model robustness. Third, our implementation is currently restricted to Google Pixel Buds, and deploying the model on other earbud hardware could introduce domain gaps due to variations in sensor processing capabilities. To address these challenges, future work could explore data augmentation techniques, domain adaptation strategies, and meta-learning approaches to enhance cross-user and cross-device generalization. Adopting a leave-one-out evaluation method could provide deeper insights into the model's adaptability across diverse user profiles, ultimately improving its reliability for real-world deployment.

8 CONCLUSION

We have introduced LeakyFeeder, a novel earable sensing system for in-air gesture motion reconstruction. LeakyFeeder repurposes the headphones' speaker and feed-forward microphone as a single-receiver SONAR system, and then tracks FMCW echos to reconstruct 3D hand gesture motion through a deep learning model. We believe this work demonstrates the untapped potential of leaky acoustic waves for fine-grained gesture motion reconstruction, paving the way for new applications in human-device interactions.

ACKNOWLEDGMENT

We sincerely appreciate the insightful feedback from the anonymous reviewers and the shepherd. This material is based upon work supported by the National Science Foundation (NSF) under Grant No.2337537. We also thank the University of Pittsburgh Center for Research Computing (CRC) for providing us the computing resource. Specifically, this work used the HTC and GPU cluster, which are supported by National Institutes of Health (NIH) under Grant S10OD028483 and NSF award number OAC-2117681.

REFERENCES

[1] Babyface pro fs - rme audio interfaces. https://rme-audio.de/babyface-pro-fs.html. Accessed: 2024-10-23.

[2] Google pixel 9 specifications. https://store.google.com/product/pixel_9_specs. Accessed: 2025-02-16.

[3] T. Amesaka, H. Watanabe, M. Sugimoto. Facial expression recognition using ear canal transfer function. ISWC '19. Association for Computing Machinery, 2019.

[4] T. Amesaka, H. Watanabe, M. Sugimoto, Y. Sugiura, B. Shizuki. User authentication method for hearables using sound leakage signals. ISWC '23. Association for Computing Machinery, 2023.

[5] I. Birznieks, P. Jenmalm, A. W. Goodwin, R. S. Johansson. Encoding of direction of fingertip forces by human tactile afferents. *Journal of Neuroscience*, 2001. doi:10.1523/JNEUROSCI.21-20-08222.2001.

[6] J. Cao, Y. Liu, L. Han, Z. Li. Finger tracking using wrist-worn emg sensors. *IEEE Transactions on Mobile Computing*, 2024.

[7] S. Cao, D. Li, S. I. Lee, J. Xiong. Powerphone: Unleashing the acoustic sensing capability of smartphones. *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, 2023.

[8] T. Chen, L. Shangguan, Z. Li, K. Jamieson. Metamorph: Injecting inaudible commands into over-the-air voice controlled systems. *Network and Distributed Systems Security (NDSS) Symposium*, 2020.

[9] T. Chen, Y. Yang, X. Fan, X. Guo, J. Xiong, L. Shangguan. Exploring the feasibility of remote cardiac auscultation using earphones. *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 357–372, 2024.

[10] T. Chen, Y. Yang, C. Qiu, X. Fan, X. Guo, L. Shangguan. Enabling hands-free voice assistant activation on earphones. *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services*, 155–168, 2024.

[11] D. G. Ćirić, D. Hammershøi. Coupling of earphones to human ears and to standard coupler. *The Journal of the Acoustical Society of America*, **120**(4), 2096–2107, 2006.

[12] X. Deng, S. Yang, Y. Zhang, P. Tan, L. Chang, H. Wang. Hand3d: Hand pose estimation using 3d neural network. *arXiv preprint arXiv:1704.02224*, 2017.

[13] C. for Disease Control, Prevention. Too loud! for too long!, 2024. Accessed: 2024-10-21.

[14] D. Duan, Z. Sun, T. Ni, S. Li, X. Jia, W. Xu, T. Li. F2key: Dynamically converting your face into a private key based on cots headphones for reliable voice interaction. *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services*, 127–140, 2024.

[15] T. Fan, H. Wu, M. Jin, T. Chen, L. Shangguan, X. Wang, C. Zhou. Towards spatial selection transmission for low-end iot devices with spotsound. *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, 1–14, 2023.

[16] X. Fan, L. Shangguan, S. Rupavatharam, Y. Zhang, J. Xiong, Y. Ma, R. Howard. Headfi: bringing intelligence to all headphones. *Proceedings of MobiCom*, 2021.

[17] H. Fletcher, W. A. Munson. Loudness, its definition, measurement and calculation. *Bell System Technical Journal*, **12**(4), 377–430, 1933.

[18] D. Gavgiotaki, S. Ntoa, G. Margetis, K. C. Apostolakis, C. Stephanidis. Gesture-based interaction for ar systems: a short review. *Proceedings of the 16th International Conference on PErvasive Technologies Related to Assistive Environments*, 284–292, 2023.

