Synthesized Millimeter-Waves for Human Motion Sensing

Xiaotong Zhang[1,2], Zhenjiang Li[1], Jin Zhang[2]
[1]Department of Computer Science, City University of Hong Kong, Hong Kong, China
[2]Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China

ABSTRACT

Millimeter-wave (mmWave)-based human motion sensing, such as activity recognition and skeleton tracking, enables many useful applications. However, it suffers from a scarcity issue of training datasets, which fundamentally limits a widespread adoption of this technology in practice, as collecting and labeling such datasets are difficult and expensive. This paper presents SynMotion, a new mmWave-based human motion sensing system. Its novelty lies in harvesting available vision-based human motion datasets, for knowing the coordinates of body skeletal points under different motions, to synthesize mmWave sensing signals that bounce off the human body, so that the synthesized signals could inherit labels (skeletal coordinates and the name of each motion) from vision-based datasets directly. SynMotion demonstrates the ability to generate such labeled synthesized data at high quality to address the training-data scarcity issue and enable two sensing services that can work with commercial radars, including 1) zero-shot activity recognition, where the classifier reads real mmWaves for recognition, but it is only trained on synthesized data; and 2) body skeleton tracking with few/zero-shot learning on real mmWaves. To design SynMotion, we address the challenges of both the inherent complication of mmWave synthesis and the micro-level differences compared to real mmWaves. Extensive experiments show that SynMotion outperforms the latest zero-shot mmWave-based activity recognition method. For skeleton tracking, SynMotion achieves comparable performance to the state-of-the-art mmWave-based method trained on the labeled mmWaves, and SynMotion can further outperform it for the unseen users.

CCS CONCEPTS

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

KEYWORDS

Human Motion Sensing; Millimeter Wave; Body Skeleton Tracking; Activity Recognition

ACM Reference Format:
Xiaotong Zhang[1,2], Zhenjiang Li[1], Jin Zhang[2]. 2022. Synthesized Millimeter-Waves for Human Motion Sensing. In *The 20th ACM Conference on Embedded Networked Sensor Systems (SenSys '22), November 6–9, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3560905.3568542

SenSys '22, November 6–9, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9886-2/22/11…$15.00
https://doi.org/10.1145/3560905.3568542

1 INTRODUCTION

Sensing human motion, such as recognizing activities and tracking body skeletons, enables many useful applications. The *vision-based* methods by using cameras now can achieve accurate human motion sensing [36], hinging on the availability of rich labeled datasets and the innovation of machine learning [9, 10]. However, vision-based methods are limited by inherent constraints [36, 48, 56], including line-of-sight views, light conditions, privacy concerns, and more.

To cope with these limitations, emerging breakthroughs explore Radio Frequency (RF) based solutions [23, 50]. One representative example is to use millimeter-wave (mmWave) from frequency-modulated continuous-wave (FMCW) radars [56, 59, 61]. However, unlike vision, mmWave-based training datasets are very **scarce**, since *collecting* such data is labor-intensive and time-consuming, and further *labeling* them, especially for fine-grained tracking services, needs to tightly synchronize mmWaves with the data from high-end cameras to collect the coordinates of skeleton points under different motions [56], which is difficult and expensive [4]. It limits the widespread adoption of mmWave sensing in practice.

In this paper, we propose a new mmWave-sensing system named SynMotion to overcome this issue. The novelty of SynMotion is to leverage available vision-based human motion datasets to synthesize mmWave sensing signals that bounce off human body. The synthesized signals can then inherit labels (including both the coordinates of body skeleton points under different motions and the name of each motion) from vision-based datasets directly. We demonstrate the ability to generate such labeled synthesized data at high quality to address the training-data scarcity issue and enable two useful sensing services that can work with commercial radars directly: 1) **zero-shot** activity recognition, where the classifier reads real signals for recognition, but it is trained only on synthesized data; and 2) body skeleton tracking with **few-shot** learning, which can serve as a seed to further enable *zero-shot* skeleton tracking designs in the future (§3).

Although some researchers have recently explored the signal synthesis for mmWave-based sensing [18, 40], they have focused on synthesizing certain sensing signatures derived from mmWave signals, like the micro-Doppler spectrum [4, 39], little research has been conducted on the synthesis of mmWave itself, which is where various sensing signatures come from — their common **source**. If other sensing signatures (such as heatmaps [61]) are required, or even new signatures are proposed, they can be directly derived from the synthesized mmWaves. Therefore, this paper provides a significant leap and progress with the following significance:

1) Bootstrapping mmWave sensing at low cost. Due to the lack of

training datasets, our design can reduce the expensive data collecting and labeling overhead, which is a meaningful step in bootstrapping the development of mmWave-based sensing systems. With enough developments, more sensing data can be generated and shared in our community to push this technique to maturity.

2) Enhancing sensing interpretability. We delve into the signal itself, with great *interpretability* for mmWave-based sensing. This ability could improve the performance of mmWave-based sensing designs (§4) and facilitate the deployment of sensing systems in the environments, from which it is not easy to deploy external cameras to collect and label training data directly.

However, to reap these benefits, we need to address the following challenges in the design of SynMotion.

1) Synthesizing mmWave sensing signals. Signal synthesis is challenged by the inherent complexity of signal *reflection* and *blocking* from various body parts. In SynMotion, by discretizing the human body into a manageable set of parameters [42], we propose a novel software pipeline to emulate the entire procedure from transmission to reception of the synthesized mmWave signals bouncing off the human body. This pipeline captures the core essence of mmWave-based sensing. The synthesized signals can be used in a zero-shot activity recognition design. They can further serve as a solid foundation for more sophisticated sensing services.

2) Micro-level signal differences. Even though mmWave signals could be delineated, there are still unavoidable subtle differences compared to real signals due to high-order reflections, discrete human models, and others. This could harm more sophisticated (yet useful) sensing services [56, 61] like skeleton tracking — If we train a skeleton tracker on the synthesized data and apply it to real mmWaves directly, the tracking error is large (§4). In fact, this is an emerging "synthetic-to-real" problem in the machine learning domain [9, 10], and the effective solution is to label a small number of real data and design dedicated algorithms to **fine-tune** neural networks trained on synthesized data. Radar can always receive real mmWaves, but the key challenge is how to obtain the skeleton point coordinates corresponding to these mmWaves (for fine-tuning), which again requires expensive labeling overhead.

To address this challenge, we propose a novel training framework. For the actual skeleton tracker, we introduce a variant for it. The *variant tracker* has the same network structure as the *actual tracker* but it takes user's initial pose of each motion as an additional input. Both trackers are trained by the synthesized data first. Because the initial body pose well describes a user's body shape, which is user-specific information and is provided explicitly, we find that it makes the variant tracker have a good *user-independent* feature after fine-tuning by using a third-party mmWave dataset, denoted as **RadarSet**.[1] Now, when we collect a few mmWaves from the target users to fine-tune the actual tracker, we also record their initial pose for each motion (such as using a smartphone to record and extracting the pose from the image), which are fed to the variant tracker. Since the variant tracker has good user-independent feature, it can directly estimate the skeleton point coordinates corresponding to these collected mmWaves, and we treat such estimated coordinates as pseudo labels to fine-tune the actual tracker. Because the variant tracker needs the initial body pose very time, we only use it as a coordinate generator to fine-tune the actual tracker.

[1]RadarSet contains the skeleton point coordinates under various motions from a group of people and the mmWave sensing signals received at the same time, which are used to fine-tune the variant tracker in SynMotion.

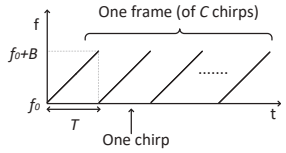

Figure 1: Illustration of FMCW chirps in one frame.

If RadarSet is public available online, the training framework above does not need to deploy any expensive cameras for the labeling purpose, *i.e.*, nearly **zero-shot** training effort. Even though target users' body shapes and how they perform each motion may not be the same as the people in RadarSet, fine-tuning of the actual tracker can handle such differences well (§3). However, as far as we know, there is no RadarSet available now. Hence, we take a pioneer step and use our collected mmWaves to build one, which could serve as a seed to attract more data contributions from our community. Since we collect a few mmWaves to form the RadarSet and use it to fine-tune the variant tracker in SynMotion, it results in a *few-shot* skeleton tracking design in this paper.

To validate the efficacy of SynMotion, we develop a prototype system using TI IWR1443BOOST radars. We also deploy an OptiTrack system to collect the ground truth of skeleton coordinates for evaluation. We have recruited 20 users (9 F and 11 M) to perform eight activities. To launch the signal synthesis, we have adopted both our collected vision-based dataset from OptiTrack and two public datasets, NTU RGB+D [44] and CMU MoCap [13]. Extensive results show promising performance. SynMotion achieves the average error of 5.8 cm to track 19 skeleton points. It is comparable to the-state-of-the-art RF-Pose3D [61] with an error of 5.3 cm, which is trained on the labeled real mmWaves directly. SynMotion can further outperform it for the unseen users by 21% to 48%. For activity recognition, SynMotion achieves 94.1% accuracy for zero-shot activity recognition, outperforming the accuracy of 84.9% by the latest zero-shot design Vid2Doppler [4] that synthesizes micro-Doppler spectrum directly. Our project site is at https://synmotion.github.io/. In summary, this paper makes the following contributions:

- We propose to synthesize the mmWave sensing signals — the source of various sensing signatures adopted before. Our design can bootstrap mmWave-based sensing with largely reduced training overhead and improving the performance of various sensing services and applications.
- We propose a series of novel and effective technologies to address two challenging issues, which are encountered in synthesizing the mmWave sensing signals and developing the more advanced skeleton tracking service.
- We develop a prototype system of SynMotion and conduct extensive experiments. Results show promising performance gains for both activity recognition and skeleton tracking compared with the state-of-the-art methods.

2 PRELIMINARY

2.1 FMCW Radio

Frequency-modulated-continuous-wave (FMCW) is a technology that can provide distance and velocity measurements of the targets.

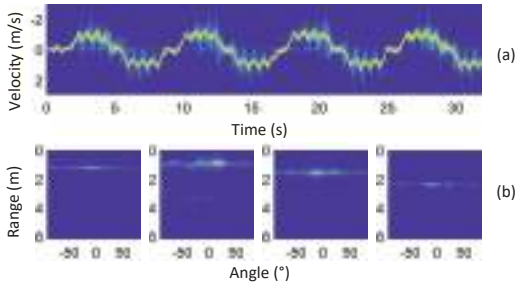

Figure 2: (a) Micro-Doppler spectrum derived from an mmWave sensing trace for walking. (b) Four heatmaps derived from four chirps from the walking motion trace.

