# Towards Generalized mmWave-based Human Pose Estimation through Signal Augmentation

Hongfei Xue[1], Qiming Cao[2], Chenglin Miao[3], Yan Ju[1], Haochen Hu[1]
Aidong Zhang[4], Lu Su[2*]
[1]State University of New York at Buffalo, [2]Purdue University, [3]Iowa State University, [4]University of Virginia
Email: [1]{hongfeix, yanju, haochenh}@buffalo.edu, [2]{cao393, lusu}@purdue.edu, [3]cmiao@iastate.edu,
[4]aidong@virginia.edu

## ABSTRACT

The unprecedented advance of wireless human sensing is enabled by the proliferation of the deep learning techniques, which, however, rely heavily on the completeness and representativeness of the data patterns contained in the training set. Thus, deep learning based wireless human perception models usually fail when the human subject is conducting activities that are unseen during the model training. To address this problem, we propose a novel wireless signal augmentation framework, named mmGPE, for Generalized mmWave-based Pose Estimation. In mmGPE, we adopt a physical simulator to generate mmWave FMCW signals. However, due to the imperfect simulation of the physical world, there is a big gap between the signals generated by the physical simulator and the real-world signals collected by the mmWave radar. To tackle this challenge, we propose to integrate the physical signal simulation with deep learning techniques. Specifically, we develop a deep learning-based signal refiner in mmGPE that is capable of bridging the gap and generating realistic signal data. Through extensive evaluations on a COTS mmWave testbed, our mmGPE system demonstrates high accuracy in generating human meshes for unseen activities.

## CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile computing**; • **Computer systems organization** → **Embedded and cyber-physical systems**.

*Lu Su is the corresponding author.


*ACM MobiCom '23, October 2–6, 2023, Madrid, Spain*

ACM ISBN 978-1-4503-9990-6/23/10…$15.00
https://doi.org/10.1145/3570361.3613302

## KEYWORDS

Wireless Sensing, mmWave, Human Mesh Estimation, Signal Augmentation, Generative Neural Network

**ACM Reference Format:**
Hongfei Xue[1], Qiming Cao[2], Chenglin Miao[3], Yan Ju[1], Haochen Hu[1], Aidong Zhang[4], Lu Su[2]. 2023. Towards Generalized mmWave-based Human Pose Estimation through Signal Augmentation. In *The 29th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '23), October 2–6, 2023, Madrid, Spain.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3570361.3613302

## 1 INTRODUCTION

In recent years, significant efforts have been put towards building intelligent wireless sensing system, with the goal of perceiving and understanding human activities using pervasive wireless signals. Thus far, various wireless human sensing systems and applications have been proposed, demonstrating the advantages of RF signals over traditional camera-based solutions that are faced with various challenges, such as occlusions, poor lighting conditions and privacy issues.

The unprecedented advance of wireless human sensing is enabled by the proliferation of the deep learning techniques. However, deep learning based human perception solutions rely heavily on the completeness and representativeness of the data (i.e., wireless signals) patterns contained in the training set. Thus, they usually fail when the signals with new patterns are fed into the model (e.g., when the human subject is conducting an activity which is unseen during the model training).

To solve the problem of unseen signal patterns, we have to augment the training data with the signals representing unseen activities. Towards this end, the pioneering work XModal-ID [18] sheds light on the task of wireless signal augmentation by physically simulating the reflection of RF signals on the surface of human body. However, the simulation method in the XModal-ID model is designed base on the WiFi signals, which are mainly cosine signals with constant frequency, and thus cannot be directly applied to other popular wireless sensing signals (e.g., FMCW signals). Furthermore, XModal-ID focus on person identification, which

is a classification task. Thus, this approach is not suitable for more sophisticated wireless human sensing tasks (e.g., human pose/mesh estimation).

To address the problems, we propose a novel wireless signal augmentation framework, named **mmGPE**, for **mm**Wave-based **G**eneralized **P**ose **E**stimation. In mmGPE, we adopt the basic idea of the physical simulator in [18] and extend it to fit the properties of mmWave FMCW signals. However, this physical simulator is far from perfect due to many factors such as the noise from the ambient reflections, imperfect modeling of the human subject, the multi-path effect, and the error-prone nature of the RF signals. Due to the imperfect simulation of the physical world, there is a big gap between the signals generated by the physical simulator and the real-world signals collected by the mmWave radar. For this reason, the physical simulator, though still being able to perform some simple tasks such as person identification, would tend to fail for more challenging tasks, such as human mesh estimation, which require much more precise perception of the human activities.

To tackle the above challenge, we propose to integrate the physical signal simulation with deep learning techniques. Specifically, we develop a deep learning-based signal refiner in mmGPE that is capable of filling the gap and producing realistic signal data. The refiner is composed of a generator and a discriminator. The goal of the refiner is to train a generator that is capable of generating realistic signal data for the unseen activities. This is achieved with the help of the discriminator who evaluates the quality of the generated signals and plays a minimax game with the generator. In addition, in our model design, we decompose the contribution of different parts of the human body on the synthesized signals, which enables our model to learn the prior knowledge from the seen activities that share similar posture on some parts of the body with the unseen activities. By combining the power of both the physical model and the deep learning model, our proposed framework is able to produce realistic signals representing the unseen poses.

In order to evaluate the performance of the proposed mmGPE framework, we build a real-world testbed for our system using COTS mmWave devices. The evaluation results demonstrate that our mmGPE system can accurately generate human meshes for unseen activities.

In summary, our contributions in this paper are:

- We study the problem of the **unseen human mesh estimation** using mmWave signals and propose **the first framework** that is able to infer the correct pose and shape of the subject when the mmWave signals are unseen in the training data.
- Our proposed novel signal augmentation solution **integrates both physical and deep learning models** and thus can produce realistic signals precisely capturing the unseen poses.
- We build a real-world testbed using **COTS mmWave devices** and extensively evaluate our system. The superior experimental results demonstrate the effectiveness of our proposed mmGPE framework.

## 2 SYSTEM OVERVIEW

The goal of our proposed mmGPE system is to reconstruct 3D human mesh from the collected mmWave signals when a subject is conducting activities in front of a mmWave radar. The activities here can be either **seen** or **unseen** by the mmGPE system before the reconstruction. Please note that in this paper the seen activities refer to *the activities for which we have collected corresponding mmWave signals and used them for training*, and the unseen activities are *those whose corresponding mmWave signals are NOT included in the training data*. We achieve the above goal by synthesizing mmWave signals from the augmented meshes of the unseen activities and training a mesh estimator using both the real-world data of seen activities and the synthesized data of unseen activities. During the evaluation, the trained mesh estimator is able to correctly infer human meshes when the real-world mmWave signals of unseen activities are fed into the model. Figure 1 illustrates both the training process and the evaluation process of the proposed mmGPE system.

### 2.1 Training Process

In the training process, we not only collect the data for the seen activities, but also propose a signal augmenter and utilize it to generate realistic mmWave signals for the unseen activities. Then, the mesh estimator is trained on both the real-world data of the seen activities and the synthesized data of the unseen activities.

**Data Collection for Seen Activities.** In this step, we collect the mmWave signals and the corresponding ground truth 3D human mesh for the seen activities. Firstly, the mmWave radar hardware mix the received signals with the transmitted signals to obtain the IF (Intermediate Frequency) signals. Then, the IF signals are fed into the Heatmap Calculator to generate two heatmaps. Among them, one heatmap illustrates the location of the reflectors in the horizontal 2D space, where a pixel with a higher energy value in the heatmap represents a stronger reflection in the corresponding 2D place. The other heatmap shows the velocity distribution of the reflectors in different ranges (more details about the heatmaps and their calculations can be found in Section 3.3). The heatmaps are then fed into the Mesh Estimator (Section 3.5) for mesh estimation. In the meanwhile, we use a VICON [39] motion capture system to capture high precision dynamic pose information of the subject, which can be utilized to generate the ground truth human mesh.

