Real-time Continuous Activity Recognition with a Commercial mmWave Radar

Yunhao Liu, *Fellow, IEEE,* Jia Zhang, *Student Member, IEEE,*
Yande Chen, *Student Member, IEEE,* Weiguo Wang, *Student Member, IEEE,*
Songzhou Yang, *Student Member, IEEE,* Xin Na, *Student Member, IEEE,*
Yimiao Sun, *Student Member, IEEE,* Yuan He, *Senior Member, IEEE*

Abstract—mmWave-based activity recognition technology has attracted widespread attention as it provides the ability of device-free, ubiquitous and accurate sensing. Recognition of human activities intrinsically demands to be real-time and continuous, but the state of the arts is still far limited with the capacity in this regard. The main obstacle lies in activity sequence segmentation, i.e. locating the boundaries between consecutive activities in an activity sequence. This is a daunting task, due to the unclear activity boundaries and the variable activity duration. In this paper, we propose ZuMa, the first mmWave-based approach to real-time continuous activity recognition. When resorting to a machine learning model for activity recognition, our insight is that the recognition confidence of the recognition model is highly correlated to the accuracy of activity sequence segmentation, so that the former can be utilized as a feedback metric to finely adjust the segmentation boundaries. Based on this insight, ZuMa is a coarse-to-fine grained approach, which includes the fast coarse-grained activity chunk extraction and the find-grained explicit segmentation adjustment and recognition. We have implemented ZuMa with the commercial mmWave radar and evaluated its performance under various settings. The results demonstrate that ZuMa achieves an average recognition error of 12.67%, which is 65.08% and 71.87% lower than that of the two baseline methods. The average recognition delay of ZuMa is only 1.86 s.

Index Terms—Wireless Sensing, Millimeter Wave, Continuous Activity Recognition

1 Introduction

Activity recognition plays a significant role in smart systems like smart home, smart health, and smart office. With efficient sensing of human activity, activity recognition becomes an increasingly important function in many smart applications, such as elderly care, fitness monitoring, interactive control, fall detection, and auxiliary rehabilitation.

A variety of technologies can provide the ability of activity recognition, including vision based [1], [2], [3], wearable based [4], [5], [6], wireless sensing based solutions [7], [8], [9], [10], [11], [12], and etc. Compared with vision based solutions, wireless sensing technology has the privacy-preserving feature and is robust in dynamic lighting conditions. The contactless feature of wireless sensing eases the burden on the user, making it more attractive than wearable based solutions.

Among the wireless sensing based approaches, millimeter wave (mmWave) based sensing appears to be a promising direction, owing to the high spatial resolution and fine-grained sensing capacity of using mmWave signals as the sensing medium. The existing works [9], [13], [14], [15], [16], [17] usually propose to use the mmWave radar(s) to collect the activity-associated mmWave signals reflected from the human body, and recognize the activity by processing and analyzing those signals. In this way, the existing proposals can recognize a single-shot activity with satisfactory accuracy.

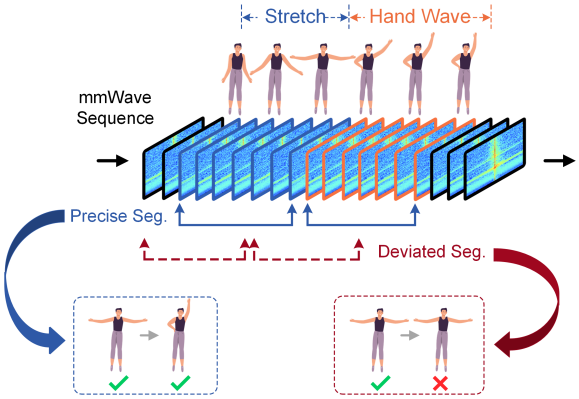


Figure 1: Segmentation is critical for continuous activity recognition.

Playing a pivotal role in interactive smart applications, activity recognition intrinsically demands to be real-time and continuous, since human activities are continuous in most cases and need to be responded in a timely manner. This fact is however often overlooked in the existing studies. None of the existing works can satisfy the above demand. The reason is that accurate and fast activity sequence segmentation is critical for continuous activity recognition but has not been well resolved. As shown in Fig. 1, the activity-associated mmWave segments are extracted from the entire

- *Y. Liu is with the Automation Department and Dean of the GIX, Tsinghua University, Beijing, China. E-mail: yunhao@greenorbs.com.*
- *J. Zhang, Y. Chen, W. Wang, S. Yang, X. Na, Y. Sun, Y. He are with the School of Software and BNRist, Tsinghua University, Beijing, China. E-mail: {j-zhang19, cyd22, wwg18, yangsz18, nx20, sym21}@mails.tsinghua.edu.cn, heyuan@mail.tsinghua.edu.cn.*

(Corresponding author: Yunhao Liu and Yuan He.)

sequence as the recognition input, while the accuracy of the segmentation largely determines the accuracy of final recognition. A deviated segmentation may even lead to a cascade of recognition errors.

Some of the existing works have identified this problem but they generally over-simplify it. Some of them capture the signal segments through fixed-length sliding windows [9], [15], which cannot be adapted for variable-duration activity recognition. Some others extract the signal segments corresponding to a single activity from a signal sequence, by tracing the change in simple low-level features, such as the velocity [18] and the number of point clouds [13], which is ineffective for consecutive activity recognition. As a whole, the existing works have apparent limitations in continuous activity recognition tasks, as they cannot accurately segment variable-duration continuous activity sequences.

By looking into the above problem, we find three critical challenges there: First, the boundaries between consecutive activities are unclear and hard to be directly located in the mmWave signals. Second, the duration of individual human activities has large variations. In our experiments, the duration of a single activity varies from 1 s to 3 s. Such large duration variations further complicate the accurate segmentation of the activity sequence. Last but not least, practical application demands real-time recognition and therefore requires a fast segmentation process.

In this paper, we rethink the above problem in a different way. Instead of trying yet another metric to define the boundaries between activities, which is difficult or even impossible to find due to the complexity and diversity of activities, we believe the machine learning model itself has the potential to identify the boundaries. Specifically, our insight is that the recognition confidence of the recognition model is highly correlated to the accuracy of segmentation. The recognition confidence can be directly obtained from the logit output of the recognition model and indicates the correct recognition probability. As the recognition model is trained with the actual activity segments, the more accurately the activity boundaries are located, the more complete the activity semantic information is, and therefore the higher the corresponding recognition confidence is.

Inspired by this insight, We propose **ZUMA**[1], the first mmWave-based approach to real-time continuous activity recognition. ZUMA utilizes the recognition confidence as a feedback metric to finely locate the segmentation boundaries and enable continuous activity recognition. Specifically, we first extract the range-Doppler spectrum around the human body to capture activity-associated information. We then detect the presence of activity sequences by calculating the velocity entropy of each range-Doppler spectrum. After that, we utilize a modified Temporal Segment Network (TSN) model for single-shot activity recognition to obtain the recognition confidence. Since the recognition confidence is positively correlated to the segmentation accuracy, we transform the activity sequence segmentation and recognition problem into the maximum recognition confidence searching problem. Finally, we propose a parallel divide-and-conquer search algorithm to quickly search for the largest recognition confidence, so as to locate the activity boundaries and achieve continuous activity recognition.

Our contributions can be summarized as follows:

- We propose a novel scheme of explicit segmentation adjustment, which exploits the potential of an activity recognition model to accurately identify the boundaries between consecutive activities. To the best of our knowledge, ZUMA is the first mmWave-based approach to real-time continuous activity recognition.
- ZUMA is a tailored design that includes velocity entropy-based activity chunk extraction, single-shot activity recognition with fixed-number sampling, and continuous activity recognition based on parallel divide-and-conquer searching. The design achieves timeliness and accuracy of recognition at the same time.
- We implement ZUMA on the commercial device (TI IWR6843ISKODS) and conduct extensive experiments. The results demonstrate that ZUMA achieves an average recognition error of 12.67%, which is 65.08% and 71.87% lower than that of the two baseline methods. The average recognition delay of ZUMA is only 1.86 s.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 introduces the preliminaries of our work. Section 4 elaborates on our design. The implementation and evaluation results are presented in Section 5 and Section 6, respectively. Section 7 discusses some practical issues and future directions. We conclude this work in Section 8.

2 Related Work

In this section, we first introduce the related works about mmWave-based activity recognition, including single-shot and continuous activity recognition. Then we introduce some continuous recognition works using other technologies to illustrate the unique challenges of mmWave-based recognition.

2.1 Activity Recognition via mmWave

With the development of commercial mmWave radars and the growing attention to human behavior, there have been many works [10], [13], [19], [20], [21], [22] utilizing mmWave radars to achieve single-shot activity recognition. They mainly focus on extracting features from well-segmented activities and proposing tailored recognition models to recognize a single-shot activity with satisfactory accuracy. For example, Soli [19] utilizes their customized mmWave radar to extract a variety of features from the range-Doppler spectrums of individual gestures and then utilizes different machine learning classifiers to achieve accurate gesture recognition. EI [10] employs a commercial 60GHz mmWave transceiver system to acquire Channel Impulse Response (CIR) measurements of individual activities for subsequent recognition. Some works notice the importance of accurate activity segmentation in their recognition tasks. DI-Gesture [13] adopts variable-length gesture segmentation instead of fixed-length segmentation within continuous range-angle frames. It employs a motion indicator to discern whether the current frame is a motion frame and implements a dynamic window mechanism for motion segmentation. Nonetheless,

1. ZUMA is a classic game of manipulating a frog-like creature to detonate consecutive sequences of balls of the same color.

this approach is tailored solely for single-shot activity extraction. In real-world activities, there lacks a distinct stationary point between consecutive activities.