FMCW radar transmits an FMCW signal called a **chirp**, which is usually in an millimeter-wave (mmWave) frequency band, such as 77–81 GHz [22]. A chirp is a sinusoid with the frequency linearly increasing over time, as shown in Figure 1, which can be characterized by a start frequency f_0, bandwidth B and chirp duration T, for example, f_0 = 77 GHz, B = 3.6 GHz and T = 32 ms. Multiple (denoted as C, such as C = 128) chirps further form one **frame**. Each received (Rx) chirp is a time-delayed version of the transmitted (Tx) chirp. Radar circuit generates an intermediate frequency (**IF**) signal of a constant frequency that equals to the frequency difference of Tx and Rx chirps, based on which radar can determine the range, velocity, and angle of the reflection object. More information on FMCW radar can be found in [22].

2.2 Sensing Signatures and Applications

Sensing signatures. With the sensing ability of FMCW radar, researchers have proposed to derive *micro-Doppler spectrum* and *heatmap* two popular sensing signatures, derived from raw chirp signals, and use them to enable various sensing designs.

1) Micro-Doppler spectrum is a 2D velocity-time sensing signature, which is generated by the short-time Fourier transform (STFT) on IF signals. One motion trace produces one such signature usually, which can be treated as an 2D image (as depicted in Figure 2(a)) and is convenient to be processed by a neural network for activity recognition. Existing mmWave synthesis works [4, 18, 39, 40] all synthesize micro-Doppler spectrum directly.

2) Heatmap is also a 2D signature (Figure 2(b)), but it describes the relationship between range and angle of the object. Specially, a range-FFT operation is first performed on the IF signals from each antenna to derive the range information, and the angle-FFT operation is further applied to the corresponding peaks across antennas to estimate the angle. As heatmaps are derived in a chirp basis, one motion trace usually produces a series of heatmaps, which are widely used in skeleton tracking designs [56, 59, 61]. So far as we know, heatmaps have not been synthesized in prior works yet.

Different from existing works, in this paper, we aim to synthesize the "source" (*i.e.*, the mmWave signal itself) of various signatures to enable sensing services flexibly and also improve the sensing performance. If new sensing signatures are proposed in the future, they can be also derived from the synthesized mmWaves.

Applications. A wide spectrum of useful applications can be enabled by mmWave-based sensing, such as the *monitoring* of elderly people, so that accidents can be detected in time yet without using video to jeopardize their privacy (*e.g.*, in bathroom) [23], *novel HCIs* that can augment systems like Kinect to work across occlusions [59], *virtual reality* gaming [50], etc. When mmWave sensing is mature in the future, it can be further applied for the more sophisticated scenarios, *e.g.*, the theft detection in shopping malls where thieves likely cover their hand movements by clothes [56] or police assessment of a hostage scenario behind a door [61].

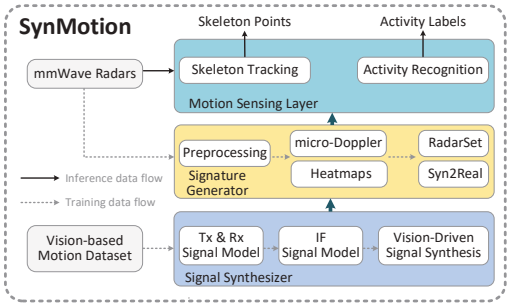

Figure 3: Overview of the SynMotion architecture.

SynMotion essentially rehearses mmWave-based sensing events. Thus, it has the potential to enable more sensing services that might not be feasible before, for example, rehearsing sensing performance in different locations to determine the best radar location before actual deployment. We will explore such possibilities in the future.

2.3 SynMotion Architecture

Figure 3 illustrates the architecture of the SynMotion design. With the vision-based data, containing both the instant skeletal point coordinates under different motions and the name of each motion, the signal synthesizer emulates the mmWave sensing procedure from the transmission to the reflection and receiving of mmWave chirps (§3.1). With the received chirps, the synthesizer further derives IF signals, which are used to generate various signatures for different sensing purposes. The micro-Doppler spectrum obtained from the synthesized mmWaves can be used to train a classifier for zero-shot activity recognition first in SynMotion (§3.2.1).

The sensing signature generator further handles the micro-level signal differences to enable skeleton tracking (§3.2.2). To address this problem, we introduce a variant of the actual tracker in SynMotion. The variant tracker is fine-tuned by using RadarSet and it can replace external cameras to generate skeletal point coordinates for fine-tuning the actual tracker. After fine-tuning, the actual tracker can read real mmWaves for skeleton tracking. This design can avoid deploying expensive cameras for the labeling purposes.

3 SYSTEM DESIGN

3.1 Sensing Signal Synthesizer

In this section, we formulate the procedure from the transmitted mmWave FMCW signal to the received one after reflections on human body to obtain the corresponding IF signal by using the vision-based motion data. We introduce a synthesis pipeline with the following steps in SynMotion.

1) Transmitted signal. For each transmitting (Tx) and receiving (Rx) antenna pair, a transmitted FMCW mmWave is a sinusoidal-like

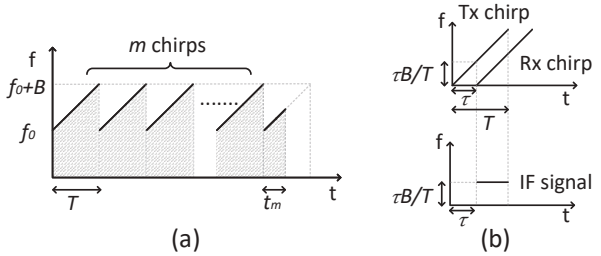

Figure 4: (a) The shaded area under the transmitted chirps represents the number of signal periods experienced since the current frame starts. (b) Illustration of the IF signal.

signal [22]. To model it, we should consider time t across various chirps within each frame. We suppose that m chirps ($0 \leqslant m \leqslant C-1$, where C is the number of chirps in one frame) have been sent out in one frame and the current $(m+1)^{th}$ chirp is being transmitted for time t_m already. Thus, the time t with respect to the beginning of each frame and the instant frequency $f(t)$ of the transmitted chirp signal can be written as:

$$t = m \times T + t_m, \text{ and } f(t) = f_0 + B \times \frac{t_m}{T}, \tag{1}$$

respectively, based on which we can then mathematically express the transmitted mmWave signal $S_{TX}(t) = A \times e^{j\phi}$, where A is the amplitude and ϕ is the phase.

As shown in Figure 4(a), the shaded area under all transmitted chirps in the current frame, *i.e.*, $\int_0^{mT+t_m} f(x) \cdot dx$, represents the number of signal periods experienced since this frame starts. Since signal phase ϕ changes 2π after each period, we have

$$\begin{aligned}
\phi &= 2\pi(\int_0^{mT+t_m} f(x) \cdot dx) + \phi_0, \\
&= 2\pi(\int_0^{mT} f(x) \cdot dx + \int_0^{t_m} f(x) \cdot dx) + \phi_0, \\
&= 2\pi(m\frac{(f_0 + f_0 + B)T}{2} + \frac{(f_0 + f_0 + B\frac{t_m}{T})t_m}{2}) + \phi_0, \\
&= 2\pi(f_0 t + \frac{B(t_m^2 + mT^2)}{2T}) + \phi_0,
\end{aligned} \tag{2}$$

where ϕ_0 is the initial phase. Then, by substituting ϕ from Eq. (2) into $S_{TX}(t) = A \times e^{j\phi}$, we obtain:

$$S_{TX}(t) = Ae^{j(2\pi(f_0 t + \frac{B(t_m^2+mT^2)}{2T}) + \phi_0)}. \tag{3}$$

2) Received signal from a single reflection point. If the transmitted signal $S_{TX}(t)$ is bounced off a *single* reflection point at distance D with respect to the receiving antenna, the received signal $S_{RX}(t)$ can be viewed as a time-delayed (and also attenuated) version of the transmitted signal and latency τ is $\frac{2D}{c}$, where c is the light speed. Thus, the received signal $S_{RX}(t)$ can be expressed as:

$$S_{RX}(t) = A' e^{j(2\pi(f_0(t-\tau) + \frac{B((t_m-\tau)^2+mT^2)}{2T}) + \phi_0)}, \tag{4}$$

where A' is the attenuated amplitude. It can be obtained according to the radar communication principle [41]:

$$A' = \frac{G_{Tx} G_{Rx} \lambda \sqrt{P\sigma}}{(4\pi)^{1.5} D^2}, \tag{5}$$

with Tx/Rx antenna gain $G_{Tx/Rx}$, wavelength λ, transmission power P and radar cross section (RCS) σ. The parameters $G_{Tx/Rx}$ and P are determined according to radar configuration and the value of RCS σ can be estimated after we introduce human body model.

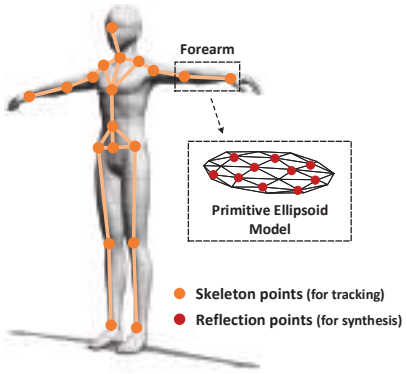

Figure 5: The skeleton tracking design in SynMotion can output 3D coordinates of 19 skeleton points over time. For the signal synthesis, we adopt a pictorial body model composed of primitive ellipsoids, *e.g.*, illustrated for the fore-arm.

3) IF signal from a single reflection point. When $S_{RX}(t)$ is received, according to the working flow of FMCW radar, $S_{RX}(t)$ is combined with the transmitted signal by a "mixer" to generate the intermediate frequency (IF) signal $S_{IF}(t)$ [22], which is the signal used by radar for sensing. The "mixer" measures the instantaneous frequency and phase differences of these two signals, as depicted in Figure 4(b). Thus, we can mathematically derive $S_{IF}(t)$ from a single reflection point as follows:

$$\begin{aligned}
&A' e^{j(2\pi(f_0 t + \frac{B(t_m^2+mT^2)}{2T}) + \phi_0 - (2\pi(f_0(t-\tau) + \frac{B((t_m-\tau)^2+mT^2)}{2T}) + \phi_0))}, \\
= \; &A' e^{j2\pi(f_0\tau - \frac{B\tau^2}{2T} + \frac{Bt_m\tau}{T})},
\end{aligned} \tag{6}$$

where the first term "$2\pi(f_0 t + \frac{B(t_m^2+mT^2)}{2T}) + \phi_0$" is from $S_{TX}(t)$ in Eq. (3) and the second term "$2\pi(f_0(t-\tau) + \frac{B((t_m-\tau)^2+mT^2)}{2T}) + \phi_0$" is from $S_{RX}(t)$ in Eq. (4) for this calculation.