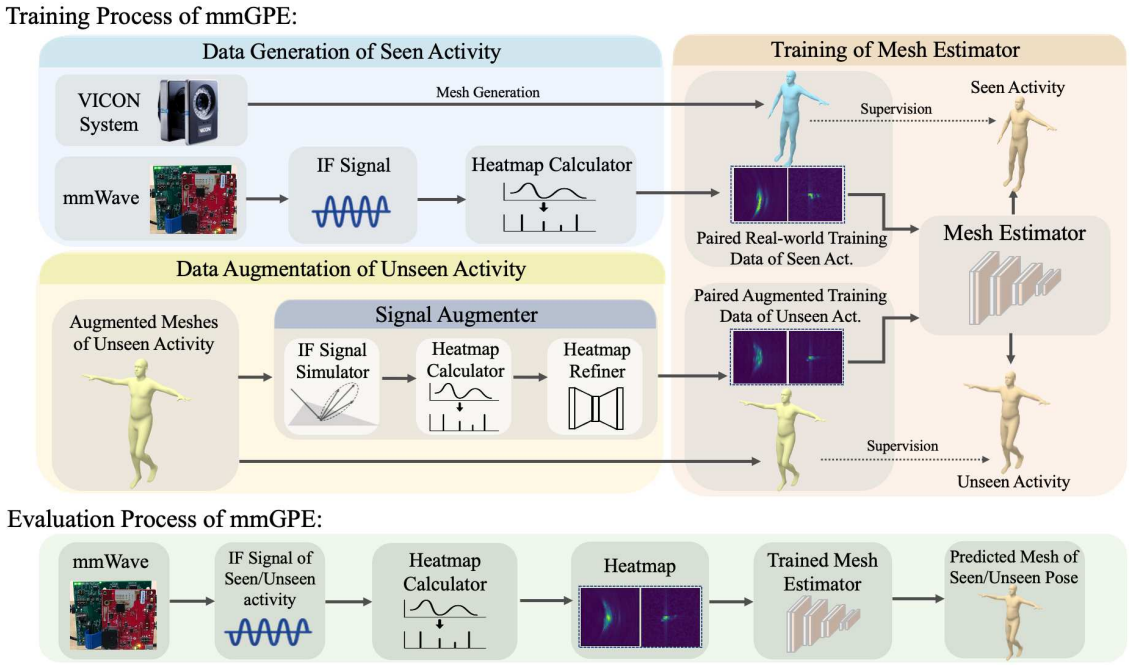



**Figure 1: System Overview of mmGPE Framework**

**Data Augmentation for Unseen Activities.** Our goal in this step is to synthesize realistic mmWave signals based on the augmented meshes of the unseen activities. The augmented meshes of the unseen activities can be easily obtained by applying existing mesh estimation algorithms to the videos that capture such activities. In real practice, we can also generate the meshes of interested activities (though their mmWave signals are unseen) using some motion capture systems (e.g., VICON) or by mesh editing. In this paper, we propose the **mmGPE Signal Augmenter** that can synthesize mmWave signals from 3D meshes. Generally, it is challenging to directly transform 3D meshes into realistic wireless signals, since their data modalities are different.

To tackle the above problem, our proposed signal augmenter combines the physical simulation and the power of deep learning to synthesize realistic signals of given 3D meshes, and eventually outputs heatmaps needed for mesh estimation. The proposed augmenter is composed of three modules: **the IF Signal Simulator**, **the Heatmap Calculator**, and **the Heatmap Refiner**. The IF Signal Simulator (Section 3.2) simulates the received IF signals by physically synthesizing the reflections of the RF signals on the human body. Then, the synthesized IF signals are fed into the Heatmap Calculator (Section 3.3) to generate coarse heatmaps. However, there is usually a gap between the generated heatmaps and the real heatmaps due to the imperfect simulation of physical world. To tackle this problem, we propose the Heatmap Refiner (Section 3.4) that utilizes deep learning techniques to fill the gap and produce high-quality heatmaps used to train the Mesh Estimator (Section 3.5) for unseen activities.

**Training of Mesh Estimator.** In this step, we train the Mesh Estimator module using both the collected mmWave signals for seen activities and the augmented data for unseen activities, as shown in Figure 1. The goal of the Mesh Estimator is to reconstruct human meshes from the mmWave signals of the activities, no matter whether they have been seen or unseen in the training set. For seen activities, the training data are the heatmaps generated from the IF signals collected by the mmWave radar, and their training process are supervised by the synchronized ground-truth meshes generated using the VICON system. For unseen activities, the training data are the heatmaps synthesized by our proposed Signal Augmenter from the augmented meshes of unseen activities, and their training process is supervised by the corresponding augmented meshes used to synthesize the heatmaps. Since the proposed Signal Augmenter can produce realistic mmWave signals of the unseen activities, the Mesh Estimator is able to *learn the mapping from the realistic signals of the unseen activities to their augmented meshes*. In this way, the trained Mesh Estimator is able to infer correct human meshes from the mmWave signals whose activities are unseen in the training set.

## 2.2 Evaluation Process

After the Mesh Estimator is fully trained, the seen and unseen activities are estimated in the same way during the evaluation process, as shown in Figure 1. In particular, the IF signals of the seen/unseen activities are first collected

by the mmWave radar. Then, the heatmaps are generated using the Heatmap Calculator. By feeding the heatmaps into the trained Mesh Estimator, the meshes of the seen/unseen activities can be generated. It is worth noting that the Signal Augmenter is no longer needed during the evaluation process. Thus, the trained Mesh Estimator can be potentially implemented in a real-time manner.

# 3 METHODOLOGY

## 3.1 Overview

The key problem in this paper is how to enable the mesh estimator to correctly estimate the 3D human meshes of the activities whose mmWave signals are unseen in the training set. Our proposed solution is to build a Signal Augmenter that can generate realistic signals and their heatmaps by taking the human meshes of the unseen activities as input and using the generated heatmaps to train the mesh estimator.

However, building such an augmenter is a challenging task since the data modalities of the 3D human mesh and mmWave signals are quite different. To address this challenge, we first build an IF signal simulator to physically simulate the received IF signals given the human mesh conducting the unseen activities. Though our developed IF signal simulator can generate mmWave signals, there is still a gap between the distribution of the simulated signals and that of the real-world signals. This is mainly because of the imperfection of the simulator, which can be caused by many factors such as the noise from the ambient reflection, imperfect modeling of the human subject, the multi-path effect, the error-prone nature of the RF signal, and the limited synchronization accuracy of the signal data and the mesh data.

To tackle the above problem, we leverage deep learning techniques to fill the distribution gap and refine the simulated signals.We first leverage the Heatmap Calculator to generate the heatmaps from the raw IF signal. Then, we propose a Heatmap Refiner to fill the gap between the coarse heatmap derived from the simulated signals and that from the real-world mmWave signals. The Heatmap Refiner is composed of a forward module and a backward module. The forward module includes a generator to generate the refined heatmap and a discriminator to ensure the quality of the generated refined heatmap. The goal of the forward module is to ensure the Heatmap Refiner correctly maps the coarse heatmap into the distribution of the real heatmap. The backward module also contains a generator and a discriminator, it maps the refined heatmap back to its corresponding simulated heatmap and forms a CycleGan structure [53]. In this way, the possible mode collapse problem can be mitigated during the training of the Heatmap Refiner [36]. In general, by combining the physical model and the deep learning model, our proposed Signal Augmenter is capable of producing realistic heatmaps of the unseen activities.

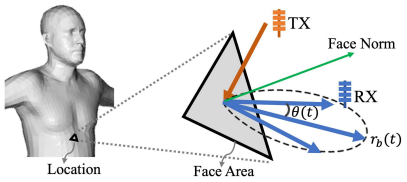



**Figure 2: Signal Simulation on Human Mesh**

## 3.2 mmWave IF Signal Simulator

The first step in our proposed Signal Augmenter is to physically simulate the IF signals. As illustrated in Figure 2, the simulation is conducted by synthesizing the signals traveling from the transmitting antennas to the receiving antennas which are reflected by all the triangle faces (the basic units composing the human mesh) on the human mesh. The simulation results are jointly determined by both the properties (i.e., locations, directions, and areas) of the triangle faces on the human mesh and the properties of the FMCW signals.

To build the IF signal simulator, we should characterize both phasor and strength of the simulated signals on each time point for every triangle face given the locations of both transmitters and receivers. For simplicity, in this section we only describe the signal simulation for one FMCW chirp between one pair of transceivers. The whole simulated signals can be obtained by repeating the calculation of all chirps on all transceiver pairs. Additionally, the Hidden Point Removal (HPR) algorithm [15] is applied to remove all the invisible triangle faces from the perspective of transceivers for computational efficiency.

Specifically, we denote the beginning time of the chirp as $t$. The transmitter and receiver locations are denoted as $l_T \in \mathbb{R}^3$ and $l_R \in \mathbb{R}^3$, respectively. We use $L_B(t) = \{l_b(t) \in \mathbb{R}^3, b = 1, ..., M\}$ to denote the set of the center locations of all the visible triangle faces at time $t$, where $M$ is the number of triangle pieces. Similarly, we use $N_B(t) = \{n_b(t) \in \mathbb{R}^3, b = 1, ..., M\}$ and $A_B(t) = \{a_b(t) \in \mathbb{R}^3, b = 1, ..., M\}$ to denote the sets of surface norms and areas of all the visible mesh triangles at time $t$, respectively.

First, we calculate the phasor of the IF signal. At time $t$, the phasor of the transmitted signal can be denoted as:

$$g(t) = \exp(j2\pi t(f_0 + \frac{S}{2}t)), \tag{1}$$

where $j$ is the imaginary unit, $f_0$ is the chirp starting frequency, and $S$ is the chirp slop of the FMCW signal. Similarly, the phasor of the received signal from the triangle face $l_b$ at time $t$ can be denoted as:

$$g(t, \tau_b(t)) = \exp(j2\pi(t - \tau_b(t))(f_0 + \frac{S}{2}(t - \tau_b(t)))), \tag{2}$$

where $\tau_b(t) = (||l_T - l_b(t)|| + ||l_b(t) - l_R||)/c$, $||\cdot||$ represents the Euclidean distance, and $c$ is the speed of light. Thus, the obtained IF signal from mesh triangle face $l_b$ at time $t$ can be denoted as:

$$f(t, \tau_b(t)) \approx \exp(j2\pi(f_0\tau_b(t) + St\tau_b(t))). \quad (3)$$

Note that the term $\frac{S}{2}\tau_b(t)^2$ is omitted since its order of magnitude is much smaller compared with $f_0\tau_b(t)$ and $St\tau_b(t)$.