Several studies [9], [14], [18], [23], [24]have investigated continuous activity recognition using mmWave radars. However, none of them simultaneously fulfills the two intrinsic requirements: real-time and continuous recognition capabilities. RFWash [23] proposes a sequence learning approach devoid of explicit segmentation to forecast gesture sequences from range-Doppler spectrums. Although effective, this method is time-consuming since each frame must be independently predicted. mHomeGes [9] takes the superposition results of the fixed-length denoised range-Doppler profiles as input and proposes a hidden Markov model-based voting mechanism to handle continuous gesture signals. While suitable for real-time single-shot activity recognition, it encounters challenges in continuous activity recognition due to the variable activity duration. M-Gesture [18] also proposes a real-time gesture recognition approach with a system status transition to determine the start and end point of a gesture. However, it only functions when there are stationary points between activity segments.

2.2 Continuous Recognition via Other Technologies

Numerous studies have explored leveraging various sensing mediums for continuous activity recognition, such as cameras, WiFi, wearable sensors and so on. However, most of these approaches either rely on time-consuming deep learning models or employ simple matching methods with limited accuracy. They are not well-suited for mmWave-based continuous activity recognition tasks due to the real-time requirements and the complexity of mmWave signals. For example, camera-based methods [1], [2], [3] have been widely explored to achieve spatial-temporal activity detection and recognition. Many machine learning techniques are used to achieve accurate region proposal extraction and recognition. Nevertheless, such methods are not applicable to mmWave-based real-time recognition tasks due to the limited imaging resolution of mmWave radars and the excessive computational overhead. Some of them [2] propose incorporating segmentation bias into the loss function to enable the recognition model to identify appropriate segments for recognition. However, this necessitates traversing the entire data sequence multiple times, resulting in significant time consumption. Moreover, such methods lack generality for variable recognition models as they require modifying the loss functions. WiFi-based methods [25], [26], [27], [28], [29], [30] utilize CSI measurements for activity recognition, often employing low-level feature matching or sequence template matching for activity segment extraction. However, due to the sensitivity of mmWave signals to human activities, the suitable matching templates are hard to select, rendering simple matching methods unsuitable for mmWave-based continuous recognition tasks. Similarly, wearable-based methods [5], [6], [31] exploit the changes in sensor data to recognize activities. They utilize template matching or learning-based matching for activity segment extraction to achieve continuous recognition. However, these methods also encounter issues of data ambiguity or huge computation overhead when applied to mmWave signals.

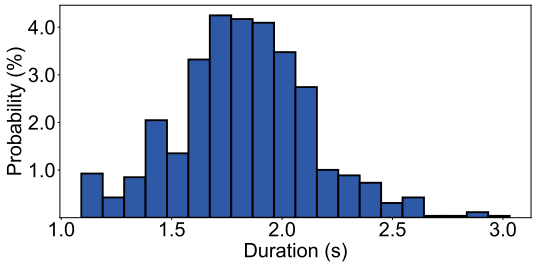


Figure 2: The distribution of the activity duration

In summary, due to the unclear boundaries between consecutive activities in mmWave signals and the variable activity duration, existing mmWave-based approaches are difficult to apply to continuous activity recognition. Besides, these methods using other technologies are unsuitable for mmWave-based continuous recognition tasks because of their high computation overhead and the complexity of the mmWave signals. On the contrary, we propose an explicit segmentation adjustment method to quickly locate the activity boundaries and enable continuous activity recognition. This method is not reliant on specific recognition models and has great generality.

3 Preliminary

In this section, we first demonstrate the variability in activity duration, which is a critical challenge for continuous activity recognition. Then we focus on our preliminary studies on the correlation between the segmentation accuracy and the recognition outputs.

3.1 The Variability in Activity Duration

We first collect and construct an extensive mmWave activity dataset, which is described in detail in Section 5. This dataset consists of about 6000 samples of twelve predefined typical activities. Each sample is obtained by manual segmentation. The predefined activities encompass four fitness activities, four interaction activities, and four rehabilitation training activities: stretch (ST), arm curl (AC), squat (SQ), boxing (BO), handclap (HC), hand waving (HW), hand crossed (HC), pull down (PD), breast expansion (BE), right stretch (RS), waist twist (WT) and stretch down (SD).

We measure the durations of these activity samples, and the distribution of the activity duration is shown in Fig. 2. We can find that the durations of these activities vary significantly from 1 s to 3 s. The reason is that different individuals (e.g., having different weights, heights or genders) and even the same individual at various times (e.g., having different fatigue and emotional states) can have disparate activity durations. Such a large variation is extremely challenging for continuous activity recognition. First off, the large duration variance makes it difficult for these segmentation methods based on fixed-length sliding windows to be applied. Furthermore, varying activity durations exacerbate the difficulty of locating activity boundaries.

3.2 The Correlation between Segmentation Accuracy and Recognition Outputs

In this section, we present our preliminary studies on the correlation between the segmentation accuracy and two

types of recognition outputs, the recognition accuracy and the recognition confidence, respectively.

We take a TSN recognition model [32] as an example to demonstrate the correlation between the segmentation accuracy and the recognition outputs. The TSN model can recognize a single activity of arbitrary length and is detailly described in Section 4.4.1.

To explore the correlation between the segmentation accuracy and the recognition outputs, we first build the test datasets corresponding to different segmentation biases based on the manual-segmented samples in the activity dataset. The segmentation bias refers to the difference between the actual activity boundary and the segmentation boundary, which is negatively correlated to the activity segmentation accuracy. The smaller the segmentation bias is, the higher the segmentation accuracy.

We assume that each sample in the manual-segmented dataset corresponds to a segment within the time range $[a_k, b_k]$ in the raw data. To construct the test dataset corresponding to a specific segmentation bias (s_b, e_b), we select the segment within the time range $[a_k + s_b, b_k + e_b]$. Here, a_k and b_k denote the start and end timestamps of the k-th sample, respectively, while s_b and e_b represent the start and end timestamp biases. We specifically set s_b and e_b to vary at intervals of 0.033 s within the range of $[-0.4, 0.4]$ seconds, respectively. The interval of 0.033 s corresponds to the segment duration for a single range-Doppler spectrum under the radar configuration. In this way, we can build the test datasets corresponding to different segmentation biases. Then the samples from these test datasets are input into the recognition model to obtain the recognition outputs.

Segmentation accuracy v.s. recognition accuracy. The average recognition accuracies of these test datasets are shown in Fig. 3. Each value in the plot represents the average recognition accuracy of all individual activities in the corresponding dataset. The results show that the recognition accuracy is positively correlated to the segmentation accuracy. When there is no segmentation bias (i.e., the manual-segmented dataset), which corresponds to the highest segmentation accuracy, the recognition accuracy is the highest and reaches 97.84%. With an increase in the start timestamp bias or the end timestamp bias, the segmentation accuracy becomes lower and the recognition accuracy decreases monotonically. The reason is that as the segmentation bias increases, the activity semantic information becomes incomplete and noisy, thereby causing misunderstandings by the recognition model regarding the activity segment.

While the recognition accuracy is indeed positively correlated with segmentation accuracy, it serves as an outcome measure rather than a predictive indicator. Given that the recognition accuracy is the ultimate target of our recognition system, it cannot be used to infer segmentation accuracy beforehand. To address this, we require an indicator directly derived from the recognition model that can aid in inferring segmentation accuracy.

Segmentation accuracy v.s. recognition confidence. We find that the logit output of the recognition model [33], [34] is a suitable indicator for inferring segmentation accuracy. In typical recognition models, the logits represent unnormalized predictions and are often processed by the *Softmax* module to generate final recognition probabilities.

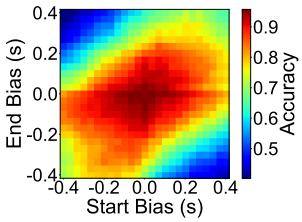


Figure 3: Segmentation accuracy *v.s.* recognition accuracy

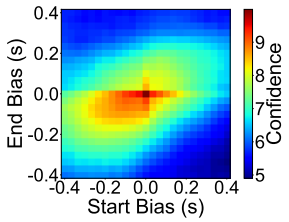


Figure 4: Segmentation accuracy *v.s.* recognition confidence

Unlike recognition accuracy, the logits corresponding to each class can be directly obtained from the recognition model. we further observe a significant correlation between the logit corresponding to the correct recognition class and the segmentation accuracy. Inspired by this observation, we exploit this unique pattern to locate the activity boundaries and determine the correct recognition class. Specifically, we take the logit corresponding to the correct recognition class in the recognition output as the recognition confidence.