4) Reflections from human body. With $S_{IF}(t)$ derived from a single-point reflection in Eq. (6), we further extend it to the scenario that the transmitted signal $S_{TX}(t)$ bounces off multiple points on human body. To this end, we adopt the 3D pictorial representation to model human body, composed of primitive ellipsoids to represent each body part [51], where Figure 5 illustrates for the fore-arm. The merit of using this model is that the core effect of the body reflection can be described by using parameters explicitly, which are needed in the subsequent signal synthesis.

For each part k of our body, $k \in \mathcal{K} = \{forearm, upperarm, torso, calf, thigh, head\}$, we consider a finite number of reflection points, denoted as n that balances a trade-off between the tracking accuracy and computation overhead as investigated in §4. For reflection points from an ellipsoid, the estimation of radar cross section (RCS) σ in Eq. (5) has been investigated and included in the RCS handbook $\mathcal{H}(\cdot)$ [14, 40] before as follows:

$$\sqrt{\sigma} = \left[\frac{\frac{1}{4}\pi R_k^4 H_k^4}{R_k^2 sin^2\theta_k + \frac{1}{4}H_k^2 cos^2\theta_k}\right]^{\frac{1}{2}} e^{-j\frac{2\pi}{\lambda}2D}, \tag{7}$$

where θ_k is the angle of the incident wave relative to the height axis

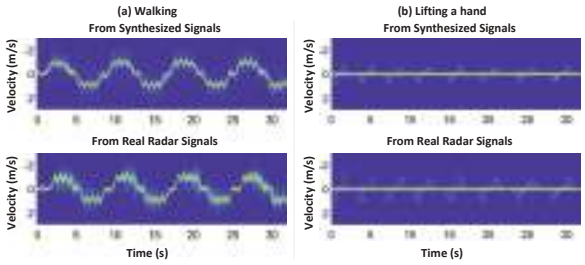

Figure 6: For two activities (a) walking and (b) lifting a hand, the 1^{st} row shows the micro-Doppler spectrum derived from our synthesized signal, and the 2^{nd} row shows the corresponding micro-Doppler spectrum from real mmWaves.

of the ellipsoid with height H_k and radius R_k. After σ is obtained, all the parameters in $S_{IF}(t)$ from a single reflection point are known, and the final IF sensing signal $S(t)$ from each Tx-Rx antenna pair can be obtained by:

$$S(t) = \sum_{k \in \mathcal{K}} \sum_{i=1}^{n} S^i_{IF}(t), \tag{8}$$

$$s.t. \quad Tx \text{ and } Rx \text{ signals are not blocked by body}, \tag{9}$$

$$\sigma \in \mathcal{H}(ellipsoid), \tag{10}$$

where only $S^i_{TX}(t)$ and $S^i_{RX}(t)$ not blocked by human body are counted (this can be calculated in the next synthesis step) and Eqs. (8)–(10) can be directly extended to multiple Tx-Rx pairs.

5) Signal synthesis with the vision-based data. In the formulation above, we consider that the distance D between each reflection point and the receiving antenna is known. In the real signal synthesis, each vision-based human motion dataset implicitly contains a 3D space, in which the coordinates of user's skeleton points are represented. Then, for each snapshot (frame) of user's body pose, we can compute the coordinates r ($=< r_x, r_y, r_z >$) of each reflection point on the user's body (Figure 5) from the skeleton point coordinates by interpolation when the ellipsoid parameters are known, such as H_k and R_k in Eq. (7). The length H_k of each body part can be obtained from skeleton point coordinates and the ellipsoid radius R_k is estimated through [49] in our current development.

Then, we can virtually deploy radar in this 3D space, where the radar position is denoted as $p =< p_x, p_y, p_z >$, and the distance D is the Euclidean distance $d(\cdot)$ from radar to each reflection point:

$$D \quad = \quad d(p, r) = \sqrt{(p_x - r_x)^2 + (p_y - r_y)^2 + (p_z - r_z)^2},$$

based on which we can obtain each $S^i_{IF}(t)$ to synthesize $S(t)$ from Eqs. (8)–(10) under various motions. Moreover, if any transmitted or received signal intersects with a body part, it is excluded from the synthesis. It is usually sufficient to deploy the virtual radar at one position, which is consistent with where the radar is planed to deploy in the real sensing field with similar facing direction and working distance. Before we continue, we note three points:

i) Vision-based data selection. When we select motion data from a vision-based dataset, the selected data should cover all the motions to be tracked using mmWaves. SynMotion (and most existing mmWave-based designs [56, 59, 61]) currently cannot track body poses well for the unseen motions.

ii) Domain differences. Even for the same motion, such as walking, users in the vision-based dataset may not perform it in exactly the same way as actual users in real environments. In addition, the body shapes of dataset users may also be different from actual users. They are the *domain differences*. The synthesized data can teach a tracker how to work in the dataset domain and we then can further compensate domain differences (§3.2.2) to adapt the tracker to a real environment without expensive labeling overhead.

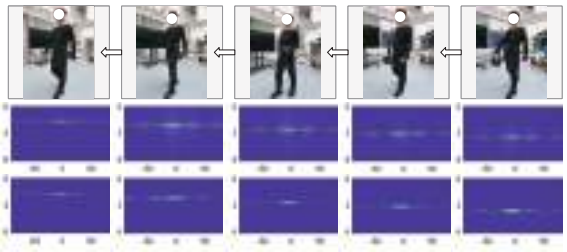

Figure 7: Five heatmaps from our synthesized signals (the 2^{nd} row) and the corresponding real mmWave signals (the 3^{rd} row) in the motion of walking.

iii) Reflections from the environment. In signal synthesis, we do not explicitly consider reflections from the environment, such as nearby objects, due to two reasons. First, radars can be initially calibrated to remove most reflections from the environment. Thus, the sensing signals mainly capture the user's motions and existing mmWave-based sensing systems [56, 59, 61] indeed show good performance in the unseen environments directly after initial calibration. We also show it in §4. Second, higher-order reflections (from body to objects to radar) are much weaker than direct reflections, which are marginal for activity recognition and can be compensated for the more complicated skeleton tracking (§3.2.2).

3.2 Sensing Signature Generator

The synthesized IF signal $S(t)$ can be used to derive various sensing signatures and enable different sensing services.

3.2.1 Sensing signatures. As a proof of concept, we examine *micro-Doppler spectrum* and *heatmap* two popular ones in this paper.

1) Micro-Doppler spectrum. We first investigate micro-Doppler spectrum, which has been widely used for activity recognition [4, 18, 39, 40]. Figure 6 visualizes the results for two daily activities, walking and lifting a hand. For each activity, the first row in Figure 6 shows the micro-Doppler spectrum derived from our synthesized signal based on the vision-based data collected from OptiTrack. The second row shows the micro-Doppler spectrum derived from the real mmWave sensing signals that are received concurrently. The experimental setup is detailed later in evaluation.

Recent studies [4, 39] have shown the possibility to synthesize micro-Doppler spectrum directly using deep learning for a zero-shot activity recognition, *i.e.*, the classifier is trained by using the vision-based data only and it can work with real mmWaves directly after training. Figure 6 suggests that SynMotion can derive high-quality micro-Doppler spectrum signatures from our synthesized mmWaves, even if there are inevitable differences between the synthesized signal and the received real signal, which can improve the recognition accuracy significantly as revealed via evaluation.

2) Heatmap. Then, we investigate heatmap, which has been commonly adopted for skeleton tracking [56, 59, 61]. As introduced in

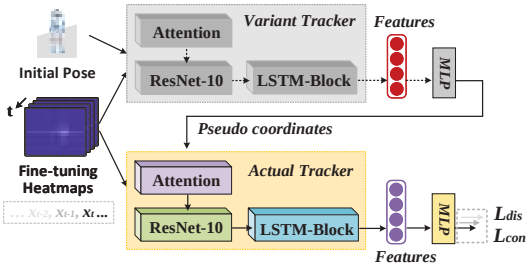

Figure 8: Our "synthetic-to-real" training framework.

§2, a heatmap can be viewed as one 2D (distance vs. angle) image, which is generated in a chirp basis. One motion trace usually produces a series of heatmaps. Figure 7 compares the synthesized and actual heatmaps from five chirps from the motion of walking, which are also similar to each other. However, as skeleton tracking is a subtle sensing task, the micro-level differences between synthesized and real signals can degrade tracking accuracy significantly. As observed through experiments, if we develop a zero-shot skeleton tracker directly, the tracking error is very large, *e.g.*, 19.17 cm on average. Therefore, we propose a novel framework in SynMotion to further improve the tracking accuracy.

3.2.2 Generalization to real mmWaves for skeleton tracking. In fact, the difference between the synthesized and real data is an emerging "synthetic-to-real" topic from the transfer learning and domain adaptation fields [11, 38]. Recent machine learning studies have explored effective solutions by labeling a few real data and then designing efficient algorithms to fine-tune the neural network. Because the neural network is pre-trained by synthetic data first, only a small number of real data is needed for fine-tuning. After fine-tuning, the network can work with the real data.

Problem. For skeleton tracking, we can ask users to perform various motions during the system setup phase. It is easy to collect real sensing signals as radar always receives mmWaves when it is working. The collection of fine-tuning data itself does not bring much overhead, but the main challenge is how to annotate the corresponding coordinates of skeleton points for these mmWaves (to perform fine-tuning) if we do not deploy any external cameras for labeling. Therefore, we cannot apply existing "synthetic-to-real" methods directly and need to design a new solution in SynMotion.

Proposed framework. Figure 8 depicts the training framework proposed in SynMotion, which contains two trackers. One is the *actual* skeleton tracker $\mathcal{T}$, and the other one is a *variant* tracker $\mathcal{T}_v$ designed to assist the fine-tuning of the actual tracker.

- The actual tracker $\mathcal{T}$ is similar to most mmWave-based tracker designs [56, 59, 61], which reads real mmWaves (mw_t) as the only input to estimate the coordinate (c_t^i) of each skeleton point i, *i.e.*, $\mathcal{T}(mw_t) \rightarrow \{c_t^i\}_{i=1}^N$, where the number of skeleton points N is 19 in SynMotion.
- The variant tracker $\mathcal{T}_v$ has the same network structure as the actual tracker. However, in addition to mmWaves, it requires a second input, *i.e.*, the user's initial body pose (ip) at the start of each motion. This variant tracker is only used to assist the fine-tuning of the actual tracker.
- Because the user's initial body pose is provided, the variant tracker is designed to output the *displacement* (d_t^i) of each skeleton point relative to its initial coordinate (the reason is stated below), *i.e.*, $\mathcal{T}_v(mw_t, ip) \rightarrow \{d_t^i\}_{i=1}^N$.[2] Then, the initial coordinate pluses a displacement also gives the instant coordinate (c_t^i) of each skeleton point, *i.e.*, $ip + \{d_t^i\}_{i=1}^N \rightarrow \{c_t^i\}_{i=1}^N$. Therefore, we still denote $\mathcal{T}_v(mw_t, ip) \rightarrow \{c_t^i\}_{i=1}^N$.