For the calculation of the signal amplitude, we jointly consider three factors: the angle of the mesh triangles to the transceivers, the distance of the mesh triangles to the transceivers, and the size of the mesh triangles. According to [18], the human body is best modeled as a quasi-specular reflector, which reflects the signal into many directions with different amplitudes. The impact of the triangle angle $l_b$ at time $t$ can be represented as $c_b(t) = \exp(-\frac{\theta^2(t)}{2\sigma^2})$, where $\sigma$ is an empirical number and $\theta(t) = \arccos\frac{(l_R(t)-l_b(t))^T r_b(t)}{||l_R(t)-l_b(t)||}$ is the angle between the strongest reflection direction $r_b(t) = \frac{l_b(t)-l_T}{||l_b(t)-l_T||} - 2\frac{(l_b(t)-l_T)^T n_b(t)}{||l_b(t)-l_T||}n_b(t)$ and the received signal direction $l_R(t) - l_b(t)$. Overall, the simulated IF signal at time $t$ can be denoted as:

$$s(t) = \sum_{b=1}^{M} \frac{a_b(t)c_b(t)\exp(j2\pi(f_0\tau_b(t) + St\tau_b(t)))}{||l_T - l_b(t)|| * ||l_b(t) - l_R||} \quad (4)$$

By repeating the calculation on every transceiver pair and at all the time points, we can generate the simulated signals. Note that by varying the shape parameter and the displacement of the mesh, we are able to simulate the signals from the subject with any shape and anywhere to perform the activities within the sensing area of the mmWave radar.

## 3.3 Heatmap Calculator

After obtaining the IF signals, mmGPE will generate two heatmaps. Among them, one is the **location-MVDR heatmap** which illustrates the location of the reflectors in the horizontal 2D space. In this heatmap, a pixel with higher energy value indicates a stronger reflection in the corresponding 2D location of this pixel. The other one is the **range-velocity heatmap** which shows the velocity distribution of the reflectors in different ranges. In this heatmap, different rows represent different velocity levels and different columns represent different range distances. And a pixel with a high energy value in this heatmap indicates that there is a strong reflection with a certain velocity in a certain range corresponding to the pixel. In summary, the range, location, and velocity information of the subject can all be characterized by these two heatmaps, so they contain enough information to produce the human meshes.

Specifically, to calculate these two heatmaps from the IF signals, we first perform the Range-FFT and remove some noise by simply subtracting the average value of the data on each range bin. The processed data can be denoted as a matrix $R \in \mathbb{R}^{P\times C\times D}$, where $P$ is the number of transceiver pairs, $C$ is the number of chirps in one frame, and $D$ is the size of the range bins. The range-velocity heatmap can be obtained by performing the Doppler-FFT on the chirp dimension of the matrix $R$ and averaging the data from all the transceiver pairs (e.g., averaging along the transceiver dimension).

Then, the beamforming algorithm MVDR [4] (Minimum Variance Distortionless Response) is utilized to generate the location-MVDR heatmap. In our design, we use the location-MVDR heatmap to denote the signal power of the specific locations on the level surface of the 1D radar array. For a specific location $(x, y)$ (we assume the 1D radar array is arranged on the $x$ axis), the distance from the location to the receiver can be represented as $d = \sqrt{x^2 + y^2}$ and its range bin index is $\tilde{d} = \lfloor\frac{d}{\Delta d}\rfloor$ where we use $\Delta d$ to denote the span of a range bin. Meanwhile, its corresponding signal data is the $\tilde{d}$-th slicing of the matrix $R$ (i.e., $R_{\tilde{d}}$). Thus, the signal power on the location $(x, y)$ can be denoted as:

$$P(x, y) = \frac{1}{S_{(x,y)}^H \cdot (R_{\tilde{d}} R_{\tilde{d}}^H)^{-1} \cdot S_{(x,y)}},$$

where $H$ denotes the conjugate transpose of the data, and $S_{(x,y)}$ denotes the steering vector from the angle of the location $(x, y)$. Specifically, in our scenario, we have:

$$S_{(x,y)} = [1, \exp(-j\pi\frac{x}{d}), ..., \exp(-j\pi P\frac{x}{d})]^T,$$

where $P$ is the number of transceiver pairs. By calculating the power on all the desired locations, we can obtain the location-MVDR heatmap.

## 3.4 Heatmap Refiner

As aforementioned, a gap exists between the distributions of the simulated signals and real-world signals. Thus, the heatmap derived from the simulated signals cannot perfectly reflect the real-world setting. To fill the gap, we propose the Heatmap Refiner to refine the derived heatmaps to make them more realistic.

As illustrated in Figure 3, the forward module in the Heatmap Refiner is composed of a generator and a discriminator. Generally, our goal is to train a generator that is capable of generating realistic heatmaps for the unseen activities. This is achieved with the help of the discriminator who evaluates the quality of the generated heatmaps and plays a minimax game with the generator [7]. The backward module is designed to mitigate the possible mode collapse during the training of the generator.

Additionally, in our model design, we incorporate the heatmap masks as additional constraints to enhance the training of the Heatmap Refiner, especially for unseen activities. While real heatmaps can effectively supervise the training of Heatmap Refiner in the case of seen activities, generating high-quality refined heatmaps for unseen activities is impeded by the absence of such supervision. To address this, we leverage the consistent availability of human meshes during model training to generate 2D heatmap masks directly

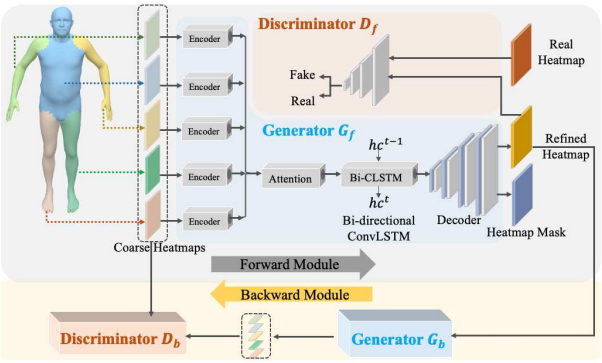



**Figure 3: Heatmap Refiner**

from these meshes for both seen and unseen activities. Specifically, the heatmap masks are images with 0/1 values that are directly estimated from the human mesh. For the heatmap mask corresponding to the Location-MVDR heatmap, we simply render human mesh in 2D space using the coordinates in the heatmap (i.e., a bird view from the top of the subject). We set the pixels covered by the human mesh to 1 and set other pixels to 0. For the heatmap mask corresponding to the Range-velocity heatmap, we estimate the speed of all the vertices on the 3D human mesh with respect to the radar and map them to the coordinates of the heatmap. In general, these masks encompass both location and velocity information from mesh vertices. It isolates the regions of each heatmap associated solely with human activities and sets all other areas to 0, filtering out the environmental noise. In this way, the heatmap masks can guide the generation of heatmap for both seen and unseen activities.

Specifically, the training schemes of the generator are illustrated in Figure 4, we use the data from both the seen and unseen activities to train the generator in the heatmap refiner. For the training upon the seen activities, we use an adversarial loss from the discriminator, a structural loss that is derived by comparing the predicted heatmap with the ground truth heatmap, a cycle-consistent loss [53] that calculates the consistency of a coarse heatmap sequentially mapped by the forward generator and the backward generator, and a mask loss that is estimated by comparing the predicted heatmap mask and the ground truth mask. For the training upon the unseen activities, we only use the adversarial loss, the cycle-consistent loss, and the mask loss to train the generator, since there is no ground truth heatmap of the unseen activities.

It is also worthwhile to mention that, we segment the human mesh into five parts: right arm, left arm, right leg, left left, and torso with head, as shown in Figure 3. During the simulation, we simulate the signals bouncing off different parts of the human body and generate the corresponding heatmaps. In this way, we can decompose the contribution of different parts of the human body on the generated heatmaps.

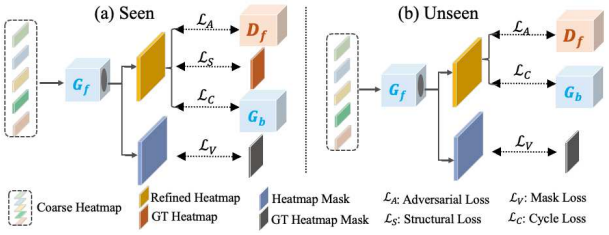



**Figure 4: Training Scheme of Heatmap Refiner**

The reason for this decomposition is as follows. In practice, the interested unseen activity may share similar postures on some parts of the body with some seen activities, and this design can help the refiner to learn the prior knowledge from these seen activities to improve the mesh estimation of the unseen activity on the shared body parts.

*3.4.1 Model Structure.* The forward module in Heatmap Refiner contains a generator $G_f$ and a discriminator $D_f$. The generator aims to learn the contribution of different body parts on the heatmaps when the subject conducts different activities. Here we use CNN-based Encoders to extract high-level features from the heatmaps of different parts of the human body and fuse the obtained representations using the attention mechanism. To reduce the influence of possible noise and the specularity of the RF signal, we feed the fused representation into a Bi-directional ConvLSTM (i.e., BiConvLSTM [10, 35, 49]) module to leverage the information from the adjacent frames to improve the generation of the current frame. Lastly, a Decoder composed of several deconvolutional layers is utilized to output both the refined heatmap and the predicted heatmap mask.