[19] H.-T. Geek. What is sound leakage?, 2021. Accessed: 2025-02-10.

[20] A. Glandon, L. Vidyaratne, N. Sadeghzadehyazdi, N. K. Dhar, J. O. Familoni, S. T. Acton, K. M. Iftekharuddin. 3d skeleton estimation and human identity recognition using lidar full motion video. *2019 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE, 2019.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. Generative adversarial networks. *Communications of the ACM*, 2020.

[22] L. Guo, Z. Lu, L. Yao. Human-machine interaction sensing technology based on hand gesture recognition: A review. *IEEE Transactions on Human-Machine Systems*, **51**(4), 300–309, 2021.

[23] Headphonesty. What is sound leakage?, 2021. Accessed: 2025-02-10.

[24] HIFIMAN Corporation. Deva pro-wired. https://www.hifiman.com/products/detail/323, 2024. Accessed: 2024-10-23.

[25] Y. Huang, M. Kaufmann, E. Aksan, M. J. Black, O. Hilliges, G. Pons-Moll. Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Transactions on Graphics (TOG)*, 2018.

[26] Inc. Leap Motion. Leap motion overview.

[27] Infineon Technologies AG. *FMCW Radar Digital Signal Processing Handout*, 2023.

[28] Infineon Technologies AG. MEMS Microphone Mechanical & Acoustical Implementation. Tech. Rep. AN557, Infineon Technologies AG, 2023. Infineon XENSIV$^{\text{TM}}$ MEMS Microphone Customers.

[29] S. Islam, H. Elmekki, A. Elsebai, J. Bentahar, N. Drawel, G. Rjoub, W. Pedrycz. A comprehensive survey on applications of transformers for deep learning tasks, 2023.

[30] S. Ji, X. Zhang, Y. Zheng, M. Li. Construct 3d hand skeleton with commercial wifi. *Proceedings of the 21st ACM Conference on Embedded Networked Sensor Systems*, 322–334, 2023.

[31] D. P. Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[32] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, O. Winther. Autoencoding beyond pixels using a learned similarity metric. *International conference on machine learning*, 1558–1566. PMLR, 2016.

[33] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, *et al.* Photo-realistic single image super-resolution using a generative adversarial network. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4681–4690, 2017.

[34] D. Li, S. Cao, S. I. Lee, J. Xiong. Experience: practical problems for acoustic sensing. *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, 381–390, 2022.

[35] D. Li, J. Liu, S. I. Lee, J. Xiong. Room-scale hand gesture recognition using smart speakers. *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, 462–475, 2022.

[36] H. Li, W. Yang, J. Wang, Y. Xu, L. Huang. Wifinger: Talk to your smart devices with finger-grained gesture. *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 250–261, 2016.

[37] P. P. Liang, A. Zadeh, L.-P. Morency. Foundations & trends in multimodal machine learning: Principles, challenges, and open questions. *ACM Computing Surveys*, **56**(10), 1–42, 2024.

[38] J. Lien, N. Gillian, M. E. Karagozler, P. Amihood, C. Schwesig, E. Olson, H. Raja, I. Poupyrev. Soli: Ubiquitous gesture sensing with millimeter wave radar. *ACM Transactions on Graphics (TOG)*, **35**(4), 1–19, 2016.

[39] H. Liu, Y. Wang, A. Zhou, H. He, W. Wang, K. Wang, P. Pan, Y. Lu, L. Liu, H. Ma. Real-time arm gesture recognition in smart home scenarios via millimeter wave sensing. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, **4**(4), 1–28, 2020.

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

[41] Y. Liu, S. Zhang, M. Gowda, S. Nelakuditi. 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), 1–28, 2022.

[42] MacRumors Staff. Apple vision pro to rely on eye movements and hand gestures for input, 2023. Accessed: 2024-11-09.

[43] T. Magazine. Why is everyone in asia making the v sign in photos? *TIME*, 2014. Accessed: 2025-02-16.

[44] E. Mansimov, E. Parisotto, J. L. Ba, R. Salakhutdinov. Generating images from captions with attention. *arXiv preprint arXiv:1511.02793*, 2015.

[45] W. Mao, J. He, L. Qiu. Cat: High-precision acoustic motion tracking. *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, 69–81, 2016.

[46] Y. Mizuho, Y. Kawasaki, T. Amesaka, Y. Sugiura. Earauthcam: Personal identification and authentication method using ear images acquired with a camera-equipped hearable device. *Proceedings of the Augmented Humans International Conference 2024*, 119–130, 2024.

[47] P. Molchanov, S. Gupta, K. Kim, J. Kautz. Hand gesture recognition with 3d convolutional neural networks. *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 1–7, 2015.

[48] Mysteek. Ath-em7 and ath-ew9 (or clip-ons in general), 2007.