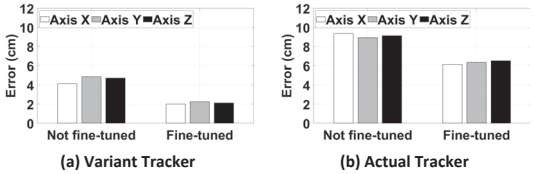

Figure 9: Tracking errors of the variant tracker and the actual tracker under different settings.

Our training framework then follows three steps to generalize the actual tracker $\mathcal{T}$ to the real mmWave sensing signals.

Step-1: The actual tracker $\mathcal{T}$ and variant tracker $\mathcal{T}_v$ are first trained on the same set of mmWaves synthesized from a vision-based dataset, where the user's initial pose of each motion trace (needed by $\mathcal{T}_v$) can be obtained from the initial coordinates of user's skeleton points. This step aims to train a primary version for both trackers. So far, both trackers are not ready to work with real mmWaves yet. As Figure 9 shows, without fine-tuning from next two steps, their tracking errors are large, such as 4.72–9.34 cm.

Step-2: The second step fine-tunes the variant tracker $\mathcal{T}_v$ only. After this step, the variant tracker can perform skeleton tracking on real mmWaves when given an initial pose of the user. The role of a fine-tuned $\mathcal{T}_v$ is to replace the external cameras to annotate the corresponding skeleton point coordinates for the real mmWaves used in the fine-tuning of the actual tracker in Step-3.

2.a) Fine-tune the variant tracker $\mathcal{T}_v$. If some people have labeled and released an mmWave dataset RadarSet, containing the skeleton points coordinates under various motions from a group of users and the mmWave sensing signals received at the same time, we can use RadarSet to fine-tune $\mathcal{T}_v$. We first assume that we had RadarSet and how to obtain it is introduced later. A key observation in this step is that after fine-tuning by RadarSet, the variant tracker can work with real mmWaves to track **unseen** users (not in RadarSet) well. Figure 9(a) shows that the tracking error of the fine-tuned variant tracker on a set of unseen users is only 2.0–2.2 cm along each axis. More experiments are conducted in evaluation (§4).

2.b) Why $\mathcal{T}_v$ has this ability? Since the initial body pose, required by the variant tracker $\mathcal{T}_v$, fully encompasses the user's body shape, such as arm and leg length, what the variant tracker $\mathcal{T}_v$ has to learn in the training or fine-tuning becomes "easier" than the actual tracker $\mathcal{T}$ — the tracked body skeleton implicitly contains the body shape information, which the actual tracker $\mathcal{T}$ needs to learn on its own, but which is provided directly to the variant tracker $\mathcal{T}_v$. It is why the error ("Not fine-tuned" in Figure 9) of the actual tracker is much larger than the variant tracker when they are trained on the same synthesized data and tested on the same unseen users.

2.c) Can we use RadarSet to fine-tune the actual tracker? Unfortunately, this is infeasible due to the difference between $\mathcal{T}$ and $\mathcal{T}_v$

[2]WiPose [23] also takes initial body pose as input and infers the displacement of each skeleton point for Wi-Fi based body tracking, while our variant tracker exploits this network structure to obtain a useful feature to fine-tune the actual tracker.

stated in 2.b). In Figure 9, we use RadarSet to fine-tune the actual tracker $\mathcal{T}$ and find that its tracking error is large when $\mathcal{T}$ works on the same set of unseen users. Although the variant tracker becomes relatively *user-independent* after fine-tuning, we cannot use it to replace the actual tracker since $\mathcal{T}_v$ needs initial pose every time, while we can leverage its user-independent feature in the next step.

Step-3: The last step fine-tunes the actual tracker $\mathcal{T}$. After this step, $\mathcal{T}$ can use real mmWaves for skeleton tracking.

To train a tracker, existing designs [56, 59, 61] collect the real sensing mmWaves from the target users in the field and further deploy external cameras to label the coordinates of skeleton points for these mmWaves. To avoid this deployment and manual labeling efforts, we leverage $\mathcal{T}_v$ to replace these cameras.

In particular, we collect a few real sensing mmWaves from the target users and also record their initial body poses for each motion (that can be captured by using a smartphone and extracted from the image or manually measured), which are feed to the variant tracker. As the variant tracker is user-independent, it can estimate the skeleton point coordinates for all these mmWaves, and we treat such coordinates as (pseudo) labels to fine-tune the actual tracker. This mmWave collection does not introduce much overhead, because while a radar is working, it is constantly receiving mmWaves.

In summary, step-1 uses a large amount of synthesized data to first teach the actual tracker how to track skeleton points from mmWaves. The rest two fine-tuning steps use a few real mmWaves to further close following two gaps, so that the actual tracker is adapted from the synthesized data domain to the real data domain.

1) Body parameters. The physical parameters (such as arm and leg length) of users from the vision-based dataset, and RadarSet may differ from real users and real users are not required to be involved in these two datasets. This difference is mainly compensated when the variant tracker adopts the initial poses (that implicitly include body parameters) of the real user to generate corresponding pseudo labels to fine-tune the actual tracker.

2) Motions. Even though the data selected from the vision-based dataset and RadarSet covers all the motions to be tracked using mmWave, *i.e.*, no unseen motions, real users may perform each motion differently from both datasets. This is the essential difference between synthesized and real data, which is mostly compensated by fine-tuning itself, since data from real users is adopted.

RadarSet. It is an important dataset in this framework, containing

- 1) skeleton point coordinates under various motions, and
- 2) mmWave sensing signals received at the same time.

RadarSet is only used to train the variant tracker $\mathcal{T}_v$. If such a dataset is public available online, the entire training in SynMotion does not need the deployment of external and expensive cameras for labeling purposes, and skeleton trackers can be developed by reusing RadarSet and our framework directly, *i.e.*, a nearly **zero-shot** training effort. However, to our best knowledge, there is no such a dataset right now. To take a pioneer step, we deploy an OptiTrack and radar to label such a dataset, which thus results in a *few-shot* training in this paper. This dataset can serve as a seed to attract more data contributions (to cover more activities and user body shapes). With an enriched RadarSet, a variant tracker with better user-independent ability can be obtained after fine-tuning, which can facilitate skeleton tracker development in the future.

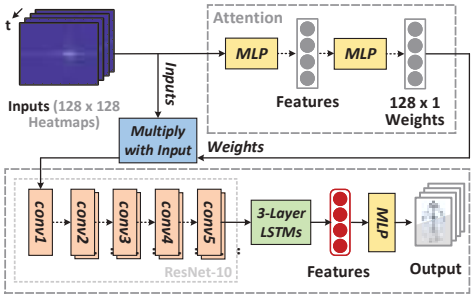

Figure 10: Skeleton tracker design, which reads heatmaps as input to estimate the 3D coordinates of 19 skeleton points.

Network design of skeleton tracker. After we obtain training and fine-tuning data with SynMotion, we find that it is enough to design a simple and compact neural network to achieve good skeleton tracking, which is easy to train and converge. Figure 10 illustrates our current development. The network reads 2D heatmaps derived from chirps as input and outputs the estimated 3D coordinates of 19 skeleton points. The variant tracker has a same network structure, but it takes user's initial pose as an additional input.

The main rationale of this network design in Figure 10 is to extract the spatial features from each heatmap (like a 2D image) first and then explore the temporal features cross heatmaps (to capture the continuous motions) before the estimation of skeleton points. Following this principle, the network contains four blocks:

1) CNN blocks. We utilize the ResNet-10 backbone [19] with five stacked CNNs. They take U (*e.g.*, $U = 100$) heatmaps $\{x_t\}_{t=1:U}$ as input to extract the input spatial features $z_{t=1:U}$, where x_t is the concatenated heatmaps ($x_t =< x_t^h, x_t^v >$) of two individual heatmpas x_t^h and x_t^v obtained from horizontal and vertical radars, respectively. The reason of using ResNet-10 is to avoid the vanishing gradient problem with multiple CNN layers [19].

2) RNN blocks. For mmWaves, human body usually acts as a reflector rather than a scatter. Therefore, only a subset of reflected mmWaves are received by radars [3], and we need to further consider the relationship of chirps over time. To this end, we feed the features $z_{t=1:U}$ extracted from CNNs to LSTM layers to produce the *spatial-temporal* features $\{f_t\}_{t=1:U}$.

3) Attention block. Because the meaningful information resides in a small part on each heatmap, we add an *attention* block to let network concentrate on such meaningful parts more effectively. Attention [37] is a technique using dynamic weights to prioritize the more crucial parts from the extracted features. In Figure 10, the attention weight vector w_t is multiplied to x_t to update this input, *i.e.*, $x_t = w_t x_t$, before the CNN blocks.

4) MLP layer. The multi-layer perception (MLP) finally predicts U snapshots of the 3D coordinates $\{C_t\}_{t=1:U}$ of 19 skeleton points from U input heatmaps, where $C_t = \{c_t^i\}_{i=1}^{19}$ and each c_t^i is the 3D coordinate of skeleton point i.

To train each tracker, we introduce two loss functions in our current implementation. The first loss $\mathcal{L}_{dis}$ aims to make the estimated coordinates c_t^i similar to their corresponding labels $\hat{c}_t^i$, where

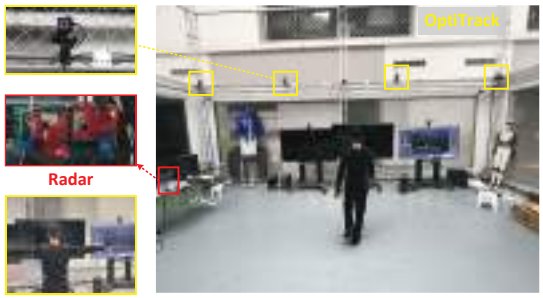

Figure 11: Illustration of our testbed.

the difference can be measured by their Euclidean distance:

$$\mathcal{L}_{dis} = \frac{1}{U}\sum_{t=1}^{U}\frac{1}{N}\sum_{i=1}^{N}\sqrt{(c_t^i - \hat{c}_t^i)^2},$$

where N is the number of skeleton points. The loss $\mathcal{L}_{dis}$ considers the estimation performance independently each time. Because human motions are continuous over time, we further introduce another loss $\mathcal{L}_{con}$ to make the difference between the estimated bod poses across consecutive times similar to the labeled ones:

$$\mathcal{L}_{con} = \frac{1}{U-1}\sum_{t=2}^{U}\frac{1}{N}\sum_{i=1}^{N}\|(c_t^i - c_{t-1}^i) - (\hat{c}_t^i - \hat{c}_{t-1}^i)\|,$$

where $\|\cdot\|$ stands for the Huber norm [20]. With these two loss functions, the network training aims to minimize the overall loss function $\mathcal{L} = \mathcal{L}_{dis} + \alpha \times \mathcal{L}_{con}$, where α is a hyper-parameter. We minimize the loss function $\mathcal{L}$ by using the Adam optimizer [24].