To learn the distribution of the real heatmap, we utilize a conditional heatmap discriminator $D_f$, which aims to identify whether the input heatmap is a real heatmap or a refined heatmap from the generator $G_f$ conditioned on the input of $G_f$. Here the generator $G_f$ and the discriminator $D_f$ play a minimax game. The discriminator tries to distinguish the heatmap generated by the generator from the real heatmap while the generator tries to fool the discriminator. Finally, the generator can successfully fool the discriminator, and generate a realistic heatmap. In our design, the discriminator is composed of several layers of convolutional neural network with spectral normalization [28, 47] and LeakyReLU [25] non-linear mapping. The real heatmaps or the simulated heatmaps are stacked into one image as the input of the discriminator. For the output, we leverage the technique in PatchGAN [12], where the discriminator tries to classify each patch in an image into two classes: real and fake.

The backward module in the Heatmap Refiner is also composed of a generator and a discriminator. The structure of the generator $G_b$ and the discriminator $D_b$ have similar structures to $G_f$ and $D_f$. By mapping back from the refined heatmap, the cycle-consistent loss can be calculated.

*3.4.2 Model Losses and Model Training.* To train the generator $G_f$ and the discriminator $D_f$ in the forward module of Heatmap Refiner, we utilize both the augmented dataset $U$ of the unseen activities and the real dataset $S$ from the seen activities. Specifically, the augmented dataset $U$ is composed of the augmented human mesh $m_a^u$, its corresponding coarse heatmap $x_a^u$ by applying the proposed IF Signal Simulator and Heatmap Generator to the augmented human mesh $m_a^u$, and the heatmap masks $v_a^u$ of the augmented mesh. The real dataset $S$ is composed of the real human mesh $m^s$ collected when the subject performs the seen activities, the corresponding heatmap $x^s$ is generated based on the IF signals from the mmWave radar, and the heatmap masks $v^s$ of the real human mesh. It is worth noting that the dataset $S$ also contains the coarse heatmap $x_a^s$ generated based on the IF Signal Simulator and Heatmap Calculator on real human mesh $m^s$. Note that we omit the time notation $t$ and part notation $p$ for simplicity.

For the training of the discriminator $D_f$ in the Heatmap Refiner, we use both the real heatmaps and the refined heatmaps as follows:

$$L_{D_f} = H(D_f(x^s|x_a^s), 1) + H(D_f(G_f(x_a^s)|x_a^s), 0),$$

where $D_f(x|y)$ denotes the output of network $D_f$ given $x$ as input and $y$ as the condition. $H$ denotes the Hinge loss.

The generator $Gs$ in the Heatmap Refiner is trained using both the structural loss, the discriminator loss, the cycle-consistent loss, and the mask loss:

$$\begin{aligned} L_{G_f} = & -M(G_f(x_a^s), x^s) - \alpha_a H(D_f(G_f(x_a^s)), 1) \\ & - \alpha_a H(D_f(G_f(x_a^u)), 1) - \alpha_b M(G_b(G_f(x_a^s)), x_a^s) \\ & - \alpha_b M(G_b(G_f(x_a^u)), x_a^u) - \alpha_c D(G_f(x_a^s), v^s) \\ & - \alpha_c D(G_f(x_a^u), v_a^u), \end{aligned} \tag{5}$$

where $\alpha_a$ and $\alpha_b$ are hyperparameters. $M$ denotes the MS-SSIM (Multi-scale Structural Similarity Index Measure) loss [43], which is calculated by applying the SSIM [42] loss on multiple scales of the input heatmaps and ranges from 0 (dissimilar) to 1 (the same). $D$ denotes the dice coefficient [27], which measures the overlap of predicted masks and the ground truth makes. The losses for generator $G_b$ and discriminator $D_b$ in the backward module of Heatmap Refiner are similar to the losses of $G_f$ and $D_f$, except that there is no mask loss in the loss function of $G_b$.

Note that, in our design, there are two kinds of heatmaps (a range-velocity heatmap and a location-MVDR heatmap). Thus, we have two heatmap refiners to generate the two types of heatmaps, respectively.

## 3.5 Mesh Estimator

In our proposed mmGPE system, we also design the Mesh Estimator $G_m$ to infer the human mesh from the heatmaps. As illustrated in Figure 5, the Mesh Estimator takes both the location-MVDR heatmap and the range-velocity heatmap as input and outputs the 3D human mesh. Note that the input heatmaps of the Mesh Estimator could be either from the Signal Augmenter (for unseen activities) or from the mmWave radar signals (for seen activities).

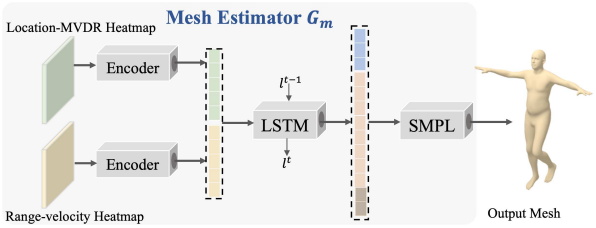



**Figure 5: Mesh Estimator**

*3.5.1 Model Structure.* First, we obtained the feature vectors from both Location-MVDR Heatmap and Range-velocity Heatmap by feeding them into the CNN-based Encoder. Next, we concatenate the obtained two feature vectors and feed them into a three-layer LSTM module to learn the activity information from both the current and previous frames. Then, we use a fully connected neural network to obtain the pose vector, shape vector, and translation vector of the human mesh. Finally, we follow the mesh generating scheme in [46] and feed the vectors into the neutral SMPL [24] (Skinned Multi-Person Linear) model (no gender information in the neutral SMPL model) to output the skeleton and the mesh of the subject. The SMPL model is a widely used parametric human model that estimates 3D human mesh by factoring the human body into a shape parameter $b^t \in \mathbb{R}^{10}$ and a pose parameter $p^t \in \mathbb{R}^{72}$. The shape parameter can be utilized to control how individuals vary in height, weight, and body proportions. The pose parameter is used to control the pose of the 3D human mesh.

*3.5.2 Model Loss.* To train the Mesh Estimator, we use both the real heatmap data and the refined heatmap data:

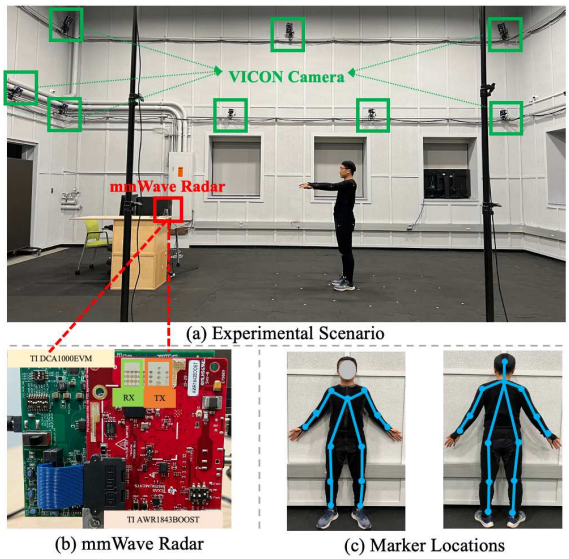
$$\begin{aligned} L_{G_m} = & L1(G_m(x^s), m^s) + \alpha_d L1(G_m(G_f(x_a^s)), m^s) \\ & + \alpha_e L1(G_m(G_f(x_a^u)), m_a^u), \end{aligned} \tag{6}$$

where $\alpha_d$ and $\alpha_e$ are the hyperparameters.

# 4 EXPERIMENTS

## 4.1 Testbeds

*4.1.1 VICON System.* The VICON [39] system is an optical camera-based system for high-precision human motion capture and we use it to generate the ground truth 3D human pose. As shown in Figure 6 (a), the VICON system consists of 21 VICON cameras that can emit and receive infrared light. The emitted infrared light can be reflected by pearl markers and captured by the VICON cameras. We attach 27 such makers to the joints of the human body so that the VICON system can generate a precise human skeleton based on the location of all the markers. Figure 6 (c) shows the positions




(a) Experimental Scenario

(b) mmWave Radar (c) Marker Locations

**Figure 6: Experiment Setting**

of these markers on the participant. The sampling rate of the system is set to 10 frames per second.

*4.1.2 mmWave Testbed.* As shown in Figure 6 (b), we use TI AWR1843BOOST mmWave radar [38] to collect mmWave data and the TI DCA1000 evaluation module to stream the data from the mmWave radar. The mmWave radar consists of 3 transmitting antennas and 4 receiving antennas, emitting and receiving FMCW signals. The RF signal in each FMCW chirp has a bandwidth of 3.9GHZ which increases from 77GHZ to 80.9GHZ linearly. The mmWave radar is also set to send 10 frames per second to align with the VICON system. Here each frame is composed of 128 chirps, and each chirp is composed of 256 sampling points. Based on our device setting, our mmWave device can reach up to 11m sensing range, 4.3cm range resolution, 4.5m/s sensing velocity, and 7.1 cm/s velocity resolution. In the experiment, we place the mmWave testbed on a table (the height is 92 cm), and the distance between the mmWave radar and the subject ranges from 1.5m to 4m.