We evaluate the correlation between the segmentation accuracy and the recognition confidence. The average recognition confidence of these test datasets is shown in Fig. 4. We observe that in the absence of segmentation bias, the average recognition confidence is highest. As the segmentation bias increases, the average recognition confidence experiences a rapid decline. We carefully analyze the underlying mechanisms of the relationship between segmentation accuracy and recognition confidence. When the recognition model is trained to withstand the environment noise and accurately recognize discrete activities, the completeness of activity segments emerges as the primary factor influencing continuous recognition outputs. A larger segmentation bias leads to more incomplete activity segments and noisier semantic information, subsequently leading to lower recognition probability and recognition confidence.

These experimental results show that it is feasible to utilize the recognition confidence to infer the segmentation accuracy. Consequently, it can serve as a suitable feedback metric for explicit segmentation adjustment. Note that the recognition confidence is part of the logit output and therefore can be directly obtained from the recognition model. However, obtaining the recognition confidence corresponding to all segmentation biases is highly time-consuming. Furthermore, the correct recognition classes are not available in advance, which makes the calculation of the recognition confidence more difficult. We solve these problems in the following section.

4 DESIGN

In this section, we first provide the overview of ZuMa and then introduce its three key modules.

4.1 Overview

The overview of ZuMa is shown in Fig. 5. ZuMa adopts a coarse-to-fine grained approach to achieve real-time continuous activity recognition. After signal preprocessing, ZuMa first quickly locates the activity sequences in the entire

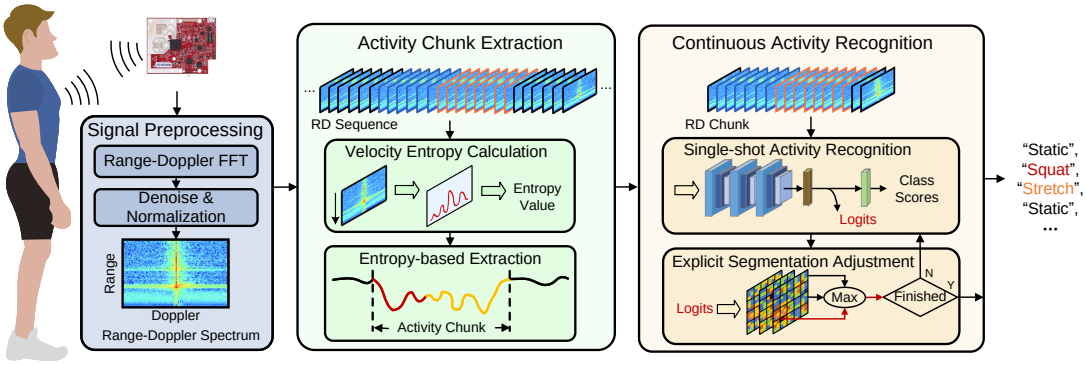


Figure 5: The overview of ZuMa

mmWave sequence to reduce computation overhead. It employs the velocity entropy as an indicator for fast coarse-grained activity chunk extraction. Subsequently, ZuMa iteratively adjusts the segmentation of each activity within the activity chunk to achieve accurate continuous activity recognition. In each adjustment, ZuMa utilizes the recognition confidence corresponding to the current segments as the feedback metric for further fine-grained segmentation adjustment. Specifically, ZuMa consists of three modules to realize this coarse-to-fine approach: signal preprocessing, activity chunk extraction, and continuous activity recognition. We introduce each module below:

• *Signal preprocessing.* ZuMa first utilizes the mmWave radar to scan the environment to obtain the range-Doppler spectrum around the human body. After localizing the position of the human body, ZuMa applies the spectrum denoising technique and normalization to the human-around range-Doppler spectrums to construct the range-Doppler spectrum sequence for further recognition.

• *Activity chunk extraction.* For each range-Doppler spectrum, ZuMa calculates its velocity entropy value to obtain the velocity entropy sequence. As the velocity entropy undergoes significant changes in the presence of activity but remains relatively constant when the human is stationary, ZuMa locates the start and end frames of the activity sequences by comparing the velocity entropy variance within a sliding window with a predefined threshold, and then extracts activity chunks.

• *Continuous activity recognition.* To obtain the recognition confidence of the segments from the activity chunk, ZuMa employs a modified TSN model for quick single-shot activity recognition. Since the segmentation accuracy is positively correlated to the recognition confidence, ZuMa transforms the activity sequence segmentation and recognition problem into the maximum recognition confidence searching problem. With the recognition confidence corresponding to the current segments serving as the feedback metric, ZuMa iteratively adjusts the segmentation of each activity and quickly locates the largest recognition confidence. Specifically, ZuMa employs a novel parallel divide-and-conquer search algorithm to expedite the process of locating the largest recognition confidence.

4.2 Signal Preprocessing

ZuMa first scans the environment surrounding the human body using the mmWave radar and captures the reflected signal. The reflected signal is mixed with the transmitted signal to obtain the intermediate frequency (IF) signal. Then ZuMa applies the classic range-FFT and Doppler-FFT operations [35] to the IF signal to obtain the entire range-Doppler spectrum, which contains the activity-associated information.

To enhance the saliency of activity-associated information in the range-Doppler spectrum, ZuMa first eliminates the spectrum components irrelevant to the human body. We notice that the magnitude of the range bins where the human body is located undergoes periodic changes due to human micro-movements such as breathing. Conversely, the magnitude of other range bins remains stable due to environmental consistency. Leveraging this insight, we locate the maximum variance of the magnitude within a second to pinpoint the position of the human body. Considering the potential significant displacement of human body parts during activities, ZuMa reserves the range-Doppler spectrum corresponding to 100 range bins around the position of the human body as the human-around spectrum.

To further mitigate the interference of the human-around environment on the activity-associated information, we employ the spectrum subtraction algorithm [36] to suppress the environment noise in the human-around spectrum. The core idea is to subtract the estimation of the average background noise spectrum from the measured spectrum, where the average background noise spectrum can be estimated from the environment where no human is present.

After removing the environmental noise, we normalize each spectrum to ensure uniform magnitude ranges across all spectrums. This normalization step is crucial for subsequent calculations of the velocity entropy sequence and single-shot activity recognition.

4.3 Activity Chunk Extraction

With continuous signal preprocessing, we can obtain a series of denoised range-Doppler spectrums. To reduce the computation cost and facilitate further recognition, ZuMa extracts the activity chunks from the entire mmWave sequence. These activity chunks contain an arbitrary number of continuous activities. Specifically, ZuMa first calculates the velocity entropy value of each range-Doppler spectrum, then ZuMa analyzes the velocity entropy variance to detect the presence of activity chunks.

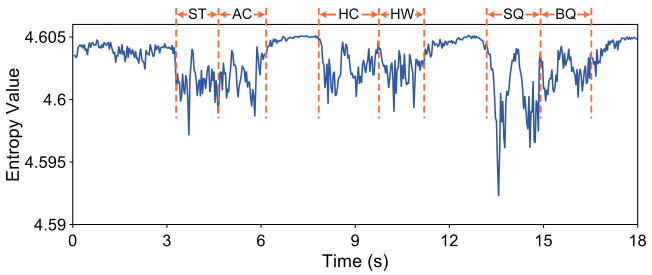


Figure 6: The velocity entropy sequence of continuous activities

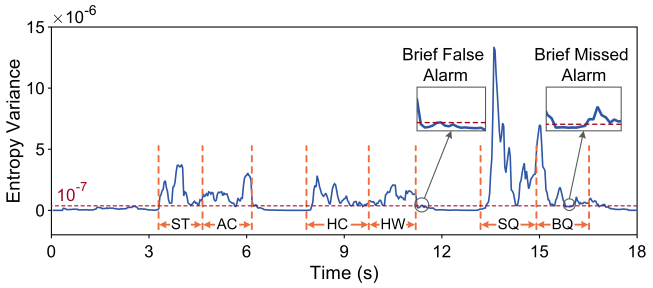


Figure 7: The velocity entropy variance sequence of continuous activities

4.3.1 *Velocity entropy calculation*

We note that the velocity distribution in the range-Doppler spectrum can serve as a unique metric to detect the presence of activities, which characterizes the intensity distribution in the Doppler spectrum. The entropy is typically used as the metric of dispersion, and we define the velocity entropy to measure the dispersion of the Doppler spectrum. When a human is stationary, the velocity of all the body parts are nearly zero, and the intensity of the Doppler spectrum tends to concentrate in the center bin, resulting in a low velocity entropy. Conversely, when an activity occurs, different body parts typically have different velocities, and the intensity of the Doppler spectrum disperses across non-center bins, resulting in a high velocity entropy. Inspired by existing works in video and wearable sensor data analysis [37], [38], [39], we calculate the velocity entropy value of the range-Doppler spectrum to characterize the velocity distribution and thereby detect the presence of activities. Specifically, we first aggregate the range-Doppler spectrum along the range dimension to obtain the corresponding column sums. Assume that the column sum v_i represents the aggregation intensity value of the i-th Doppler bin, the velocity entropy can be calculated by:

$$VE = -\sum_{i=1}^{D} v_i * \log v_i \tag{1}$$

where D represents the number of the Doppler bins.