In this training framework, after we finish the fine-tuning of the variant tracker $\mathcal{T}_v$ in step-2, we *freeze* its parameters $\tilde{\theta}_{T_v}$ to fine-tune the parameters $\tilde{\theta}_T$ of the actual tracker $\mathcal{T}$ in step-3.

4 EVALUATION

4.1 Implementation

Setup. Figure 11 shows our testbed using TI IWR1443BOOST radar, which is portable and small in size (8 cm × 7 cm). Each radar has two Tx and four Rx antennas. We mount two radars at a height of 0.9 m to cover the horizontal and vertical directions. We also deploy an OptiTrack motion capture system with 12 cameras to collect the ground truth for evaluation and build the RadarSet dataset.

Radar transmits 20 frames per second (fps), and a frame contains 128 chirps from each Tx antenna. For a chirp, the start frequency f_0, bandwidth B and chirp duration T are set to be $f_0 = 77$ GHz, $B = 3.6$ GHz and $T = 32$ ms in the experiment. With the current setting, the maximum sensing range is 6.2 m and ranging resolution is 4.9 cm. The maximum velocity and velocity sensing resolutions are ±2.5 m/s and 2.9 cm/s, respectively. The average distance between the activity area in the testbed and the radars is about 3 m.

Data collection. We have recruited 20 users to participate into the data collection for evaluation. This study has received the university's ethical approval. We have collected data on eight different activities. In addition to our data collection, we also use two public datasets. We use the sensing data from eight activities for both skeleton tracking and activity recognition. These activities cover many daily activities studied in recent mmWave-based sensing designs, including 1) walking, 2) swinging arms, 3) swinging hands, 4) lifting a hand, 5) lifting a leg, 6) sitting, 7) using a phone, and 8) chatting. These activities contain the movements of isolated body parts and the whole body for a comprehensive evaluation.

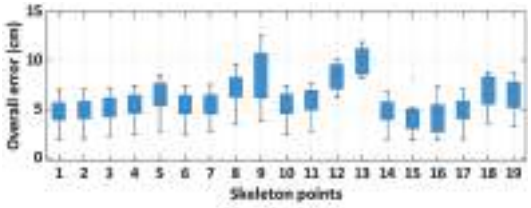

Figure 12: Overall error distribution of each skeleton point.

1) Our dataset. At the beginning of data collection, we perform initial calibration for radars to remove reflections from environment. During data collection, we use OptiTrack to collect the ground truth of the coordinates for 19 skeleton points at 120 fps, which are downsampled to be the same as our tracker's output rate at 20 fps for evaluation. Similar to [59, 61], we timestamp OptiTrack frames and heatmaps to synchronize two data streams. For each activity, users perform it five rounds, each taking approximately 30 seconds. We divide users into two groups for the following purposes:

- *Group-a)* contains 10 users (4F and 6M) to collect our vision-based dataset for signal synthesis. We also use the mmWave sensing signals from these 10 users to build the first version of RadarSet to fine-tune the variant tracker.
- *Group-b)* contains another 10 **new** users (5F and 5M) for evaluation only, whose data does not appear in Group-a). For each new user, we feed one round of sensing data to the variant tracker to generate pseudo skeleton point coordinates (to fine-tune the actual tracker), and the rest for evaluation.

2) Public datasets. In addition to our collected vision-based data, we also use two public datasets, NTU RGB+D [44] and CMU MoCap [13], to trigger signal synthesis, and then test the system performance on the 10 users from Group-b) for evaluation.

Network training. For each vision-based dataset, we perform the signal synthesis to obtain synthesized IF signals per activity per user from the vision-based dataset. For each vision-based dataset, we use the synthesized signals to train a primary version of the variant tracker and the actual tracker. Then, we use the data of 10 users from Group-a) to fine-tune each variant tracker and further use the fine-tuned variant tracker to generate pseudo coordinates for each one-round sensing data from 10 new users in Group-b) to fine-tune the actual tracker. In our current development, the actual tracker outputs the coordinates of 19 skeleton points at 20 fps.

4.2 Overall Performance

We first examine the skeleton tracking performance. In this evaluation, we compare the following two mmWave-based methods:

- **RF-Pose3D** [61]: the state-of-the-art mmWave-based skeleton tracking method, which is trained on the labeled mmWaves. We examine three versions of RF-Pose3D for a comprehensive evaluation, which are detailed soon.
- **SynMotion**: the method proposed in this paper.

Performance of SynMotion. We first evaluate the tracking performance of SynMotion for 19 skeleton points whose positions are

Skeleton Points	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	Average
Axis X	1.9	2.0	2.0	2.2	2.8	2.2	2.5	2.6	5.3	2.2	2.5	3.5	5.3	2.0	2.6	2.6	2.1	4.3	3.4	2.9
Axis Y	1.4	1.4	1.6	1.5	1.8	1.5	1.8	4.1	4.5	1.6	2.3	4.7	5.9	1.4	0.6	0.6	1.4	2.1	2.5	2.5
Axis Z	2.7	2.7	2.9	3.2	3.5	3.1	3.1	3.7	4.8	3.2	3.3	4.4	4.2	2.6	2.7	2.7	2.7	2.6	2.9	3.2
Overall	4.5	4.6	4.8	5.2	6.0	5.1	5.4	7.0	8.8	5.2	5.8	7.9	9.1	4.6	4.3	4.3	4.6	6.5	6.4	5.8

Table 1: Tracking errors (unit: cm) of 19 skeleton points in SynMotion along X, Y and Z axes, whose positions are shown in Figure 13(a). The overall error is the Euclidean distance between the estimated skeleton point and its ground truth in 3D space.

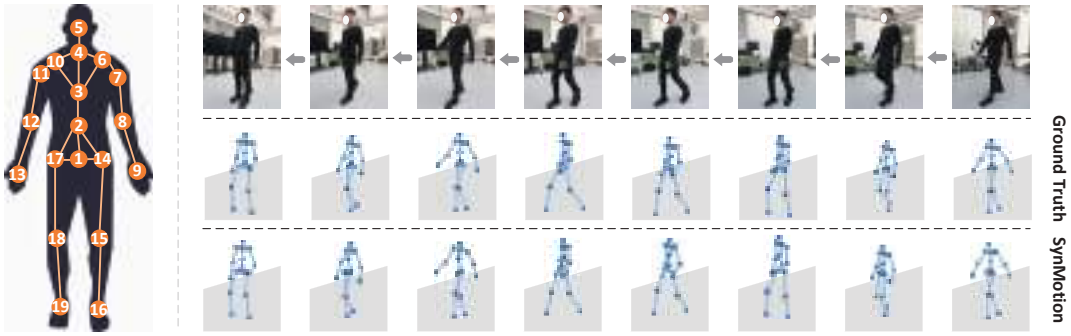

Figure 13: (a) Positions of 19 skeleton points tracked by SynMotion. (b) Reconstructed body skeleton points (the third row) as one user walks compared to the ground truth (the second row).

given in Figure 13(a). Table 1 shows the tracking errors along the three axes. The overall error is the Euclidean distance between the estimated skeleton point and its ground truth in 3D space, which is larger than the error along each individual axis [47]. It can be seen from Table 1 that for the 10 new users from Group-b), the average tracking errors of SynMotion along three axes are 2.9 cm, 2.5 cm and 3.2 cm respectively, and the overall tracking error of 5.8 cm is also small. Figure 13 illustrates the tracking performance of SynMotion as one user walks. We can see that the body pose reconstructed by SynMotion is close to the ground truth.

In Figure 12, we further plot the error distribution for each skeleton point and observe that the overall error is within 5–6 cm for most skeleton points, while the errors for skeleton points 8, 9, 12 and 13 are slightly larger (7.0 cm, 8.8 cm, 7.9 cm and 9.1 cm), corresponding to each user's elbows and wrists. This problem has also been observed in previous tracking designs, like [23], possibly due to subtle movements of the arms, but their reflection areas for sensing signals are relatively small. We will further improve network sensitivity for recognizing these four points in future work.

Performance comparison. Next, we compare the performance of different skeleton tracking methods. All users from Group-b) are *unseen* users for SynMotion. We develop the following three versions of RF-Pose3D for a more comprehensive evaluation:

- **RF-Pose3D-Ori**: This version follows the original setting from [61]. We split all the data from Group-b) into two subsets, including training (75%) and testing (25%).
- **RF-Pose3D-Tun**: We train the tracker using real mmWaves (and their coordinate labels) from Group-a) and further fine-tune it using one round of sensing data from each user in Group-b). Different from SynMotion, the skeleton point coordinates to fine-tune RF-Pose3D-Tun is from OptiTrack.

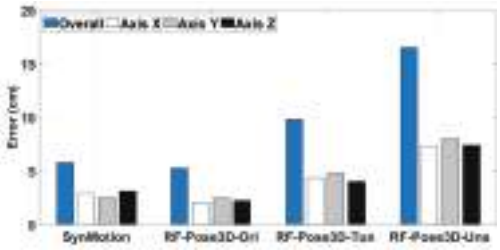

Figure 14: Performance comparison of different methods.

- **RF-Pose3D-Uns**: We use real mmWaves (and labels) from Group-a) to train the tracker and test it directly on Group-b).

We can see from Figure 14 that with the real sensing data and the corresponding label of skeleton point coordinates from real users, "RF-Pose3D-Ori" has the best performance (5.3 cm), but the cost to collect sufficient labeled training data is high. When "RF-Pose3D-Tun" is only fine-tuned, it does not fully adapt to the unseen users in this experiment, and the tracking error increases to 4.0–4.8 cm along three axes with the overall error of 9.8 cm. If "RF-Pose3D-Uns" works on the unseen users directly without fine-tuning, the tracking error is large, *i.e.*, 7.3–8.0 cm along three axes with the overall error of 16.5 cm. For comparison, SynMotion can reduce the tracking error of each axis to 2.5–2.9 cm, achieving 21.0–48.0% performance improvement compared to "RF-Pose3D-Tun" that has the most similar setting to SynMotion. The improvement mainly comes from the network structure design with LSTMs in SynMotion to better capture temporal features from sensing signals.