## 4.2 Data Collection

*4.2.1 Activity Design.* In our experiment, 4 subjects (including both male and female subjects) are recruited to perform dozens of activities which are split into two datasets: an immobile activity dataset and a mobile activity dataset.

In the immobile dataset, the subjects are performing activities in a specific spot. As illustrated in Figure 7, the elementary activities are: (a/b) waving right/left arm horizontally , (c/d) raising right/left arm to the front vertically, (e/f) raising right/left arm to the side vertically, (g/h) lifting right/left knee, (i/j) lifting right/left leg to front, (k/l) lifting right/left leg to side, (m/n) a step leftward/forward and back. The compositive activities are the combination of the elementary activities, including the activities using both arms (i.e., the

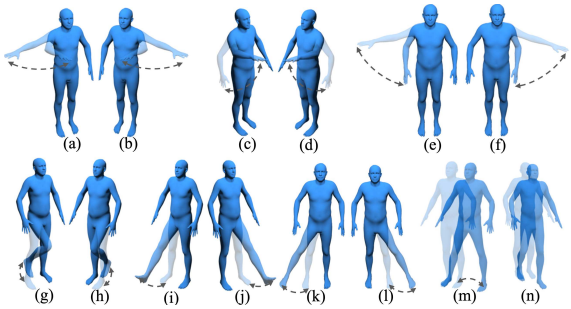



**Figure 7: Elementary Activities**

combination of two activities from (a)-(f)), the activities using one arm and one leg (i.e., the combination of one activity from (a)-(f) and one activity from (g)-(n)), and the activities using two arms and one leg (i.e., the combination of two activities from (a)-(f) and one activity from (g)-(n)). In the mobile dataset, the subjects are required to walk back and forth in a line (about 2.3m) in front of the radar while performing both elementary activities from (a) to (f) in Figure 7 and compositive activities. The compositive activities are the combination of the walking activity and two elementary upper body activities (i.e., the combination of walking and two activities from (a)-(f)).

In our experiments, for both immobile and mobile datasets, all the elementary activities are used as the seen activities, and all the compositive activities are unseen activities.

*4.2.2 Data Collection.* In our experiments, we perform the same data collection process for both the immobile dataset and the mobile dataset. For each dataset, there are three steps in the data collection. The first step is to collect the data for seen activities. We ask each subject to repeatedly perform each seen activity in front of the mmWave radar for 135 seconds, and we simultaneously collect the IF signal data from the mmWave radar and the corresponding pose data from the VICON system. The second step is to collect the augmented meshes of unseen activities. As we discussed previously, there are different ways/opportunities to collect the augmented meshes, e.g., image/video-based methods [14, 16, 22], motion-capture-based solutions [29, 39], existing 3D skeleton/mesh dataset [5, 26] and mesh editing. In this experiment, we use the VICON system to generate the augmented meshes. Specifically, the subjects are asked to perform the unseen activities for 135 seconds, and we only collect the pose data from the VICON system. Note that *the collected augmented meshes are used in the training process as the input of Signal Augmenter, therefore no IF data is collected in this process.* The third step is to collect the data for the model evaluation. In this step, we ask the subjects to perform the unseen activities for 15 seconds. Different from step 2, in this step we collect both the pose data from the VICON system and the IF signal data from the mmWave radar. Note that the real-world IF signals of the unseen activities are

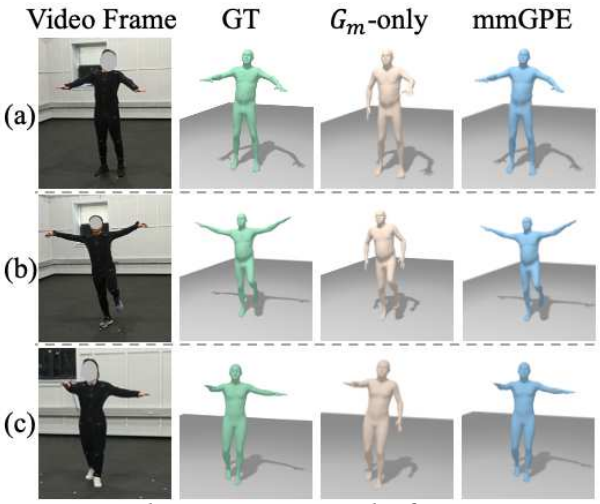



**Figure 8: Mesh Estimation Results for Unseen Activities**

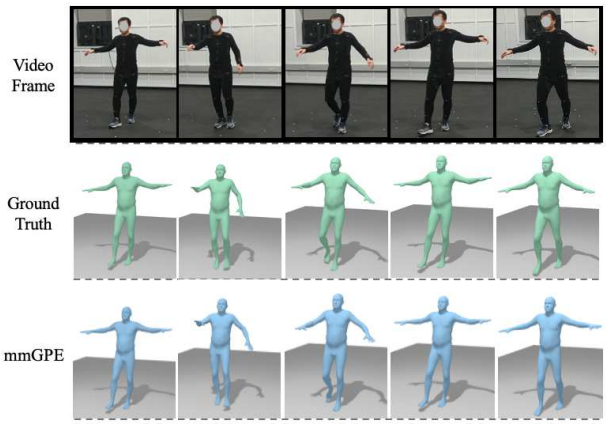



**Figure 9: Mesh Estimation Results for Consecutive Frames of an Unseen Activity**

only used for model evaluation. We use the same methods as presented in [46] to generate the SMPL-based ground truth meshes.

## 4.3 Model Setting and Model Training

In this section, we describe the implementation details of our mmGPE framework. The signal simulator utilizes an NVIDIA A6000 GPU, enabling each frame of data to be simulated in 1.5 seconds. For the settings of the deep model, the Encoder in both Generators $G_f$ and $G_b$ has 5 convolutional layers. In our design, 2 layers of BiConvLSTM [10, 35, 49] are used before the Decoder, which has 2 deconvolutional layers and 4 convolutional layers. The Discriminators $D_f$ and $D_b$ are composed of 4 convolutional layers. For the Mesh Estimator, the Encoder also contains 7 convolutional layers. The number of LSTM layers is set to 3. The total number of trainable parameters in our model is 2.3 million. For our model design, each convolutional layer or deconvolutional layer is followed by a Batch Norm [11] layer and a LeakyReLU [25] non-linear mapping, except the Discriminator $D_f$ where the Batch Norm layers are replaced with the Spectral Norm layers [28, 47]. In Equation 5, the hyperparameter $\alpha_a$, $\alpha_b$ and $\alpha_b$ are set to 0.01, 0.01, and 0.1, respectively. In Equation 6, the hyperparameter $\alpha_d$ and $\alpha_e$ are set to 1. Finally, we use PyTorch [30] to implement all of our deep learning models and NVIDIA A6000 to train the model.

## 4.4 Evaluation Metrics

We use the following metrics to evaluate the generated meshes for the unseen activities:

**Average Vertex Error (V)** [3, 51]. We compute the average vertex error by averaging the Euclidean distance between the vertices located on the predicted human mesh and the corresponding vertices on the ground truth mesh for all the subjects and the unseen activities. This metric can evaluate the overall performance of the location error, pose error, and shape error.

**Average Joint Localization Error (S)** [13, 51]. This metric is defined as the average Euclidean distance between the joint locations of the predicted human mesh and the ground truths for all the subjects and activities.

**Procrustes Alignment-Mean Per-Joint Position Error (PA-S)**. Since the predicted poses come with translation and orientation, we also evaluate the wellness of the aligned poses. This metric first aligns the estimated 3D pose to the ground truth by a rigid transformation called Procrustes [8] and then calculates the mean Euclidean distance between the aligned predicted and the ground truth skeletons for all the subjects and activities.

**Average Joint Rotation Error (Q)**. This metric is defined as the average differences between predicted joint rotations and the ground truth rotations, where only the rotations of shoulder joints, elbow joints, hip joints, and knee joints from both sides of the subject are considered.

**Mesh Localization Error (T)**. We also use mesh localization error to assess the precision of subject localization. This metric is defined as the average Euclidean distance between the root joint location of the predicted human mesh skeleton and the ground truths for all the subjects and activities.

## 4.5 Models for Evaluation

In the experiments, we evaluate the performance of the following models:

**$G_m$-only.** This model is a baseline where the Mesh Estimator is trained using only the data of seen activities.

**$G_m$+Simulator.** This model is also a baseline where the Mesh Estimator is trained using both the data of the seen activities and the coarse heatmap of the unseen poses (i.e., the Heatmap Refiner in the Signal Augmenter is removed).

**mmGPE-Whole.** This is the proposed model except that we conduct the simulation using the whole human body instead of using 5 decomposed parts of the human body respectively.