We analyze a mmWave sequence containing multiple activities and calculate the corresponding velocity entropy sequence, illustrated in Fig. 6. In this sequence, six typical activities are executed in sequence, with each pair of activities executed continuously. The ground truth of each activity duration is clearly marked in the plot. We find that whenever an activity chunk occurs, there is a significant change in the velocity entropy sequence. This is attributable to the distinct velocity distribution during human activities compared to stationary periods. Besides, as there are few stationary intervals between consecutive activities, the velocity entropy exhibits continuous fluctuations within the activity chunks, making it challenging to precisely locate boundaries between consecutive activities using solely such a low-level feature.

4.3.2 *Entropy-based extraction*

The subsequent task involves accurately extracting the activity chunks from the entire range-Doppler spectrum sequence. Although the velocity entropy changes significantly when the activity occurs, directly applying a velocity entropy threshold often leads to high rates of false alarms and missed alarms due to its high variance. To address this issue, we calculate the short-term variance of the velocity entropy using a sliding window of length 10 frames. Unlike directly applying a velocity entropy threshold, the velocity entropy variance maintains a high value during activities and is resistant to the transient noise, which frequently appears in the velocity entropy sequence.

The velocity entropy variance sequence corresponding to the aforementioned mmWave sequence is shown in Fig. 7. In ZUMA, we utilize an empirical threshold of 10^{-7} to detect the presence of activities. We can find that all of the activity chunks in the mmWave sequence are successfully detected with limited start and end biases. The threshold applies to various locations and orientations of the person, assuming that the user faces the device within a moderate range of orientation angles. As Eq.1 in Section 4.3.1 shows, the velocity entropy is determined by the quantity and reflector signal strength of body parts with different velocities, and it is irrelevant to the velocity values. Therefore, as a moderate orientation angle deviation changes the radial velocities while keeping the body parts in different Doppler bins, the velocity entropy is approximately invariant with the orientation.

In the extraction process, some short-term static segments may exhibit velocity entropy variances higher than the threshold, while certain activity segments may have velocity entropy variances lower than the threshold. These instances lead to wrong chunk extraction, referred to as brief false alarms and brief missed alarms, respectively. To mitigate brief false alarms, we compare the durations of these segments with the minimum activity duration (i.e., 1 s in our preliminary study). If these segments are shorter than the minimum activity duration, they are flagged as false alarms and can be neglected. For brief missed alarms, we utilize a duration threshold of 10 frames to determine whether it represents a continuous activity chunk or two separate activity chunks.

4.4 Continuous Activity Recognition

The extracted activity chunks contain an arbitrary number of continuous activities, necessitating accurate segmentation and sequential recognition. To accomplish this task, we first select a predetermined number of overlapping segments from the start of the activity chunk. These segments are stacked together and fed into a single-shot activity recognition model to obtain the corresponding recognition confidence. Based on the distribution of the recognition confidence, ZUMA iteratively refine the segment selection

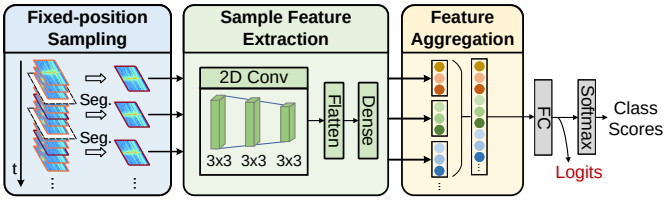


Figure 8: The single-shot activity recognition model structure

process for the next round of adjustment. Through multiple iterative adjustments, ZUMA can locate the maximum recognition confidence, thereby determining the boundaries of the first activity and obtaining the corresponding recognition results. ZUMA then repeats this process for the remainder of the activity chunk, resulting in continuous activity recognition.

4.4.1 Single-shot activity recognition

Considering the variability in activity duration and the requirement for real-time continuous recognition capability, a straightforward recognition model capable of accommodating variable activity durations is essential. In ZUMA, we leverage a modified Temporal Segment Network (TSN) model [32] to fulfill these needs. The TSN model operates by extracting a fixed number of frames from the activity segment of arbitrary length and subsequently extracting features from these frames, enabling straightforward and efficient single-shot activity recognition.

To improve the sensitivity of the TSN model to the segmentation, we modify the sampling method to fixed-position sampling. The modified TSN model structure is shown in Fig. 8. Specifically, we divide the activity segment into ten equal parts, one third of the number of frames in one second. Based on the observation that adjacent frames are quite similar and make activity segmentation harder, we set the number of parts as 10 to avoid adjacent frames to be selected in a typical activity that lasts no less than 1 second (30 frames). Subsequently, we select the first frame from the beginning part, the last frame from the end part, and the middle frame from each of the remaining parts as the selected frames. This fixed-position sampling method ensures that the trained model is more sensitive to segmentation and yields higher recognition confidence when the activity is accurately segmented.

Then each selected frame passes through the sample feature extraction module to obtain the corresponding features. They are first processed via three convolutional layers with a 3×3 range-Doppler kernel, 2 strides, and 1 padding, each of which follows a batch normalization layer and a ReLu layer. The channel numbers of the three Convolutional layers progressively increase from 8 to 16 to 32. After that, a flatten layer and a dense layer with 128 units are employed to obtain the feature vector.

These features are further concatenated and fed into a fully connected module to obtain the recognition results. The fully connected module comprises two layers: the first layer consists of 32 units, followed by a second layer with 7 units. Its output is exactly the logits corresponding to each class. Finally, a Softmax layer is applied to obtain the final recognition scores.

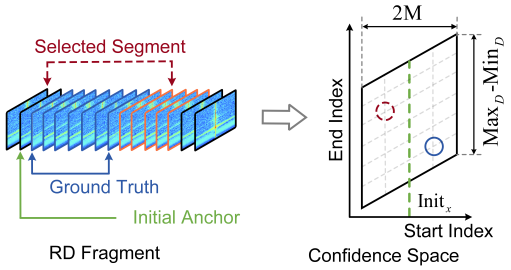


Figure 9: The continuous activity recognition problem is transformed into maximum recognition confidence searching.

4.4.2 Explicit segmentation adjustment

As we mentioned before, the recognition confidence (i.e., the logit corresponding to the correct class) increases as the segmentation bias decreases. We exploit this property to transform the continuous activity sequence segmentation and recognition problem into the maximum recognition confidence searching problem. The problem formulation is shown in Fig. 9. Each selected segment corresponds to a sample in the recognition confidence space. Its x-coordinate and y-coordinate represent the start frame and the end frame respectively, and its value represents the corresponding recognition confidence of the selected segment. In this way, the activity boundaries can be located by searching the maximum value in the recognition confidence space.

We further determine the search space based on the distribution of activity duration and the coarse-grained initial start frame obtained from activity chunk extraction. Specifically, assuming the activity duration varies within the range of $[Min_D, Max_D]$ frames and the initial start frame is denoted as $Init_x$, we select the initial start frame as the initial anchor to locate the search space. Since the initial start frame may not precisely align with the actual activity boundary, we select M frames before and after the initial start frame to encompass the potential activity boundary. We define the number of search frames around the initial start frame M as the search margin. Considering the constraints of activity duration, the search space is calculated as follows:

$$\begin{cases} Init_x - M \le x \le Init_x + M \\ Min_D \le y - x \le Max_D \end{cases} \tag{2}$$

where x and y represent the start frame and the end frame of the selected segment, respectively. In ZUMA, the search margin M is set to 10 according to our experiment results, which is detailed evaluated in Section 6.3.7.

In this way, the continuous activity sequence segmentation and recognition problem can be transformed into the problem of searching for the largest recognition confidence in such a diamond search space. However, obtaining the recognition confidence corresponding to each sample in the diamond search space can only be done after inputting the corresponding segments into the recognition model. This process is time-consuming and impractical to perform for all samples in the search space. Additionally, the recognition confidence relies on knowing the correct recognition class, which cannot be determined in advance. This limitation prevents us from selecting the logit space corresponding to the correct recognition class as the recognition confidence space to locate the activity boundaries.

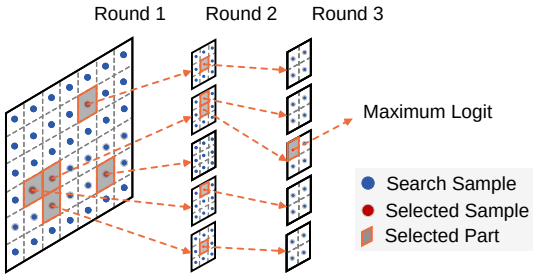


Figure 10: The process of divide-and-conquer search

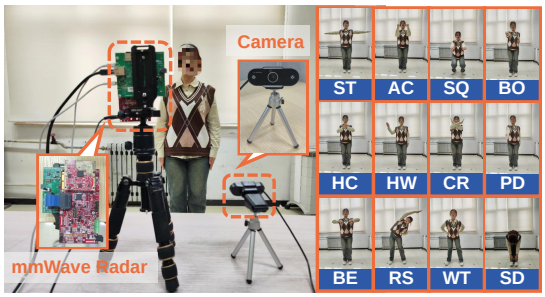


Figure 11: The experiment scenairo

To handle these challenges, we design a novel parallel divide-and-conquer search algorithm to accurately search the maximum recognition confidence with minimal search iterations. Our core observation is that in the recognition confidence space, the closer the selected segment is to the actual activity segment, the higher its corresponding recognition confidence. Additionally, the maximum value across the logit spaces corresponding to all classes is usually the recognition confidence of the actual activity segment and is surrounded by relatively large logits. Inspired by this observation, we parallelly search for the maximum logits in the logit space corresponding to all classes, and utilize the maximum value among these logits as the maximum recognition confidence to locate the activity boundary and obtain the recognition result.