Ablation study. To thoroughly understand the performance of SynMotion and the effectiveness of our proposed technique, we conduct an ablation study in this experiment. First, for the signal synthesis part, we examine the impact of the number of reflection

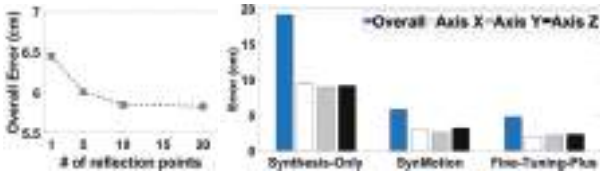

Figure 15: (a) Impact of reflection point for signal synthesis. (b) Effectiveness of the training framework design.

points adopted for each body part that is an important parameter in the human body model. In Figure 15(a), we vary this number from 1 to 20. The result shows that even if each body part is represented by one reflection point, the overall tracking error is not significantly large, *e.g.*, 6.45 cm. When the number is increased to 10, the error is less than 6 cm. If we increase it further, the error reduction is trivial. Because the computation overhead of signal synthesis increases linearly with this number, we empirically adopt 10 in the current implementation to balance accuracy and overhead.

In Figure 15(b), we further examine the effectiveness of our "synthetic-to-real" design. If the actual tracker is trained by the synthesized data only, its tracking error ("Synthesis-Only") for users from Group-b) is large, *e.g.*, 19.17 cm. Using our proposed training framework, the overall error ("SynMotion") is reduced to 5.8 cm on average. If we use the ground-truth coordinates of the fine-tuning data (instead of the pseudo coordinates estimated by the variant tracker), the overall error ("Fine-Tuning-Plus") is 4.75 cm, *i.e.*, the error reduction is insignificant. These results demonstrate the effectiveness of our proposed training framework.

4.3 Micro-Benchmarks

In this subsection, we conduct micro-benchmark experiments for a comprehensive understanding of SynMotion's performance.

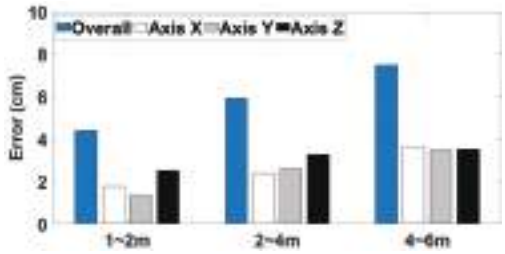

Figure 16: Performance at different working distances.

Working distance. The default working distance between the center of sensing area and the radar is about 3 m, and we change the working distance in this experiment. For ease of illustration, we divide the working distances into three groups in Figure 16. When the distance is moderate (such as 2–4 m), the per-axis error is 2.4–3.3 cm and the overall error is less than 6 cm. When the radar is closer to the sensing area (such as 1–2 m), the per-axis error is further reduced to 1.8–2.5 cm and the overall error is 4.4 cm. When the working distance is large (such as 4–6 m), the overall error increases to 7.5 cm. The results suggest that SynMotion performs well at common radar working distances (such as 1.5–3 m).

Environmental factors. In this experiment, we investigate the influence of a range of environmental factors, including surrounding furniture, occlusion (via two large monitors) and non-line of sight (NLOS, via sponge blocks), as shown in Figure 17. In this experiment, we use the actual tracker from §4.2 ("Overall Performance") before without involving these factors. We denote it as a "clean" setting. Before radars start working, the initial calibration is conducted to remove the reflections from the environment, and we use SynMotion (trained in the "clean" setting) to conduct skeleton tracking directly in these three scenarios. We find that SynMotion handles these environmental factors well. Figure 17 shows some examples of body poses reconstructed by SynMotion.

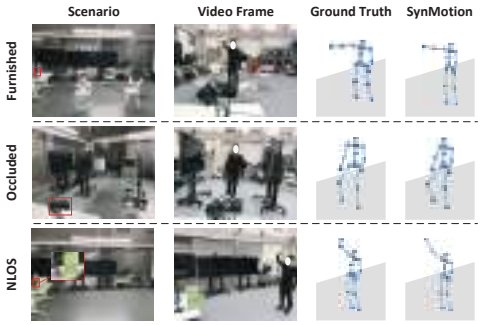

Figure 17: Examples with various environmental factors.

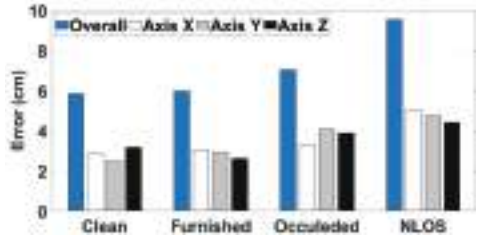

Figure 18: Performance with various environmental factors.

In Figure 18, we further plot the tracking error for each scenario. When furniture is present in the environment, we do not observe an obvious performance drop compared to the "clean" setting. When occlusion occurs, the tracking performance drops about 17% due to strong reflections from two large monitors. NLOS scenario has a greater impact on the system performance. It results in 39% increase in error compared to the "clean" setting, which is also not a recommended scenario to use radar in practice.

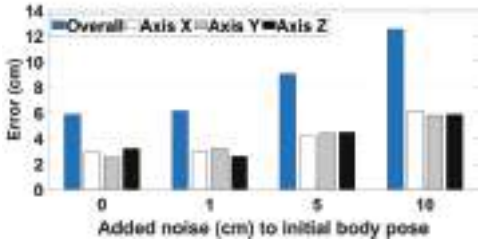

Figure 19: Impact of initial body pose errors.

Initial body-pose error. To fine-tune the variant tracker, SynMotion needs the initial body pose for each activity. In this experiment, we investigate how errors from initial body poses will affect performance. To this end, we intentionally add noise to 19 skeleton points

of the initial body pose, where the average value of the added noise varies between 1–10 cm. Figure 19 shows that when the noise is moderate (for example, within 5 cm), the increased tracking error is not significant, *e.g.*, less than 3 cm. When noise is added to the initial body pose, the direction of drift caused to each skeleton point is independent, rather than a consistent drift for all skeleton points. Therefore, the resulting tracking error does not increase linearly.

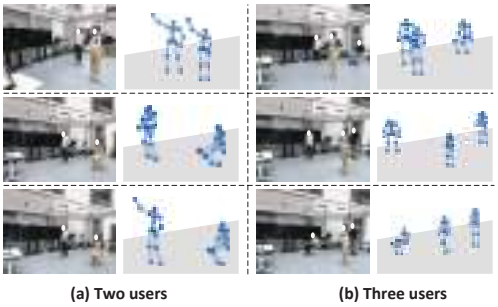

Figure 20: Examples of tracking multiple users.

Multi-users. SynMotion also has the potential to track multiple users simultaneously. To this end, we include the vision-based data for two or three concurrent users performing different activities in signal synthesis. In this experiment, different users are not very close to each other (*e.g.*, about 1–1.5 m away). The response in the heatmap exhibits several clusters and each one corresponds to one user. Figure 20 depicts examples of body poses reconstructed by SynMotion for two and three concurrent users. Compared to the one-user case, Figure 21 shows that the tracking error for multiple users is moderate, within 3.8–4.5 cm and 5.8–5.9 cm per axis with two and three concurrent users, respectively. The tracking performance degradation is mainly caused by additional reflections from users, which are not captured explicitly in the signal synthesis.

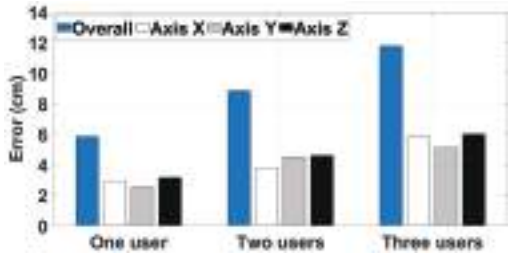

Figure 21: Performance of tracking concurrent users.

Different vision-based datasets. In this experiment, we use two public vision-based datasets, NTU RGB+D [44] and CMU MoCap [13], for signal synthesis[3], and evaluate the system performance on 10 users from Group-b), which are new users in a new environment for these two datasets. Because these two datasets track different sets of skeleton points, they need to be transformed by interpolation to obtain 19 skeleton points, which may lead to a certain transform error at first. Compared to our collected RadarSet dataset, resulting in 2.5–3.2 cm per-axis error, the tracking error along each axis based on these two public datasets increases only slightly, which is

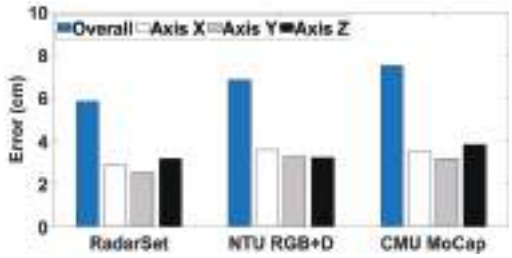

Figure 22: Performance from different vision-based datasets.

3.2–3.6 cm and 3.2–3.8 cm, respectively.

Radar facing directions. In this experiment, we investigate the effect of radar's facing direction. In particular, we synthesize signals for one virtual radar position and deploy the radar at a similar position, which serves as a reference. Then, we rotate the radar's facing direction relative to the reference from 15° to 30° to perform skeleton tracking. Figure 23 shows that when the facing direction of the radar is consistent with the reference, the tracking error of each axis is indeed small, such as 2.9–3.2 cm. When radar is rotated by 15°, the error is moderate and increases to 3.9–4.1 cm per axis. When the rotation is 30°, the error is relatively large, such as 5.2–5.6 cm per axis in our experiment. Therefore, it is suggested to deploy the radar in the facing direction as that used in the signal synthesis for good system performance.

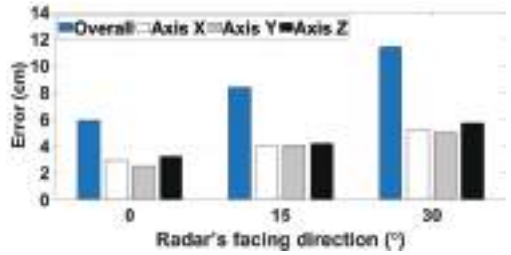

Figure 23: Impact of radar's facing direction.

New environments. In this experiment, we test SynMotion in new and unseen environments. All synthesized mmWaves still come from Groug-a) as before. In each new environment, we collect only one round of mmWave sensing signals and the initial body pose, which are fed to the variant tracker, *i.e.*, the same variant tracker used in §4.2 ("Overall Performance"), to generate the pseudo coordinates to fine-tune the actual tracker without using external cameras to annotate them. Figure 24 shows the performance. Since we do not have OptiTrack to collect the ground-truth coordinates of skeleton points in these new environments, a common issue for evaluating wireless sensing systems in new environments, we skip the numerical tracking error in this experiment. We can see from Figure 24 that the body pose estimated by SynMotion matches well with the corresponding body-pose image, which shows that using SynMotion, we can easily set up a skeleton tracking system and avoid deploying expensive cameras (such as OptiTrack) to collect and label training data.

[3]Since some activities are not included in these two public datasets, we adopt the same activities as the data we collected in this experiment, with five and four activities selected from NTU RGB+D and CMU MoCap, respectively.