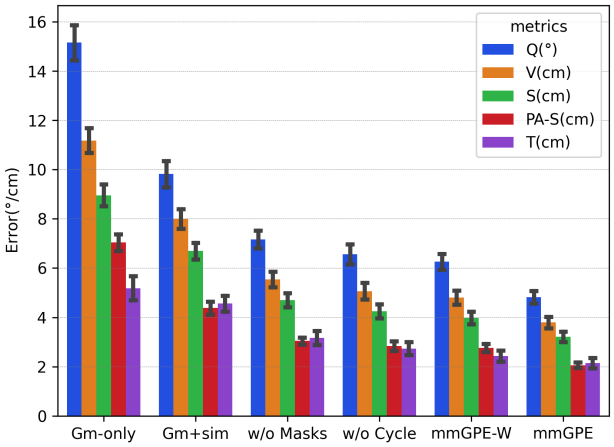



**Figure 10: Results for Immobile Activities**

Accordingly, the Generator in the Heatmap Refiner only has one Encoder and has no Attention module.

**mmGPE.** This is the complete implementation of our proposed model in the paper.

## 4.6 Qualitative Mesh Estimation Results

In this section, we qualitatively evaluate the baseline models and our proposed models. The mesh estimation results for the unseen activities are shown in Figure 8. Rows (a) and (b) show different subjects conducting unseen activities in the immobile dataset. These activities are the combinations of the elementary activities (i.e., activities a+b, e+f+h, respectively), including the activities using both arms, and using two arms and legs. Rows (c) shows one subject conducting the unseen activity of the mobile dataset. The subject is conducting the combinations of the elementary activities while walking (i.e. walk+b+c). In the figure, the video frame, ground truth mesh, the mesh constructed by the baseline $G_m$-only, and results from our proposed mmGPE model are shown in each row from the left to right.

From Figure 8, we can observe that the meshes (both poses and shapes) generated by our mmGPE model closely match the ground truth meshes, demonstrating our model's ability to accurately reconstruct the human meshes for unseen activities. This is achieved by training the Mesh Estimator using the realistic synthesized data generated by the Signal Augmenter. In contrast, the baseline model can only generate the poses from the seen activities, even if the subjects are performing unseen activities. It demonstrates that the Mesh Estimator cannot function well without the help of the proposed Signal Augmenter.

Figure 9 shows the consecutive frames (interval of 0.8$s$ for better visualization) where a subject is conducting a compositive activity (e.g., walk+a+f) in the mobile dataset. The video frame, ground truth mesh, and our results are shown in different rows. We can see that the generated meshes are smooth and accurate despite the complexity of the unseen activities. One reason behind this is that the BiConvLSTM can utilize information from the time dimension to synthesize smooth changes among the consecutive heatmap frames.

**Table 1: Results for Different Training Data Size**

| Length | Q(°) | V(cm) | S(cm) | PA-S(cm) | T(cm) |
|---|---|---|---|---|---|
| 135s | 4.81 | 3.79 | 3.21 | 2.06 | 2.14 |
| 120s | 5.43 | 4.47 | 3.89 | 2.27 | 2.83 |
| 105s | 6.08 | 5.11 | 4.46 | 2.53 | 3.33 |
| 90s | 6.45 | 5.30 | 4.53 | 2.72 | 3.18 |

## 4.7 Quantitative Mesh Estimation Results

In this section, we quantitatively evaluate the accuracy of the meshes produced by the proposed and baseline models. The results, displayed in Figure 10 with standard error, demonstrate the superior performance of our proposed mmGPE model compared to baseline methods across all metrics. From the Figure 10, we can see that the $G_m$+simulator model performs better than the $G_m$-only model, which demonstrates even though our physical simulator is imperfect, the generated coarse heatmap still characterizes certain basic patterns (e.g., velocity distribution, body-limb relative positions) of unseen poses and can improve the performance of the mesh estimator on the unseen pose. Then, to study the effectiveness of mask loss and CycleGAN structure, we conducted an ablation study by removing the mask loss (i.e., w/o Masks) and the backward module (i.e., w/o Cycle), respectively. It can be seen that the model performance drops without mask loss or CycleGAN structure, which demonstrates both designs can help the Heatmap Refiner to generate more accurate refined heatmaps to enable accurate human mesh estimation. This figure also shows that the mmGPE model outperforms the mmGPE-whole model. This is mainly because the mmGPE model decomposes the contribution of different human parts during the synthesis of the signal and

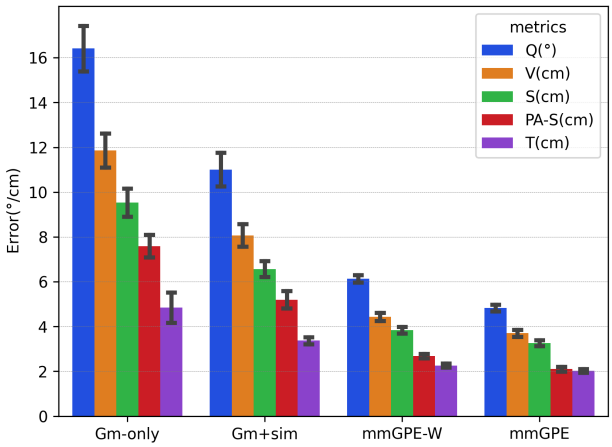



**Figure 11: Results for Mobile Activities**

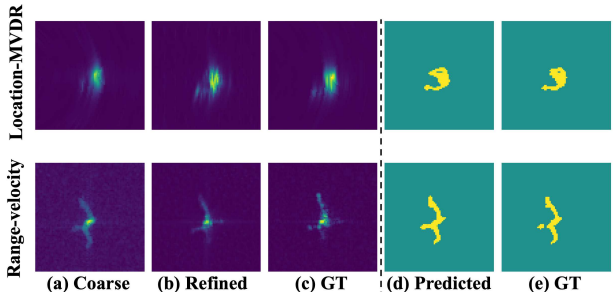



**Figure 12: Heatmaps in Immobile Dataset**

hence is capable of utilizing the prior knowledge from the seen activities when the seen/unseen activities share the same poses on some parts of the human body.

In Table 1, we examine the influence of varying training data size (proportional to the duration of recorded activities) on the performance of mmGPE model. The results show a reduction in performance as the duration of recorded activities in the training data decreases. Nonetheless, even with a reduction from 135s to 90s in activity duration, the model's performance remains relatively stable, demonstrating the robustness of our proposed approach.

Figure 11 shows the performance of the models on the mobile dataset. We can see that our mmGPE-based models still outperform the baselines with large margins on all the metrics. This demonstrates the effectiveness of our framework in estimating the unseen poses for moving subjects.

## 4.8 Quality of Refined Heatmaps

In this section, we evaluate the results of the refined heatmaps generated by the Heatmap Refiner. Figure 12 shows the location-MVDR and range-velocity heatmaps when the subject is walking while waving the left arm horizontally and raising the right arm vertically. As we can see, the first row shows the heatmaps or heatmap masks related to Location-MVDR, and the second row shows the heatmaps or heatmap masks related to Range-Velocity. From left to right columns, we show the coarse heatmaps by averaging the coarse heatmaps from five parts of the subject, the refined heatmaps by Heatmap Refiner, ground truth heatmaps from mmWave radar, predicted heatmap masks, and ground truth heatmap masks. It can be seen that the coarse heatmaps are similar to the ground truth heatmaps, which demonstrates the effectiveness of the physical simulator to characterize the patterns of unseen activities. Furthermore, the similarity can still be improved in the refined heatmap by our Heatmap Refiner. Especially, the body parts are more distinguishable in the refined Location-MVDR heatmap than those in coarse heatmap. This also indicates our designed refiner successfully reduces the gaps between the simulated coarse heatmaps and the real-world heatmaps. Additionally, the predicted heatmap masks are close to the ground truth heatmap masks, which demonstrates the generator can extract activity-specific information to correctly generate the heatmap masks.

**Table 2: Results of the Heatmap Refiner**

| Dataset | Type | Coarse | Refined |
|---|---|---|---|
| Immobile | location-MVDR | 0.874 | 0.945 |
| | range-velocity | 0.880 | 0.915 |
| Mobile | location-MVDR | 0.895 | 0.968 |
| | range-velocity | 0.910 | 0.961 |

We further report the MS-SSIM scores by comparing the generated heatmaps with the ground truth heatmaps on all the unseen activities in Table 2. MS-SSIM score indicates the multi-scale structural similarity between two heatmaps ranges from 0 to 1, where higher values indicate higher similarity. In the table, we can see that the simulated coarse heatmaps can achieve decent similarity with the real heatmaps and are further improved in the refined heatmaps. It demonstrates the effectiveness of both the physical simulator and the Heatmap Refiner for the task of generating realistic heatmaps of unseen activities.