For each logit space. we employ the divide-and-conquer search algorithm to quickly locate the maximum logit, The search process is shown in Fig. 10. Specifically, we first divide the search space into multiple parts evenly and take the center sample in each part as the search sample to obtain the corresponding logits. It is worth noting that these samples are stacked into a batch and fed into the recognition model simultaneously to get the logits, which significantly accelerates the process compared to processing one sample at a time. Then we select the parts corresponding to the top T largest samples from the obtained logits as the new search space for the next round of searching. In this way, the part containing the largest logit can always be selected and the largest logit can be quickly located. Sometimes the local maximum values of the logit corresponding to incorrect segmentations may be temporarily selected. However, as the iterative search progresses across multiple selected parts, these local maximum values are discarded when the global maximum logit is located, which corresponds to the actual activity segment.

Furthermore, we notice that the majority of the logits corresponding to the correct class in the search space are larger than those of the other classes. Since we can obtain the logits corresponding to all classes of these search samples, we select the five classes with the highest average logits in the first round as the candidate classes for subsequent searches. The subsequent searches of these classes are conducted in parallel to obtain the largest logit corresponding to each class. Finally, the largest logit among these classes is selected, which corresponds to the correct segmentation boundary and recognition result.

In rare cases that the logits corresponding to a wrong class is larger than the correct class, the classification and segmentation are both unreliable. However, the error is unlikely to interfere the recognition of the following activities. As discussed in Section 4.4.2, the searching mechanics is likely to guarantee the correctness of the next segmentation and avoid an accumulated error. Even if centered at a wrong frame, the search space still possibly includes the right segmentation for the next activity.

With such a search algorithm, ZuMa can accurately segment and recognize the first activity in the activity chunk. After that, the first frame after the recognized activity is utilized as the new initial anchor for the next activity recognition. Considering that the search space of the maximum recognition confidence contains a certain number of frames around the initial anchor, ZuMa can effectively prevent the occurrence of cascade recognition errors. In addition, we carefully choose the parameters to reduce unnecessary logit calculation, thereby reducing the computation overhead. Since the total number of logit calculations is determined by the number of selected samples, the number of search rounds, and the size of the part selected in each round, we minimize the values of these parameters while maintaining recognition accuracy. Specifically, the number of selected samples T is set to 5, which is enough to obtain the part where the largest logit is located. The number of search rounds is set to 3, and the size of the part selected in each round is set to 5×5, 2×2, and 1×1, respectively.

5 Implementation

In this section, we introduce the implementation of ZuMa.

Configuration. We implement ZuMa based on a commercial mmWave radar Texas Instruments IWR6843ISKODS [40]. 3 Tx antennas and 2×2 Rx antennas are built on the radar board. In our implementation, three Tx antennas take turns transmitting FMCW signals starting at 60.25 GHz with a bandwidth of 3.11 GHz. Each chirp consists of 512 samples and the ADC sample rate is 6250 kHz, achieving a range resolution of 4.8 cm and a maximum sensing range of 24.68 m. The chirp duration is set to 330 us and each frame includes 100 chirps, providing a Doppler resolution of 0.075 m/s and a maximum Doppler range of 7.52 m/s. This configuration adequately covers typical daily activity scenarios. The raw data from the radar is captured by a Ti DCA1000EVM board [41] and then transmitted to a computer equipped with an Intel Core i9-11900H processor for further processing. Our recognition model is trained using the Adam optimizer with a learning rate of 0.000005 and a batch size of 16. We employ the cross-entropy loss function during training, and the training epoch is set to 200 to prevent overfitting The model is implemented in PyTorch

[42] and is trained and evaluated on an NVIDIA GeForce RTX 1080ti GPU.

Experiment setup. The experiment scenario is shown in Fig. 11. The radar is placed on a table to capture the reflected signals from volunteers and a commercial camera is employed to collect the ground truth data. The mmWave signals are aligned with the ground truth data by the local timestamps. All the experiments are IRB-approved, and all data are anonymized.

Dataset. To construct our dataset, fifteen volunteers performed twelve types of typical activities within a range of 1 m to 2 m from the radar. These activities include stretch (ST), arm curl (AC), squat (SQ), boxing (BO), handclap (HC), hand waving (HW), hand crossed (HC), pull down (PD), breast expansion (BE), right stretch (RS), waist twist (WT) and stretch down (SD). The angle between the volunteer and the radar varies from 0° to 45°. To further enhance the dataset diversity, volunteers participated in multiple data collection sessions over three months, and data were collected in various experimental scenarios such as conference room, hallway, and laboratory. We collected a total of approximately 600 mmWave sequences, with a total recording duration of approximately 20000 s. Each mmWave sequence may contain multiple activity chunks, and each activity chunk consists of a varying number of consecutive activities. Among them, the number of consecutive activities within a single activity chunk varies between 1 and 10, and the duration of a single activity varies between 1 s and 3 s. In total, we collected approximately 6000 activity samples through manual segmentation. We use 70%, 20%, and 10% of these samples as the training set, validation set, and test set of the activity recognition model.

6 Evaluation

In this section, we evaluate the performance of ZuMa in practical continuous activity scenarios.

6.1 Methodology

As ZuMa is tailored for real-time continuous activity recognition, we evaluate its performance across a spectrum of continuous activity scenarios. These scenarios encompass variations in the number of distinct activities, the number of repetition activities, activity durations, distances between the volunteer and the radar, angles between the volunteer and the radar, and experiment scenes. We further evaluate the latency of each module in ZuMa to verify its real-time recognition capability.

Specifically, we use Activity Error Rate (AER) to evaluate the performance of ZuMa. AER is defined as the minimum number of insertions, deletions, and substitutions required to transform the predicted activity sequence into the ground truth activity sequence, divided by the number of activities in the ground truth. For example, If an activity sequence $[A_1, A_2, A_3]$ is recognized as $[A_1, A_3]$, the AER can be calculated as $1/3 \approx 33.3\%$. If it is recognized as $[A_1, A_4, A_2]$, the AER is $2/3 \approx 66.7\%$.

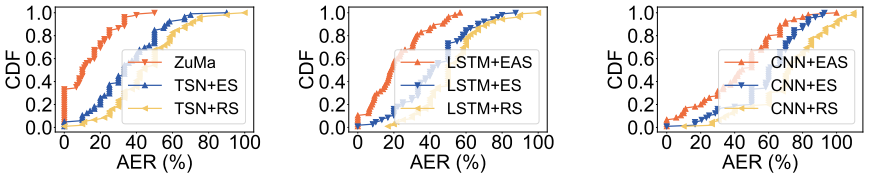
6.2 Overall Performance

We evaluate the continuous activity recognition accuracy by comparing the performance of various recognition methods. Specifically, we select three single-shot activity recognition models: TSN, CNN and LSTM (Long Short-Term Memory) [43], and three segmentation methods: EAS (Explicit Adjustment-based Segmentation), ES (Equal Segmentation) and RS (Random Segmentation), for comparison. The structure of the CNN model is the same as the feature extraction module in TSN. It averages all of the feature vectors to obtain the recognition results. In contrast, the LSTM model takes these feature vectors as sequential input and outputs the final recognition results. Among the selected segmentation methods, the ES method takes every 50 consecutive frames as a segment, which is the most common in our activity dataset. This method is similar to the methods [9], [15] based on the fixed-length sliding windows and can be used to evaluate the impact of variable activity duration. Meanwhile, the RS method randomly selects a frame number from the activity duration candidates for extraction. It's worth noting that the methods [13], [18] based on the changes in simple low-level features, such as velocity and point cloud number, are ineffective in continuous activity scenarios and are thus not included in the segmentation methods.

We separately evaluate the AER of each method. The Cumulative Distribution Functions (CDFs) of the AERs of these methods are shown in Fig. 12. Our approach achieves the lowest average AER of 12.67% and outperforms other methods. On the one hand, the results show that our EAS method significantly improves the continuous activity recognition accuracy across all three models. Taking the TSN model as an example, the average AERs of the ES method and the RS method are 36.28% and 45.04%, respectively. Our approach reduces the average AER by 65.08% and 71.87% respectively compared to these two methods. These results verify that our EAS method can be applied to various activity recognition models and significantly improve their performance in continuous activity recognition scenarios.