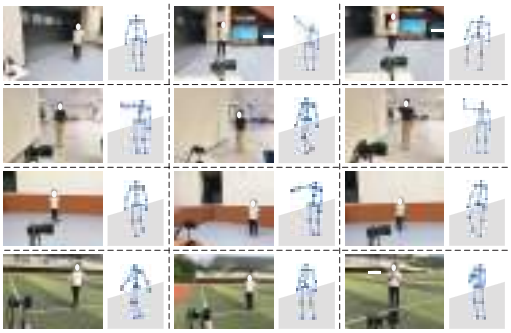

Figure 24: Examples from the new and unseen environments.

4.4 Activity Recognition

In this subsection, we examine the performance of activity recognition by comparing the following two methods:

- **Vid2Doppler** [4]: the state-of-the-art mmWave-based zero-shot design that employs a neural network to directly synthesize micro-Doppler spectrum for activity recognition.
- **SynMotion**: the method proposed in this paper. We convert the synthesized signal to micro-Doppler spectrum and use a simple neural network for activity recognition.[4]

For the above two methods, we use the data from Group-a) to train the activity recognizer and directly test its performance on the data from Group-b), *i.e.*, zero-shot recognition for both methods.

	1	2	3	4	5	6	7	8
Walking 1	1	0	0	0	0	0	0	0
Swinging arms 2	0	0.93	0	0.01	0	0	0.02	0.04
Lifting a hand 3	0	0	0.88	0	0.1	0	0	0.01
Sitting 4	0	0.02	0	0.7	0	0	0.2	0.07
Swinging hands 5	0	0	0.06	0.01	0.93	0	0	0
Lifting a leg 6	0	0.01	0	0.01	0	0.97	0	0.02
Using a phone 7	0	0	0	0.28	0	0	0.65	0.06
Chatting 8	0	0.03	0.01	0.02	0.01	0.03	0.16	0.74

Figure 25: Confusion matrix of Vid2Doppler.

Figure 25 shows the confusion matrix of Vid2Doppler. The recognition accuracy for the eight activities ranges from 65% to 100%, with an average accuracy of 84.9%. Vid2Doppler basically treats each micro-Doppler spectrum as a 2D image and employs deep learning to generate synthesized ones. From the results, we find that due to the complexity of the spectrum, it is difficult to achieve good synthesis with this end-to-end approach, which in turn limits the recognition accuracy.

	1	2	3	4	5	6	7	8
Walking 1	1	0	0	0	0	0	0	0
Swinging arms 2	0	0.96	0	0	0	0	0.01	0.02
Lifting a hand 3	0	0	1	0	0	0	0	0
Sitting 4	0	0	0	0.85	0	0	0.14	0.01
Swinging hands 5	0	0	0	0	1	0	0	0
Lifting a leg 6	0	0	0	0	0	1	0	0
Using a phone 7	0	0	0	0.09	0	0	0.88	0.03
Chatting 8	0	0	0	0.02	0	0	0.12	0.86

Figure 26: Confusion matrix of SynMotion.

[4]The network takes micro-Doppler spectrum as input, which is then processed by three CNN layers, one LSTM and one MLP layer to output the classification result.

In SynMotion, we delve into the signal itself, providing clear interpretability for mmWave-based sensing. Therefore, we can obtain better micro-Doppler spectrum from our synthesized mmWaves, thereby achieving the higher recognition accuracy. Figure 26 shows that the accuracy of recognizing these eight activities ranges from 85% to 100%, with an average accuracy of 94.1%.

4.5 System Overhead

Finally, we understand SynMotion's overhead, including the computation of signal synthesis and overhead of neural network inference.

Computation of signal synthesis. Figure 27 summarizes the detailed computational overhead of different stages involved in signal synthesis. In particular, for a 30-second motion trace, it takes about 8 seconds to complete the computation of 16 Tx-Rx on a desktop with an AMD 3700X CPU. Since synthesis is an **one-time** effort before training, this (offline) computational cost is acceptable, and can be further shortened by using multi-threaded parallel execution.

	Operations	Delay (s)
Pre-proc.	Load Data	1.015 (13.02%)
	Paras & Interpolations	0.031 (3.97‰)
Signal Synthesis	Distances & Angles of All Reflection Points	0.371 (4.76%)
	Computing All RCSs	2.268 (29.10%)
	Synthesizing All Signals	4.110 (52.73%)
Total		7.795 (100%)

Figure 27: Computational overhead of signal synthesis.

Inference latency. When SynMotion works, Table 2 shows the average execution time of four main neural network components, including attention, convulational (ResNet), LSTM and MLP layers for one second of mmWave data on an NVIDIA 2080Ti GPU. We can see from Table 2 that the overall inference time is short, *i.e.*, only 2.15 ms, which is lightweight. In our current implementation, SynMotion is configured to output estimated body poses at 20 fps.

	Attention	ResNet	LSTM	MLP	Overall
Time	0.21 ms	1 ms	0.9 ms	0.04 ms	2.15 ms

Table 2: Inference time of each neural network component.

5 POINTS OF DISCUSSION

Multiple users. With our current design, system performance degrades as more users are tracked simultaneously. It is mainly caused by additional reflections from each other that are not captured explicitly in the signal synthesis. Therefore, an important future work is to explore effective signal synthesis and sensing methods for multiple users, especially when users are in close proximity.

Unseen motions. In this paper, all the motions to be tracked using mmWave signals should be covered in the selected vision-based dataset. In other words, the current SynMotion design is not positioned to track the skeleton points of the user's body under

unseen motion, and we plan to further remove or mitigate this limitation in our future work.

Other wireless sensing signals. The design of SynMotion focuses on FMCW mmWaves. Its overall signal synthesis workflow and training framework have the potential to be extended to other sensing signals, but dedicated solutions, such as signal formulation and tracker design, need to be investigated for these new sensing signals. We plan to explore this opportunity in the future.

6 RELATED WORK

Human motion sensing. Human motion sensing and analytics [53] were actively studied using wearable sensors [35, 38, 45, 46], microphones [31], pedometers [16], etc. With a rapid advancing of deep learning, vision-based methods [1, 2, 6, 21, 30] by using cameras or depth sensors can achieve even higher sensing accuracy. However, they are limited to the inherent constraints on line-of-sight views, light conditions, occlusions and privacy issues [48].

To address these issues, RF-based solutions [55] emerge recently. E-eyes [52] uses commodity Wi-Fi devices to recognize different daily activities. RF-Net [15] proposes a framework based on meta-learning for one-shot recognition of human activities, which works with Wi-Fi and other wireless signals like FMCW and impulse radio. OneFi [54] uses Wi-Fi to enable one-shot recognition for unseen activities. Octopus [12] introduces a general platform to enable human/object imaging, passive localization and vibration sensing. In addition to activity recognition, WiFi-Person [50] and WiPose [23] further achieve 2D and 3D body pose tracking using Wi-Fi. However, Wi-Fi sensing signals are sensitive to many factors from environments, such as the room layout and nearby objects [47], which may affect the tracking performance.

Another popular family of solutions employs FMCW mmWave specifically for sensing, because FMCW mmWave can achieve more accurate signal measurements, such as time of flight (ToF) [56]. For example, by using an assembled T-shaped mmWave antenna array, the authors achieve the 2D and then 3D pose tracking in [59] and [61], respectively. By using commercial radars, the authors in [27, 34, 58] enable human activity recognition. Soli [32] performs gesture sensing using mmWave radar. Other designs [28, 43] take one more step to leverage deep learning to achieve human skeletal posture reconstruction using commercial radars. Recently, RF-Avatar [60] and mmMesh [56] further enable human mesh construction from mmWave signals. mPose [47] uses domain discriminator to remove user-dependent features to track unseen users. m3Track [25] achieves an mmWave-based posture tracking for multiple users. However, these existing methods face a common scarcity issue of training data. SynMotion aims to address this important problem and facilitate mmWave-based human sensing.

mmWave synthesis. There are existing studies to synthesize the sensing signatures of radar data. One popular example is the micro-Doppler spectrum. Authors in [33] use the motion capture data to synthesize the micro-Doppler spectrum for activity classification, such as walking and running. However, as the data is relatively sparse in previous methods, studies [17, 29] further perform the synthesis by using point clouds from depth cameras. To make synthesis more similar to real ones, generative adversarial networks are then adopted with a physics-aware design in [39], augmentation in [18], video inputs in recent Vid2Doppler [4], etc. They can address the scarcity of mmWave training data for activity recognition [7, 8]. All these designs propose to synthesize the micro-Doppler signature for activity recognition merely, while SynMotion is positioned to synthesize the source of various sensing signatures to prompt the interpretability of mmWave-based sensing. With high-quality synthesized signals, we can use them to derive various sensing signatures to enable sensing services flexibly and improve the sensing performance (§4). One recent work [5] also proposes to facilitate RF sensing from vision-based datasets, which designs for Wi-Fi signals and focuses on activity recognition. In SynMotion, we synthesize mmWaves at the signal level and further address the "synthetic-to-real" problem for fine-grained skeleton tracking.

In fact, the classical ray tracing technique [57] is proposed to formulate the process from signal transmission to signal reception via reflections, and our design is a special type of ray tracing, specifically for FMCW mmWave, by considering its properties, such as linearly varying phase and IF signals, in the signal synthesis.

Learning techniques. In this paper, SynMotion is inspired by the "synthetic-to-real" strategy [11, 38] to cope with the subtle distinct between synthesized signals and real ones, which is one popular technology in the transfer learning and domain adaption fields. Recent success of synthetic-to-real design [9, 10] leverages a baseline model fine-tuned by "ImageNet" [26] and use it to guide the generalization of the real network. This motivates us to design a variant tracker and build a RadarSet dataset, which can be further enriched and contributed by our community in a crowd-sourcing manner. In addition, how to obtain the corresponding coordinates of user skeleton points for the mmWaves to enable the fine-tuning of the actual tracker has not been studied before. Therefore, we propose a new training framework in SynMotion. Finally, the zero-shot learning starts to be explored in the internet of things (IoT) field via wearable sensors [36]. In this paper, we focus on such an activity recognition design using mmWave-based sensing signals.

7 CONCLUSION

This paper presents SynMotion, a new system to address the scarcity issue of training dataset for mmWave-based human motion sensing. The key novelty of SynMotion is to employ available vision-based datasets to synthesize mmWave sensing signals that bounce off the human body. By doing this, the synthesized signals can inherit labels from vision-based datasets directly. SynMotion demonstrates the ability to generate such labeled synthesized data with high quality. To design SynMotion, we address two challenges, including the inherent complication of mmWave synthesis and the micro-level differences compared to real mmWave sensing signals. We address these challenges and develop a prototype system. Extensive experiments show performance gains achieved by SynMotion compared to the state-of-the-art mmWaved-based sensing methods.