## 4.9 Reconstructing Unseen Elementary Activities

In this section, we demonstrate our proposed framework can also be applied to unseen elementary activities. Note that the unseen elementary activities have no overlap with seen activities. Specifically, as shown in Table 3, we split the elementary arm activities (a)-(f) in Figure 7 as seen and unseen activities, and use seen elementary activities to reconstruct unseen ones. In each row from top to bottom, we set elementary activities (a) & (b), (c) & (d), (e) & (f), and (a) & (c) & (e) as unseen activities, respectively. Note that in the fourth row, the seen activities are performed by left arms, and the unseen activities are performed by the right arms. As shown in Table 3, our model is able to accurately reconstruct the unseen arm elementary poses. Similarly, in Table 4, we apply the same approach to elementary leg activities (g)-(l) in Figure 7, splitting them into seen and unseen categories. We can see our model can still accurately reconstruct the unseen elementary leg poses. These results indicate that our

**Table 3: Results for Elementary Arm Activities**

| Unseen Acts | Q(°) | V(cm) | S(cm) | PA-S(cm) | T(cm) |
|---|---|---|---|---|---|
| (a) & (b) | 3.54 | 3.07 | 2.42 | 1.70 | 1.44 |
| (c) & (d) | 4.13 | 3.13 | 2.41 | 2.03 | 1.28 |
| (e) & (f) | 4.08 | 3.55 | 2.86 | 1.94 | 1.84 |
| (a) & (c) & (e) | 6.67 | 5.31 | 4.16 | 3.37 | 2.37 |

**Table 4: Results for Elementary Leg Activities**

| Unseen Acts | Q(°) | V(cm) | S(cm) | PA-S(cm) | T(cm) |
|---|---|---|---|---|---|
| (g) & (h) | 4.32 | 3.61 | 3.59 | 1.94 | 2.65 |
| (i) & (j) | 4.93 | 3.47 | 3.67 | 2.70 | 2.22 |
| (k) & (l) | 4.37 | 3.52 | 3.50 | 1.84 | 2.69 |
| (g) & (i) & (k) | 5.67 | 5.79 | 5.68 | 2.31 | 5.10 |

model has the ability to estimate the pose of unseen activities that have no overlap with the seen activities.

## 4.10 Results of Different Orientations

In this section, we report the results when the unseen activities are performed in different orientations from seen activities. Firstly, as illustrated in Figure 13, the subjects are asked to perform elementary activities facing the mmWave radar as seen activities, which we refer to as the reference orientation (i.e., the orientation of 0°). Then, the subjects rotate their orientations for 23°, 45°, and 90° with respect to the reference orientation to perform both the elementary and compositive activities as unseen activities.

In Table 5, we evaluate our model's performance when the subject was facing different directions. We can see our model is able to correctly reconstruct both elementary and compositive activities, albeit with a decrease in performance as the angle of the subject's orientation increases. When the orientation difference between seen and unseen activities reaches 90°, their reflected signals' patterns differ significantly, and the proposed model would be likely to fail.

## 4.11 Results for Unseen Environments and Subjects

Firstly, we evaluate the performance of our framework when the subjects conduct mobile or immobile compositive (unseen) activities in two different furnished environments. Note that the data of seen elementary activities for model training are collected in the basic unfurnished environment as shown in Figure 6 (a). As illustrated in Figure 14, in scenario (a), some furniture and objects are placed around the subject. In scenario (b), in addition to the furniture, we also placed two strong reflectors (metal boards) behind the subject to change the reflecting pattern of the signals. As shown in Table 6, our system can accurately reconstruct unseen human poses for both immobile and mobile activities even in different environments, albeit with a minor decrease in performance compared to the basic environment. One possible reason is

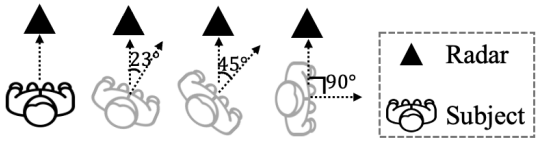



**Figure 13: Different Orientations**

**Table 5: Results for Different Orientations**

| Angles | Q(°) | V(cm) | S(cm) | PA-S(cm) | T(cm) |
|---|---|---|---|---|---|
| 23°-elementary | 6.26 | 6.62 | 6.18 | 2.74 | 5.09 |
| 23°-compositive | 7.85 | 8.17 | 7.52 | 3.62 | 6.12 |
| 45°-elementary | 8.23 | 7.63 | 6.96 | 3.84 | 4.21 |
| 45°-compositive | 8.13 | 10.23 | 9.50 | 3.74 | 7.13 |
| 90°-elementary | 12.1 | 21.4 | 20.3 | 5.12 | 19.3 |
| 90°-compositive | 14.7 | 17.9 | 16.4 | 7.25 | 14.0 |

that the incorporation of heatmap masks can help our model reduce the impact from the environment. This experiment demonstrates the robustness of our system to environmental changes without additional training effort.

In addition, we also evaluate the performance of our system on unknown subject in the basic environment. The setting of this experiment is the same as the experiments in Figure 10. The difference is that we select one subject as the unseen subject and remove the elementary (seen) activities data of that subject from training data. Then we train the model and report the results on compositive (unseen) activities of that subject. As shown in the last row of Table 6, our model can still achieve decent performance even if the real-world signals of the subject was unseen during the training stage. This demonstrates the strong generalization ability of our proposed model.

## 4.12 Mesh Augmentation by Mesh Editing

To demonstrate the effectiveness of our proposed framework using different mesh augmentation methods, we use the mesh editing method to generate the meshes of the unseen activities by combining the meshes only from the seen activities. Specifically, we use the mesh of the elementary activities in the immobile dataset from one subject and fuse the limbs of these activities to form the unseen activities of that subject. However, we still use the same evaluation data as that in Figure 10. Our proposed models still output good results (Q-6.83°, V-5.22$cm$, S-4.43$cm$, PA-S-3.06$cm$, T-2.43$cm$), which are comparable to the results in Figure 10. This demonstrates that our proposed mmGPE framework is general and extendable to different mesh augmentation methods in different real-world scenarios for the unseen mesh estimation task.

## 4.13 Limitations

In our study, the proposed framework is more effective when reconstructing unseen activities that have higher similarities with the seen activities, while struggling with activities that have less or no similarities with the seen activities.

To demonstrate it, we conduct experiments when the orientation of the subject changes. Specifically, the seen activities are conducted when the subject's orientation is 0 degree, and the unseen activities are conducted when the subject's orientations are 23, 45, and 90 degrees respectively.

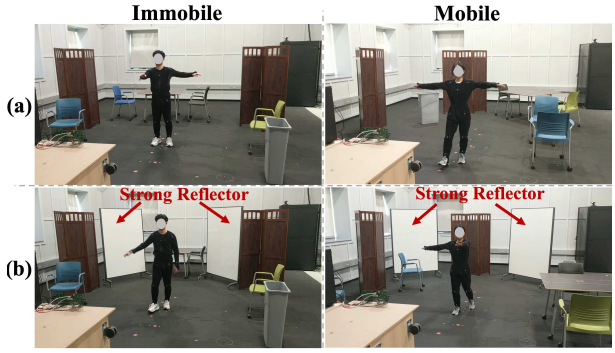



**Figure 14: Furnished Environments**

Intuitively, with the increase of the orientation angles, the similarity between the reflected signals of seen and unseen activities is dropping. As illustrated in Table 5, the estimation errors increase as the angle of the subject's orientation increases. When the orientation difference between seen and unseen activities reaches 90 degrees, their reflected signals' patterns differ significantly, and the proposed model would be likely to fail.

Furthermore, we also conducted experiments using elementary arm activities (i.e., (a)-(f)) as seen activities, and elementary leg movements (i.e., (g)-(l)) as unseen activities in Figure 7. We observe that our model is also struggling to infer unseen activities (results: Q-10.6°, V-13.5$cm$, S-12.9$cm$, PA-S-3.98$cm$, T-13.2$cm$) due to the significant differences in signal patterns caused by arm and leg movements. Despite this issue, our model remains capable of reconstructing unseen poses across similar limbs (Section 4.9), reasonable orientation changes of the subject (Section 4.10), environment changes (Section 4.11), and mesh editing methods (Section 4.12).

# 5 RELATED WORK

## 5.1 Human Pose/Mesh Estimation from Wireless Signals

In recent years, many wireless sensing systems have been developed to estimate human pose and mesh [1, 13, 31, 33, 34, 40, 41, 46, 50–52]. Among them, [40, 41, 50] focus on 2D pose estimation, and [13, 17, 33, 34, 52] are capable of estimating 3D human pose. Different from these works, our work estimates 3D human mesh, which contains both pose and shape information of the subjects. There are also two works that have explored 3D human mesh construction task using wireless signals. RF-Avatar [51] is a pioneering work that demonstrates the RF signals contain sufficient information for the human mesh estimation. mmMesh [46] and M$^4$esh [45] are another works which use the commercial mmWave radar to conduct human mesh estimation. However, none of the above works takes the unseen activities into consideration. In contrast, we focus on the task of 3D human mesh estimation for unseen activities in this paper.