On the other hand, the performance of the TSN model is much better than the other two models. The average AERs of the LSTM model and the CNN model with the EAS method are 24.81% and 42.36%, respectively, much higher than our approach. The reason is that the TSN model fully preserves the segmentation sensitivity through fixed-position sampling and feature vector aggregation, while the CNN model and the LSTM model are less sensitive to activity segmentation, resulting in poor performance in continuous activity recognition tasks. Nevertheless, our EAS method effectively exploits their limited segmentation sensitivity to significantly improve their performance in continuous activity scenarios. Due to space constraints, we only present the results of the TSN model in the following sections.

We further demonstrate an example of the continuous activity recognition result of ZuMa to analyze our recognition process. The mmWave sequence contains two activity chunks and each chunk contains six distinct activities. The ground truth and the recognition result are shown in Fig. 13. ZuMa first extracts the activity chunks and then applies



(a) TSN (b) LSTM (c) CNN

Figure 12: The overall performance of different recognition models

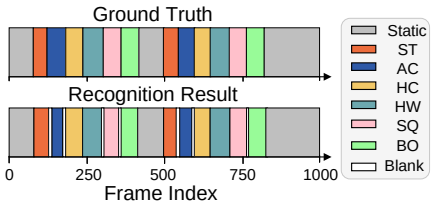


Figure 13: The example of the continuous activity recognition result of ZUMA

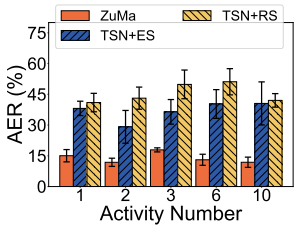


Figure 14: The impact of distinct activity number

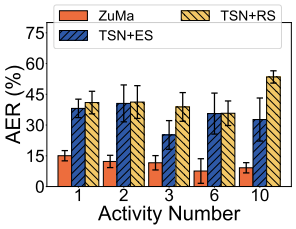


Figure 15: The impact of repetitive activity number

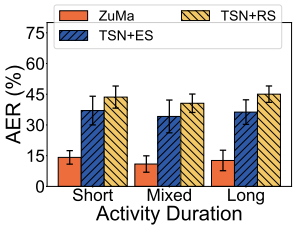


Figure 16: The impact of activity duration

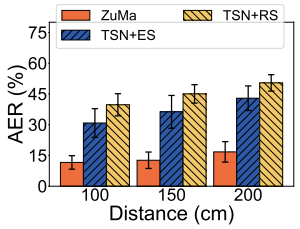


Figure 17: The impact of the distance between people and radar

explicit segmentation adjustment to each chunk. It can be found that all activities are correctly recognized and the segmentation biases are tiny. The average start bias and the average end bias are 0.093 s and 0.096 s, respectively. The recognition results further show that there are some blank segments between the recognized activities. This is because none of these activities encompass these parts during recognition. These blank segments don't affect the recognition results and can be easily distinguished.

6.3 Impact Factors

To evaluate the performance of ZUMA in continuous activity scenarios, we first apply it to the mmWave sequences that contain multiple distinct or repetitive activities within a single chunk. The mmWave sequences may contain multiple activity chunks. We vary the number of activities in a chunk to verify the performance of ZUMA under different activity sequence lengths, which are variable in actual scenarios.

6.3.1 *The number of distinct activities*

We apply ZUMA to the mmWave sequences containing consecutive 1, 2, 3, 6, and 10 activities within a chunk respectively to evaluate the impact of the distinct activity number. The adjacent activities in each chunk are distinct. The average AERs are shown in Fig. 14. We find that ZUMA can adapt to various activity numbers and achieve accurate activity recognition. The average AER of ZUMA varies from 11.83% to 17.89% with different activity numbers. The reason is that ZUMA can explicitly adjust the activity sequence segmentation to handle various activity sequence lengths. In contrast, the other two methods exhibit much higher average AERs. We further notice that the ES method is superior to the RS method only when the activity number is 1. The reason is that the ES method can retain most of the activity information when the activity is discrete, while the RS method is likely to divide a single activity into two activities, resulting in a larger average AER. We further note that since the RS method is more likely to segment the activity sequence into the wrong number of activity segments than the ES method, using the RS method often leads to a larger average activity error rate.

6.3.2 *The number of repetitive activities*

The same experiment configuration is used to evaluate the impact of the repetitive activity number on ZUMA. The only difference is that the activities in each chunk are the same. The average AERs are shown in Fig. 15. We find that the performance of ZUMA in repetitive activities is similar to that in distinct activities. The average AER of ZUMA varies from 7.58% to 15.04%, significantly outperforming the other two methods. These results verify that ZUMA can achieve accurate continuous activity recognition across arbitrary activity sequences.

6.3.3 *Activity duration*

We further evaluate the impact of activity duration on the performance of ZUMA. We ask volunteers to execute activities at two speeds separately and then alternate the two speeds in a single chunk. The recognition results are shown in Fig. 16. Since our explicit adjustment-based segmentation can accurately locate the largest recognition confidence in the search space, ZUMA can achieve stable recognition accuracy under various activity durations, varying slightly from 10.93% to 14.17%. In contrast, the other two methods perform poorly under variable durations as they cannot locate the boundaries of the variable-duration activities well.

6.3.4 *Distance between people and radar*

After evaluating the impact of activity sequence length and activity duration, we further evaluate the impact of the distance between the volunteer and the radar on the performance of ZUMA. The distance between the volunteer and the radar varies from 1 m to 2 m. The recognition results are shown in Fig. 17. The results show that ZUMA can maintain a low AER under different distances between the volunteer and the radar. As the distance between the volunteer and the radar increases, the AER of ZUMA increases from 12.67% to 16.75%. This increase can be attributed to the

diminishing SNR of the reflected signal with the increasing distance, consequently leading to a decrease in the average recognition accuracy of ZuMa.

6.3.5 Angle between people and radar

We then evaluate the impact of the angle between the volunteer and the radar on the performance of ZuMa. The angle between the volunteer and the radar varies from 0° to 45°. The average AERs are shown in Fig. 18. We find that as the angle between the volunteer and the radar increases, the AER of ZuMa increases from 12.67% to 25.46%. This increase can be attributed to the weakening intensity of the reflected signal received by the mmWave radar with the increasing angle, which reduces the integrity of activity-related information in the reflected signals, resulting in a significant increase in the average recognition error.

6.3.6 Experiment scene

We further evaluate the impact of experimental scenes on ZuMa's performance. We ask volunteers to execute activities in three experimental scenes: meeting room, hallway, and laboratory. Among them, the meeting room scene is the simplest and the laboratory scene is the most complex. The recognition results are shown in Fig. 19. ZuMa achieves its lowest AER of 12.67% in the meeting room scene, whereas it achieves the highest AER of 20.38% in the laboratory scene. As the scene layout becomes more complex, the radar is subject to more pronounced multipath effects, thereby diminishing the effectiveness of spectrum subtraction in signal preprocessing to suppress environmental noise.

Besides, we evaluate the generalizability of ZuMa to unseen scenes. We generate an alternative training dataset TS1 containing data collected in the meeting room and the hallway, while the original training dataset TS contains data collected in all three scenes. Training with TS1 and testing with data in the unseen laboratory yields an AER of 24.20%, slightly larger than training with TS (20.38%). The AER does not rise significantly, showing good generalizability of ZuMa to unseen scenes.

6.3.7 The size of search space

The size of search space in the explicit segmentation adjustment module is an important impact factor for the performance of ZuMa and is determined by the search margin M. It significantly affects the accuracy of locating the largest recognition confidence with the parallel divide-and-conquer search algorithm. In this part, we respectively set the search margin to 5, 10, 15, and 20 to evaluate the impact of the search space size. The recognition results of ZuMa with different search margins are shown in Fig. 20. The average AER reaches the minimum value of 12.67% when the search margin is 10. When the search margin is small, the search space may not encompass the actual activity segment, leading to higher recognition errors. When the search margin is large, the search space may include a large part of another activity segment, thereby leading to increased recognition errors.

6.3.8 The size of selected part

The size of the selected part in each search round is another important impact factor for ZuMa. It has a large impact on the accuracy and efficiency of explicit segmentation adjustment. We respectively set the size set as $[10, 5, 2, 1]$, $[5, 2, 1]$, $[5, 1]$, and $[1]$ to evaluate its impact on the performance of ZuMa. Each number in these sets represents both the width and height of the selected part in a certain round, which are always the same in ZuMa. The size set of $[1]$ represents a traversal search in the search space. The recognition results are shown in Fig. 21. We find that the average AER under the size set $[5, 2, 1]$ is the lowest. When the size is too large, the center sample of the part where the actual activity segment is located may be distant from the actual activity segment, causing this part to not be selected for the next search round. Conversely, when the size is too small, the searching process is close to or equal to a traversal search, making it more likely to locate the local extreme points of other classes in the search space, leading to higher recognition errors.

6.4 Latency

We measure the recognition latency as the time between the moment the human ends the activity and the moment the corresponding recognition result is obtained. The aggregated recognition latency consists of four components: *i)* the *waiting (WA) delay*, that is, the time required to accumulate sufficient mmWave sequence to construct the search space; *ii)* the *signal preprocessing (SP) delay*, which equals the computation time of the Range-Doppler spectrums; *iii)* the *Chunk extraction (CE) delay*, that is, the time taken for calculating the velocity entropy sequence and conducting threshold comparisons; and *iv)* the *explicit adjustment-based segmentation (EAS) delay* that is the time ZuMa requires to locate the activity boundaries.