ACKNOWLEDGEMENTS

We sincerely thank anonymous reviewers for their helpful comments to improve the quality of this paper. This work is supported by the GRF grant from Research Grants Council of Hong Kong (CityU 11213622). Corresponding authors: Zhenjiang Li and Jin Zhang.

REFERENCES

[1] Optitrack. https://optitrack.com/.
[2] Vicon. https://www.vicon.com/.
[3] F. Adib, C. Y. Hsu, H. Mao, D. Katabi, and F. Durand. Capturing the human figure through a wall. *Acm Transactions on Graphics*, 2017.
[4] K. Ahuja, Y. Jiang, M. Goel, and C. Harrison. Vid2doppler: Synthesizing doppler radar data from videos for training privacy-preserving activity recognition. In *Proceedings of ACM CHI*, 2021.
[5] H. Cai, B. Korany, C. R. Karanam, and Y. Mostofi. Teaching rf to sense without rf training measurements. *Proceedings of ACM on interactive, mobile, wearable and ubiquitous technologies*, 2020.
[6] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of IEEE CVPR*, 2017.
[7] Q. Chen, Y. Liu, F. Fioranelli, M. Ritchie, and K. Chetty. Eliminate aspect angle variations for human activity recognition using unsupervised deep adaptation network. In *Proceedings of IEEE Radar Conference*, 2019.
[8] Q. Chen, M. Ritchie, Y. Liu, K. Chetty, and K. Woodbridge. Joint fall and aspect angle recognition using fine-grained micro-doppler classification. In *Proceedings of IEEE Radar Conference*, 2017.
[9] W. Chen, Z. Yu, S. De Mello, S. Liu, J. M. Alvarez, Z. Wang, and A. Anandkumar. Contrastive syn-to-real generalization. In *Proceedings of ICLR*, 2020.
[10] W. Chen, Z. Yu, Z. Wang, and A. Anandkumar. Automated synthetic-to-real generalization. In *Proceedings of PMLR*, 2020.
[11] Y. Chen, W. Li, and L. Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *Proceedings of IEEE CVPR*, 2018.
[12] Z. Chen, T. Zheng, and J. Luo. Octopus: a practical and versatile wideband mimo sensing platform. In *Proceedings of ACM MobiCom*, 2021.
[13] CMU Graphics Lab. CMU Graphics Lab motion capture database. http://mocap.cs.cmu.edu/.
[14] J. Crispin and A. Maffett. Radar cross-section estimation for simple shapes. *Proceedings of IEEE*, 1965.
[15] S. Ding, Z. Chen, T. Zheng, and J. Luo. Rf-net: A unified meta-learning framework for rf-enabled one-shot human activity recognition. In *Proceedings of ACM SenSys*, 2020.
[16] M. Ermes, J. Parkka, and L. Cluitmans. Advancing from offline to online activity recognition with wearable sensors. In *Proceedings of IEEE EMBC*, 2008.
[17] B. Erol and S. Z. Gurbuz. A kinect-based human micro-doppler simulator. *IEEE Aerospace and Electronic Systems Magazine*, 2015.
[18] B. Erol, S. Z. Gurbuz, and M. G. Amin. Gan-based synthetic radar micro-doppler augmentations for improved human activity recognition. In *Proceedings of IEEE Radar Conference*, 2019.
[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of IEEE CVPR*, 2016.
[20] P. J. Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*. Springer, 1992.
[21] N. Hussein, E. Gavves, and A. W. Smeulders. Timeception for complex action recognition. In *Proceedings of IEEE/CVF CVPR*, 2019.
[22] C. Iovescu and S. Rao. The fundamentals of millimeter wave sensors. *Texas Instruments*, 2017.
[23] W. Jiang, H. Xue, C. Miao, S. Wang, S. Lin, C. Tian, S. Murali, H. Hu, Z. Sun, and L. Su. Towards 3d human pose construction using wifi. In *Proceedings of ACM MobiCom*, 2020.
[24] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *Computer Science*, 2014.
[25] H. Kong, X. Xu, J. Yu, Q. Chen, C. Ma, Y. Chen, Y.-C. Chen, and L. Kong. m3track: mmwave-based multi-user 3d posture tracking. In *Proceedings of ACM MobiSys*, 2022.
[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 2012.
[27] S. M. Kwon, S. Yang, J. Liu, X. Yang, W. Saleh, S. Patel, C. Mathews, and Y. Chen. Hands-free human activity recognition using millimeter-wave sensors. In *Proceedings of IEEE DySPAN*, 2019.
[28] G. Li, Z. Zhang, H. Yang, J. Pan, D. Chen, and J. Zhang. Capturing human pose using mmwave radar. In *Proceedings of IEEE PerCom Workshops*, 2020.
[29] J. Li, A. Shrestha, J. Le Kernec, and F. Fioranelli. From kinect skeleton data to hand gesture recognition with radar. *IET The Journal of Engineering*, 2019.
[30] Z. Li, K. Gavrilyuk, E. Gavves, M. Jain, and C. G. Snoek. Videolstm convolves, attends and flows for action recognition. *Elsevier Computer Vision and Image Understanding*, 2018.
[31] D. Liang and E. Thomaz. Audio-based activities of daily living (adl) recognition with large-scale acoustic embeddings from online videos. *Proceedings of ACM on interactive, mobile, wearable and ubiquitous technologies*, 2019.
[32] J. Lien, N. Gillian, M. E. Karagozler, P. Amihood, C. Schwesig, E. Olson, H. Raja, and I. Poupyrev. Soli: Ubiquitous gesture sensing with millimeter wave radar. *ACM Transactions on Graphics*, 2016.
[33] Y. Lin and J. Le Kernec. Performance analysis of classification algorithms for activity recognition using micro-doppler feature. In *Proceedings of IEEE CIS*, 2017.
[34] H. Liu, Y. Wang, A. Zhou, H. He, W. Wang, K. Wang, P. Pan, Y. Lu, L. Liu, and H. Ma. Real-time arm gesture recognition in smart home scenarios via millimeter wave sensing. *Proceedings of ACM on interactive, mobile, wearable and ubiquitous technologies*, 2020.
[35] Y. Liu, Z. Li, Z. Liu, and K. Wu. Real-time arm skeleton tracking and gesture inference tolerant to missing wearable sensors. In *Proceedings of ACM MobiSys*, 2019.
[36] Y. Liu, S. Zhang, and M. Gowda. When video meets inertial sensors: zero-shot domain adaptation for finger motion analytics with inertial sensors. In *Proceedings of ACM/IEEE IoTDI*, 2021.
[37] M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of EMNLP*, 2015.
[38] X. Pan, P. Luo, J. Shi, and X. Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *Proceedings of ECCV*, 2018.
[39] M. M. Rahman, S. Z. Gurbuz, and M. G. Amin. Physics-aware design of multi-branch gan for human rf micro-doppler signature synthesis. In *Proceedings of IEEE Radar Conference*, 2021.
[40] S. S. Ram and H. Ling. Simulation of human microdopplers using computer animation data. In *Proceedings of IEEE Radar Conference*, 2008.
[41] S. Rao. Introduction to mmwave sensing: Fmcw radars. *Texas Instruments (TI) mmWave Training Series*, 2017.
[42] N. B. Reese and W. D. Bandy. *Joint range of motion and muscle length testing*. Elsevier Health Sciences, 2016.
[43] A. Sengupta, F. Jin, R. Zhang, and S. Cao. mm-pose: Real-time human skeletal posture estimation using mmwave radars and cnns. *IEEE Sensors Journal*, 2020.
[44] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of IEEE CVPR*, 2016.
[45] S. Shen, M. Gowda, and R. Roy Choudhury. Closing the gaps in inertial motion tracking. In *Proceedings of ACM MobiCom*, 2018.
[46] S. Shen, H. Wang, and R. Roy Choudhury. I am a smartwatch and i can track my user's arm. In *Proceedings of ACM MobiSys*, 2016.
[47] C. Shi, L. Lu, J. Liu, Y. Wang, Y. Chen, and J. Yu. mpose: Environment-and subject-agnostic 3d skeleton posture reconstruction leveraging a single mmwave device. *Elsevier Smart Health*, 2022.
[48] H. Truong, S. Zhang, U. Muncuk, P. Nguyen, N. Bui, A. Nguyen, Q. Lv, K. Chowdhury, T. Dinh, and T. Vu. Capband: Battery-free successive capacitance sensing wristband for hand gesture recognition. In *Proceedings of ACM SenSys*, 2018.
[49] P. van Dorp and F. Groen. Human walking estimation with radar. *IET Radar, Sonar and Navigation*, 2003.
[50] F. Wang, S. Zhou, S. Panev, J. Han, and D. Huang. Person-in-wifi: Fine-grained person perception using wifi. In *Proceedings of IEEE/CVF ICCV*, 2019.
[51] M. Wang, F. Qiu, W. Liu, C. Qian, X. Zhou, and L. Ma. Monocular human pose and shape reconstruction using part differentiable rendering. In *Proceedings of Computer Graphics Forum*, 2020.
[52] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu. E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures. In *Proceedings of ACM SenSys*, 2014.
[53] Y. Wang, J. Shen, and Y. Zheng. Push the limit of acoustic gesture recognition. In *Proceedings of IEEE INFOCOM*, 2020.
[54] R. Xiao, J. Liu, J. Han, and K. Ren. Onefi: One-shot recognition for unseen gesture via cots wifi. In *Proceedings of ACM SenSys*, 2021.
[55] Y. Xie, J. Xiong, M. Li, and K. Jamieson. md-track: Leveraging multi-dimensionality for passive indoor wi-fi tracking. In *Proceedings of ACM MobiCom*, 2019.
[56] H. Xue, Y. Ju, C. Miao, Y. Wang, S. Wang, A. Zhang, and L. Su. mmmesh: towards 3d real-time dynamic human mesh construction using millimeter-wave. In *Proceedings of ACM MobiSys*, 2021.
[57] Z. Yun and M. F. Iskander. Ray tracing for radio propagation modeling: Principles and applications. *IEEE access*, 3:1089–1100, 2015.
[58] R. Zhang and S. Cao. Real-time human motion behavior detection via cnn using mmwave radar. *IEEE Sensors Letters*, 2018.
[59] M. Zhao, T. Li, M. Abu Alsheikh, Y. Tian, H. Zhao, A. Torralba, and D. Katabi. Through-wall human pose estimation using radio signals. In *Proceedings of IEEE CVPR*, 2018.
[60] M. Zhao, Y. Liu, A. Raghu, T. Li, H. Zhao, A. Torralba, and D. Katabi. Through-wall human mesh recovery using radio signals. In *Proceedings of IEEE/CVF ICCV*, 2019.
[61] M. Zhao, Y. Tian, H. Zhao, M. A. Alsheikh, T. Li, R. Hristov, Z. Kabelac, D. Katabi, and A. Torralba. Rf-based 3d skeletons. In *Proceedings of ACM SIGCOMM*, 2018.