[49] P. Panda, M. J. Nicholas, D. Nguyen, E. Ofek, M. Pahud, S. Rintel, M. Gonzalez-Franco, K. Hinckley, J. Lanier. Beyond audio: Towards a design space of headphones as a site for interaction and sensing. DIS '23, 2023.

[50] S. M. Patole, M. Torlak, D. Wang, M. Ali. Automotive radars: A review of signal processing techniques. *IEEE Signal Processing Magazine*, **34**(2), 22–35, 2017. doi:10.1109/MSP.2016.2628914. Discusses the application of FMCW radar in automotive systems.

[51] T. Pendlebury. Bang & olufsen a8 earphones review: Bang & olufsen a8 earphones, 2009. Accessed: 2024-10-08.

[52] Razer Inc. Razer kraken v4 x gaming headset. https://www.razer.com/gaming-headsets/razer-kraken-v4-x, 2024. Accessed: 2024-10-23.

[53] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, H. Lee. Generative adversarial text to image synthesis. *International conference on machine learning*, 1060–1069. PMLR, 2016.

[54] M. A. Richards. *Fundamentals of Radar Signal Processing*. McGraw-Hill Education, New York, 2nd edn., 2014. Chapter 10 covers FMCW radar in automotive applications.

[55] S. Suzuki, T. Amesaka, H. Watanabe, B. Shizuki, Y. Sugiura. Earhover: Mid-air gesture recognition for hearables using sound leakage signals. *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, 1–13, 2024.

[56] S. Tan, J. Yang, Y. Chen. Enabling fine-grained finger gesture recognition on commodity wifi devices. *IEEE Transactions on Mobile Computing*, **21**(8), 2789–2802, 2020.

[57] M. K. Tatarunis, Steve. Acoustic resistance, secret sauce for speakers, 2015.

[58] M. R. Torloni, N. Vedmedovska, M. Merialdi, A. P. Betrán, T. Allen, R. González, L. D. Platt. Safety of ultrasonography in pregnancy: Who systematic review of the literature and meta-analysis. *Ultrasound in Obstetrics & Gynecology*, **33**(5), 599–608, 2009. doi:https://doi.org/10.1002/uog.6328.

[59] Ultraleap. Leap motion controller 2 downloads, 2025. Accessed: 2025-02-10.

[60] O. Vinyals, A. Toshev, S. Bengio, D. Erhan. Show and tell: A neural image caption generator. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3156–3164, 2015.

[61] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, C. Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018.

[62] Y. Wang, J. Shen, Y. Zheng. Push the limit of acoustic gesture recognition. *IEEE Transactions on Mobile Computing*, **21**(5), 1798–1811, 2020.

[63] D. Wu, R. Gao, Y. Zeng, J. Liu, L. Wang, T. Gu, D. Zhang. Fingerdraw: Sub-wavelength level finger motion tracking with wifi signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, **4**(1), 1–27, 2020.

[64] J. Wu, C. Zhang, T. Xue, B. Freeman, J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 2016.

[65] Z. Xiao, T. Chen, Y. Liu, Z. Li. Mobile phones know your keystrokes through the sounds from finger's tapping on the screen. *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, 965–975. IEEE, 2020.

[66] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, X. He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1316–1324, 2018.

[67] D. Xue, X. Fan, T. Chen, G. Lan, Q. Song. Leveraging foundation models for zero-shot iot sensing. *arXiv preprint arXiv:2407.19893*, 2024.

[68] Y. Yang, T. Chen, Y. Huang, X. Guo, L. Shangguan. Maf: Exploring mobile acoustic field for hand-to-face gesture interactions. *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 1–20, 2024.

[69] Y. Yang, T. Chen, L. Shangguan. Toward next-generation human–computer interface based on earables. *IEEE Pervasive Computing*, **23**(4), 50–52, 2025.

[70] Y. Youm, T. Gillespie, A. Flatt, B. Sprague. Kinematic investigation of normal mcp joint. *Journal of Biomechanics*, 1978. doi:https://doi.org/10.1016/0021-9290(78)90003-9.

[71] N. Yu, W. Wang, A. X. Liu, L. Kong. Qgesture: Quantifying gesture distance and direction with wifi signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, **2**(1), 1–23, 2018.

[72] X. Zhang, H. Huang, J. Tan, H. Xu, C. Yang, G. Peng, L. Wang, J. Liu. Hand image understanding via deep multi-task learning. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11,281–11,292, 2021.

[73] F. Zhou, F. De la Torre. Spatio-temporal matching for human pose estimation in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **38**(8), 1492–1504, 2016.

[74] M. Zhou, S. Su, Q. Wang, Q. Li, Y. Zhou, X. Ma, Z. Li. Printlistener: Uncovering the vulnerability of fingerprint authentication via the finger friction sound. *arXiv preprint arXiv:2404.09214*, 2024.