The aggregated latency of activity recognition using ZuMa with different activity durations is shown in Fig. 22. The only varying latency component is the WA delay. Longer activity durations entail capturing less additional mmWave sequence, resulting in shorter waiting delays. Our measurements indicate that the average SP delay is 0.75 s with a standard deviation of 0.06 s. Additionally, the average FE delay and EAS delay are measured as 0.005 s and 0.091 s, respectively. The plot illustrates a linear relationship between the aggregated latency and the activity duration. This is because ZuMa necessitates sufficient search space to ensure the accuracy of explicit segmentation adjustment. As a whole, the average recognition delay is measured as 1.86 s. Furthermore, the entire computation delay is measured as 0.85 s, which is less than the duration of any activity, affirming ZuMa's capability to achieve real-time continuous activity recognition in practical deployment scenarios. The low computation delay is achieved by selecting range bins with human activities and parallelizing signal preprocessing operations and logit value computation.

6.5 Ablation Study

We finally conduct ablation studies to verify the necessity of each module in ZuMa. Given that the TSN model and the explicit segmentation adjustment module have been comprehensively evaluated previously, we focus on evaluating the activity chunk extraction module and the feedback metric selection individually.

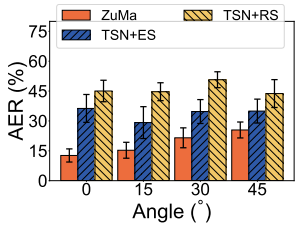


Figure 18: The impact of the angle between people and radar

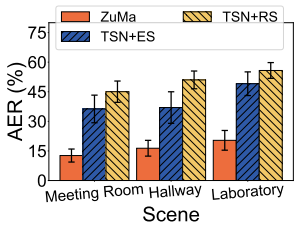


Figure 19: The impact of experiment scene

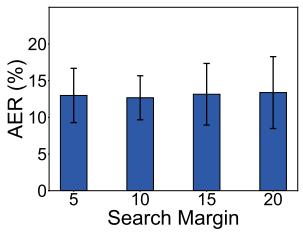


Figure 20: The impact of search space size

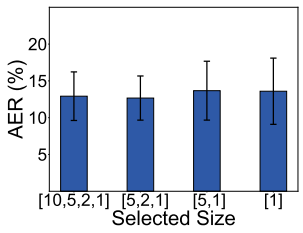


Figure 21: The impact of selected part size

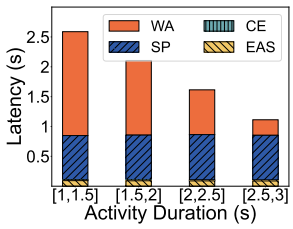


Figure 22: Aggregated latency

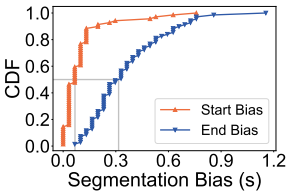


Figure 23: The performance of activity chunk extraction

6.5.1 *Activity chunk extraction*

We evaluate the chunk extraction accuracy by calculating the segmentation bias between the ground truth and the extracted chunks. We separately calculate the start bias and the end bias to show the extraction results. The CDFs of segmentation biases are shown in Fig. 23. The average start bias is 0.1 s and the average end bias is 0.38 s. The results show that both the start and end boundaries are well located by our chunk extraction module, and the start boundary is located more accurately. The reason is that the mutation of velocity entropy variance is always synchronized with the human state change at the activity start, while it may be delayed at the activity end due to potential redundant swinging or shaking movements by the volunteer.

To further evaluate the necessity of the activity chunk extraction module. We directly disable this module and set the first frame of the entire mmWave sequence as the initial start frame in the first segmentation adjustment. In this case, ZuMa achieves a much higher average AER of 83.53%. The reason is that if the initial start frame is not well located, the segmentation adjustment will be confused and may even start from the middle of an activity, which leads to a cascade of recognition errors. Besides, when this module is disabled, the computational overhead increases significantly due to a lot of unnecessary logit calculations.

6.5.2 *Recognition confidence v.s. recognition score*

To evaluate the necessity of selecting recognition confidence as the feedback metric, we replaced it with recognition scores obtained directly from the logits via the *Softmax* module. The *Softmax* module takes logits as input and normalizes them to the recognition scores proportional to the exponentials of the logits. In this case, ZuMa achieves an average AER of 14.98%, which is higher compared to using recognition confidence. The reason is that the *Softmax* module tends to over-amplify larger values in recognition scores through exponential normalization. As a result, recognition scores become less sensitive to the segmentation, leading to a higher recognition error.

7 Discussion

In this section, we discuss some practical problems and potential opportunities, including generalization, multi-target recognition, and system limitations.

7.1 Generalization

As shown in Section 6.2, we have verified that our explicit segmentation adjustment method significantly enhances the activity recognition accuracy of three recognition models in continuous activity scenarios. In fact, ZuMa has the potential to extend any single-shot activity recognition model to one suited for continuous activity recognition, provided the model can handle variable-duration data and is sensitive to activity sequence segmentation. This is because the explicit segmentation adjustment method only uses these recognition models as black boxes, feeds activity segments into them, and collects corresponding logits. In ZuMa, we choose the TSN model since it satisfies the above conditions and characterizes lightweight implementation as well as excellent recognition capability.

Moreover, ZuMa can be extended to various modalities like Wi-Fi, RFID, acoustic and IMU if two challenges can be tackled. First, a low-level feature should be found to distinguish the frames with and without human activities. In ZuMa, the variance of the velocity entropy within a time window acts as the low-level feature to clip out the human activities. Second, a feedback model tailored to the modality should be developed. In ZuMa, the Doppler spectrum of the mmWave signal can be processed by convolutional neural networks and the logit value indicates the accuracy of segmentation; in another modality, another indicator of the quality of segmentation is required.

It is also possible to perform authentication of persons with ZuMa after some modifications. For the authentication task, a new training dataset is required and the model should be retrained. The human activity recognition solution in ZuMa stresses the common features of each activity and ignores the personal details of activities. However, the authentication task relies on the personal details and ignores the common features. Moreover, different activities of the same person will be labeled as the same class (the class label for the authentication task is the person's ID), which requires are more accurate segmentation method.

7.2 Multi-target Recognition

As the mmWave radar can track and sense multiple targets simultaneously [14], [24], [44], multi-target activity recognition can be implemented by extending our design. The signal preprocessing module can be modified to simultaneously extract the reflected signals of multiple targets by using multiple Tx and Rx antennas. Once the multiple targets can be separated, their corresponding range-Doppler spectrums can be obtained, enabling multi-target recognition by individually analyzing each range-Doppler spectrum sequence. However, the multipath issue and limited angle resolution of the mmWave radar may prevent multiple target separation, which need to be carefully addressed but are beyond the scope of this paper.

7.3 Deep Learning Model Selection

We choose the TSN model considering the computational overhead and the classification accuracy.

The computational overhead should be minimized to achieve real-time human activity recognition, as the model is frequently used for inference in the divide-and-conquer search in Section 4.4.2. Compared to LSTM, RNN and attention-based models like the transformer, the TSN model has less layers and neurons, resulting in less computational overhead and time cost.

Besides, we make a trade-off between the classification accuracy and the dataset size. With the same training dataset, we achieve a better classification accuracy with the TSN model than LSTM (a specific instance of RNN) and CNN as shown in Section 6.2 and Fig. 12. A better accuracy can be achieved by training the LSTM with a much larger dataset. Nevertheless, a long collection process of the large dataset reduces the flexibility of ZuMa and prevents frequent updates to support new activities.

7.4 Analysis and Improvement of Recognition Accuracy

The recognition errors are mainly caused by deviation of activities, environmental changes and environmental noise. To mitigate these errors, some measures can be taken to improve the dataset. When collecting the dataset, it is recommended to record the activities in various indoor or outdoor environments. Different types of deviation from the standard activity pattern should also be recorded. Moreover, data augmentation techniques can be applied to improve the size and coverage of the dataset.

7.5 mmWave Signal Preprocessing

mmWave-based sensing solutions are typically based on the point cloud or the Doppler spectrum, and ZuMa applies to both of them. The point cloud-based and Doppler spectrum-based solutions have different advantages and disadvantages when applied to continuous activity recognition.

The point cloud is easier to obtain than the Doppler spectrum. The point cloud is available in more hardware toolkits, for example, a single TI IWR1642 radar provides the point cloud, while collecting the Doppler spectrum requires the radar to cooperate with a TI DCA1000 data capture board.

Nevertheless, the Doppler spectrum is more accurate and leads to better activity classification accuracy. Based on the range-FFT operation and the angle-FFT operation (or beamforming), the accuracy of the point cloud is limited by the radar's angle resolution (typically 15 degrees). On the other hand, the Doppler spectrum comes from the range-FFT operation and the Doppler-FFT operation, and its accuracy is determined by the velocity resolution (0.075 m/s in the configuration in ZuMa).