**Table 6: Results for Unseen Environments & Subjects**

| Scenarios | Q(°) | S(cm) | PA-S(cm) | V(cm) | T(cm) |
|---|---|---|---|---|---|
| Env. (a) - Mobile | 7.26 | 5.19 | 4.41 | 3.11 | 2.54 |
| Env. (b) - Mobile | 7.85 | 5.94 | 5.06 | 3.41 | 3.06 |
| Env. (a) - Immobile | 6.93 | 5.24 | 4.43 | 3.27 | 2.52 |
| Env. (b) - Immobile | 7.74 | 5.99 | 5.41 | 3.45 | 3.68 |
| Unknown Subj. | 6.38 | 6.15 | 5.20 | 2.61 | 4.21 |

## 5.2 Data Augmentation for Human Sensing

Data augmentation methods has been attracting much attention for human sensing on different data modalities. Inertial Measurement Unit (IMU) data obtained from wearable devices has shown the capability of accurate human gesture sensing in [9, 23]. To simulate IMU data from existing large video libraries, generative algorithms of machine learning and forward kinematics are utilized to either directly map videos to IMU data or estimate joint directions in [19, 20, 32, 37]. The raw RF signals can also be augmented by physically analyzing the signal reflections on the human body and simulating the received wireless signals [18]. Furthermore, the Doppler signals, which are extracted from raw RF signals and used in many activity recognition tasks, can be augmented from the cloud points generated by depth-cameras in [6, 21], the human meshes generated from video data [2], or signal modeling [44]. Different from these works, our proposed framework focus on pose estimation, a regression task, instead of classification tasks. Recently, SynMotion [48] is proposed to conduct both activity recognition and pose estimation tasks by synthesizing mmWave sensing signals that bounce off the human body. However, few-shot samples are needed during the training process for the pose estimation task.

# 6 CONCLUSION

In this paper, we study the problem of the unseen human mesh estimation using mmWave signals. Specifically, we proposed a novel signal augmentation solution, named mmGPE, by synthesizing realistic mmWave signals from the augmented meshes of the unseen activities and training a mesh estimator using both the real-world data of seen activities and the synthesized data of unseen activities. In our design, the signal augmenter is capable of combining the power of both the physical simulator and the deep learning technique to ensure realistic simulation results, as well as learning the prior knowledge from the seen activities that share similar posture on some parts of the body with the unseen activities using a decomposing model structure. The superior mesh estimation results of unseen activities demonstrate the effectiveness of our proposed mmGPE framework.

# ACKNOWLEDGMENTS

This work was supported in part by the US National Science Foundation under Grant CNS-2154059.

# REFERENCES


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

[2] Karan Ahuja, Yue Jiang, Mayank Goel, and Chris Harrison. 2021. Vid2Doppler: synthesizing Doppler radar data from videos for training privacy-preserving activity recognition. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. 1–10.

[3] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. 2016. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In European Conference on Computer Vision. Springer, 561–578.

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

[5] CMU. 2000. CMU graphics lab motion capture database. http://mocap.cs.cmu.edu

[6] Baris Erol, Sevgi Z Gurbuz, and Moeness G Amin. 2019. GAN-based synthetic radar micro-Doppler augmentations for improved human activity recognition. In 2019 IEEE Radar Conference (RadarConf). IEEE, 1–5.

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. Advances in neural information processing systems 27 (2014).

[8] John C Gower. 1975. Generalized procrustes analysis. Psychometrika 40, 1 (1975), 33–51.

[9] Jiahui Hou, Xiang-Yang Li, Peide Zhu, Zefan Wang, Yu Wang, Jianwei Qian, and Panlong Yang. 2019. Signspeaker: A real-time, high-precision smartwatch-based sign language translator. In The 25th Annual International Conference on Mobile Computing and Networking. 1–15.

[10] Yan Huang, Wei Wang, and Liang Wang. 2015. Bidirectional recurrent convolutional networks for multi-frame super-resolution. Advances in neural information processing systems 28 (2015).

[11] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning. PMLR, 448–456.

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition. 1125–1134.

[13] 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.

[14] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. 2018. End-to-end recovery of human shape and pose. In Proceedings of the IEEE conference on computer vision and pattern recognition. 7122–7131.

[15] Sagi Katz, Ayellet Tal, and Ronen Basri. 2007. Direct visibility of point sets. In ACM SIGGRAPH 2007 papers. 24–es.

[16] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. 2020. Vibe: Video inference for human body pose and shape estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 5253–5263.

[17] 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.

[18] Belal Korany, Chitra R Karanam, Hong Cai, and Yasamin Mostofi. 2019. XModal-ID: Using WiFi for through-wall person identification from candidate video footage. In The 25th Annual International Conference on Mobile Computing and Networking. 1–15.

[19] Hyeokhyen Kwon, Catherine Tong, Harish Haresamudram, Yan Gao, Gregory D Abowd, Nicholas D Lane, and Thomas Ploetz. 2020. IMUTube: Automatic extraction of virtual on-body accelerometry from video for human activity recognition. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 4, 3 (2020), 1–29.

[20] Hyeokhyen Kwon, Bingyao Wang, Gregory D Abowd, and Thomas Plötz. 2021. Approaching the real-world: Supporting activity recognition training with virtual imu data. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 5, 3 (2021), 1–32.

[21] Jiayi Li, Aman Shrestha, Julien Le Kernec, and Francesco Fioranelli. 2019. From Kinect skeleton data to hand gesture recognition with radar. The Journal of Engineering 2019, 20 (2019), 6914–6919.

[22] Kevin Lin, Lijuan Wang, and Zicheng Liu. 2021. End-to-end human pose and mesh reconstruction with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 1954–1963.

[23] Yilin Liu, Fengyang Jiang, and Mahanth Gowda. 2020. Finger gesture tracking for interactive applications: A pilot study with sign languages. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 4, 3 (2020), 1–21.

[24] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. 2015. SMPL: A skinned multi-person linear model. ACM transactions on graphics (TOG) 34, 6 (2015), 1–16.

[25] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. 2013. Rectifier nonlinearities improve neural network acoustic models. In Proc. icml, Vol. 30. Citeseer, 3.

[26] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. 2019. AMASS: Archive of motion capture as surface shapes. In Proceedings of the IEEE/CVF international conference on computer vision. 5442–5451.

[27] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In 2016 fourth international conference on 3D vision (3DV). Ieee, 565–571.

[28] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957 (2018).

[29] NaturalPoint. 2017. Motion Capture Systems - OptiTrack Webpage. https://www.optitrack.com

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32. Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[31] Yili Ren, Zi Wang, Sheng Tan, Yingying Chen, and Jie Yang. 2021. Winect: 3d human pose tracking for free-form activity using commodity wifi. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 5, 4 (2021), 1–29.

[32] Vitor Fortes Rey, Peter Hevesi, Onorina Kovalenko, and Paul Lukowicz. 2019. Let there be IMU data: Generating training data for wearable, motion sensor based activity recognition from monocular rgb videos. In Adjunct proceedings of the 2019 ACM international joint conference

on pervasive and ubiquitous computing and proceedings of the 2019 ACM international symposium on wearable computers. 699–708.

[33] Arindam Sengupta, Feng Jin, Renyuan Zhang, and Siyang Cao. 2020. mm-Pose: Real-time human skeletal posture estimation using mmWave radars and CNNs. IEEE Sensors Journal 20, 17 (2020), 10032–10044.

[34] Cong Shi, Li Lu, Jian Liu, Yan Wang, Yingying Chen, and Jiadi Yu. 2022. mPose: Environment-and subject-agnostic 3D skeleton posture reconstruction leveraging a single mmWave device. Smart Health 23 (2022), 100228.

[35] Bharat Singh, Tim K Marks, Michael Jones, Oncel Tuzel, and Ming Shao. 2016. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In Proceedings of the IEEE conference on computer vision and pattern recognition. 1961–1970.

[36] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. 2017. Veegan: Reducing mode collapse in gans using implicit variational learning. Advances in neural information processing systems 30 (2017).

[37] Shingo Takeda, Tsuyoshi Okita, Paula Lago, and Sozo Inoue. 2018. A multi-sensor setting activity recognition simulation tool. In Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers. 1444–1448.

[38] TI. 1930. Texas Instruments. http://www.ti.com

[39] VICON. 2008. VICON Motion Systems. https://www.vicon.com

[40] Fei Wang, Stanislav Panev, Ziyi Dai, Jinsong Han, and Dong Huang. 2019. Can WiFi estimate person pose? arXiv preprint arXiv:1904.00277 (2019).

[41] Fei Wang, Sanping Zhou, Stanislav Panev, Jinsong Han, and Dong Huang. 2019. Person-in-WiFi: Fine-grained person perception using WiFi. In Proceedings of the IEEE International Conference on Computer Vision. 5452–5461.

[42] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing 13, 4 (2004), 600–612.

[43] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. 2003. Multiscale structural similarity for image quality assessment. In The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, Vol. 2. Ieee, 1398–1402.

[44] 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.

[45] 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.

[46] 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.

[47] Yuichi Yoshida and Takeru Miyato. 2017. Spectral norm regularization for improving the generalizability of deep learning. arXiv preprint arXiv:1705.10941 (2017).

[48] Xiaotong Zhang, Zhenjiang Li, and Jin Zhang. 2022. Synthesized Millimeter-Waves for Human Motion Sensing. In Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems. 377–390.

[49] Yu Zhang, William Chan, and Navdeep Jaitly. 2017. Very deep convolutional networks for end-to-end speech recognition. In 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 4845–4849.

[50] 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. 7356–7365.

[51] Mingmin Zhao, Yingcheng Liu, Aniruddh Raghu, Tianhong Li, Hang Zhao, Antonio Torralba, and Dina Katabi. 2019. Through-wall human mesh recovery using radio signals. In Proceedings of the IEEE International Conference on Computer Vision. 10113–10122.

[52] 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.

[53] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision. 2223–2232.