7.6 System Limitations

While our system can quickly locate the activity boundaries and enable accurate continuous activity recognition, it does suffer from some practical limitations. Firstly, the coverage of the mmWave radar is constrained due to the rapid signal attenuation and the limited Field of View (FoV), which necessitates careful deployment of the radar to cover critical sensing scenarios. Secondly, as the received signal of the mmWave radar contains complex environment background noise, background noise estimation is required for the spectrum subtract algorithm in signal preprocessing when deploying our system in new scenarios. Fortunately, this collection process is a one-time task and easy to implement. Finally, although our system can operate in real-time as the entire computation delay is less than the activity duration, there is a certain recognition latency due to the necessity of acquiring sufficient mmWave sequences for searching.

8 Conclusion

In this paper, we present ZuMa, the first mmWave-based approach for real-time continuous activity recognition, which leverages explicit adjustment of activity sequence segmentation to achieve high recognition accuracy. We first point out that unclear activity boundaries and variable activity durations are the main obstacles preventing accurate continuous activity recognition. Then we provide an in-depth analysis of the correlation between the segmentation accuracy and the recognition outputs. Inspired by this observation, we propose our coarse-to-fine grained approach including a series of modules, from signal preprocessing to explicit segmentation adjustment. Extensive experiments under real-world scenarios show that ZuMa can achieve accurate continuous activity recognition in real time.

References

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems (NIPS'15)*, 2015.

[2] T. Lin, X. Zhao, and Z. Shou, "Single shot temporal action detection," in *Proceedings of the 25th ACM International Conference on Multimedia (MM'17)*. California, USA: ACM, 2017.

[3] X. Yang, X. Yang, M.-Y. Liu, F. Xiao, L. S. Davis, and J. Kautz, "Step: Spatio-temporal progressive learning for video action detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'19)*. Long Beach, CA: IEEE, 2019.

[4] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, "Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities," *ACM Computing Surveys*, 2021.

[5] X. Ouyang, X. Shuai, J. Zhou, I. W. Shi, Z. Xie, G. Xing, and J. Huang, "Cosmo: contrastive fusion learning with small data for multimodal human activity recognition," in *Proceedings of the 28th Annual International Conference on Mobile Computing and Networking (MobiCom'22)*. Sydney, Australia: ACM, 2022.

[6] J. Hou, X.-Y. Li, P. Zhu, Z. Wang, Y. Wang, J. Qian, and P. Yang, "Signspeaker: A real-time, high-precision smartwatch-based sign language translator," in *Proceedings of the 25th Annual International Conference on Mobile Computing and Networking (MobiCom'19)*. Los Cabos, Mexico: ACM, 2019.

[7] Y. Zhang, Y. Zheng, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Widar3. 0: Zero-effort cross-domain gesture recognition with wi-fi," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[8] J. Zhang, R. Xi, Y. He, Y. Sun, X. Guo, W. Wang, X. Na, Y. Liu, Z. Shi, and T. Gu, "A survey of mmwave-based human sensing: Technology, platforms and applications," *IEEE Communications Surveys & Tutorials*, 2023.

[9] 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 the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (UbiComp'20)*, 2020.

[10] W. Jiang, C. Miao, F. Ma, S. Yao, Y. Wang, Y. Yuan, H. Xue, C. Song, X. Ma, D. Koutsonikolas *et al.*, "Towards environment independent device free human activity recognition," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom'18)*. New Delhi, India: ACM, 2018.

[11] Y. Ma, G. Zhou, S. Wang, H. Zhao, and W. Jung, "Signfi: Sign language recognition using wifi," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (UbiComp'18)*, 2018.

[12] W. Chen, H. Yang, X. Bi, R. Zheng, F. Zhang, P. Bao, Z. Chang, X. Ma, and D. Zhang, "Environment-aware multi-person tracking in indoor environments with mmwave radars," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (UbiComp'23)*, 2023.

[13] Y. Li, D. Zhang, J. Chen, J. Wan, D. Zhang, Y. Hu, Q. Sun, and Y. Chen, "Towards domain-independent and real-time gesture recognition using mmwave signal," *IEEE Transactions on Mobile Computing*, 2022.

[14] H. Xue, Q. Cao, Y. Ju, H. Hu, H. Wang, A. Zhang, and L. Su, "M4esh: mmwave-based 3d human mesh construction for multiple subjects," in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems (SenSys'22)*. Boston, USA: ACM, 2022.

[15] H. Liu, K. Cui, K. Hu, Y. Wang, A. Zhou, L. Liu, and H. Ma, "Mtranssee: Enabling environment-independent mmwave sensing-based gesture recognition via transfer learning," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (UbiComp'22)*, 2022.

[16] P. S. Santhalingam, A. A. Hosain, D. Zhang, P. Pathak, H. Rangwala, and R. Kushalnagar, "mmasl: Environment-independent asl gesture recognition using 60 ghz millimeter-wave signals," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (UbiComp'20)*, 2020.

[17] S. Palipana, D. Salami, L. A. Leiva, and S. Sigg, "Pantomime: Mid-air gesture recognition with sparse millimeter-wave radar point clouds," *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies (UbiComp'21)*, 2021.

[18] H. Liu, A. Zhou, Z. Dong, Y. Sun, J. Zhang, L. Liu, H. Ma, J. Liu, and N. Yang, "M-gesture: Person-independent real-time in-air gesture recognition using commodity millimeter wave radar," *IEEE Internet of Things Journal*, 2021.

[19] 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.

[20] R. Song, D. Zhang, Z. Wu, C. Yu, C. Xie, S. Yang, Y. Hu, and Y. Chen, "Rf-url: unsupervised representation learning for rf sensing," in *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking (MobiCom'22)*. Sydney, Australia: ACM, 2022.

[21] 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 the 2018 Conference of the ACM Special Interest Group on Data Communication (SIGCOMM'18)*. Budapest, Hungary: ACM, 2018.

[22] 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 the 19th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys'21)*. New York, USA: ACM, 2021.

[23] A. Khamis, B. Kusy, C. T. Chou, M.-L. McLaws, and W. Hu, "Rfwash: a weakly supervised tracking of hand hygiene technique," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems (SenSys'20)*. Yokohama, Japan: ACM, 2020.

[24] 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 the 20th Annual International Conference on Mobile Systems, Applications and Services (MobiSys'22)*. New York, USA: ACM, 2022.

[25] H. Wang, D. Zhang, Y. Wang, J. Ma, Y. Wang, and S. Li, "Rt-fall: A real-time and contactless fall detection system with commodity wifi devices," *IEEE Transactions on Mobile Computing*, 2016.

[26] S. Li, X. Li, Q. Lv, G. Tian, and D. Zhang, "Wifit: Ubiquitous bodyweight exercise monitoring with commodity wi-fi devices," in *2018 IEEE SmartWorld*. IEEE, 2018.

[27] 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 the 20th Annual international conference on Mobile Computing and Networking (MobiCom'14)*. Maui, Hawaii, USA: ACM, 2014.

[28] D. Zhang, X. Zhang, S. Li, Y. Xie, Y. Li, X. Wang, and D. Zhang, "Lt-fall: The design and implementation of a life-threatening fall detection and alarming system," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (UbiComp'23)*, 2023.

[29] 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 the 26th Annual International Conference on Mobile Computing and Networking (MobiCom'20)*. London, United Kingdom: ACM, 2020.

[30] T. Li, L. Fan, M. Zhao, Y. Liu, and D. Katabi, "Making the invisible visible: Action recognition through walls and occlusions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV'19)*. Seoul, Korea: ACM, 2019.

[31] S. Ashry, T. Ogawa, and W. Gomaa, "Charm-deep: continuous human activity recognition model based on deep neural network using imu sensors of smartwatch," *IEEE Sensors Journal*, 2020.

[32] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks for action recognition in videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[33] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, 2015.

[34] C. M. Bishop, *Neural networks for pattern recognition*. Oxford University Press, 1995.

[35] C. Jiang, J. Guo, Y. He, M. Jin, S. Li, and Y. Liu, "mmvib: micrometer-level vibration measurement with mmwave radar," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking (MobiCom'20)*. London, United Kingdom: ACM, 2020.

[36] S. V. Vaseghi, *Advanced digital signal processing and noise reduction*. John Wiley & Sons, 2008.

[37] S. Deldari, D. V. Smith, A. Sadri, and F. Salim, "Espresso: Entropy and shape aware time-series segmentation for processing heterogeneous sensor data," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (UbiComp'20)*, 2020.

[38] A. Howedi, A. Lotfi, and A. Pourabdollah, "An entropy-based approach for anomaly detection in activities of daily living in the presence of a visitor," *Entropy*, 2020.

[39] F. A. Pujol, M. J. Pujol, C. Rizo-Maestre, and M. Pujol, "Entropy-based face recognition and spoof detection for security applications," *Sustainability*, 2019.

[40] "Iwr6843 intelligent mmwave overhead detection sensor (ods) antenna plug-in module," https://www.ti.com/product/IWR6843ISK-ODS/part-details/IWR6843ISK-ODS, 2023.

[41] "Real-time data-capture adapter for radar sensing evaluation module," http://www.ti.com/tool/DCA1000EVM, 2023.

[42] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems (NIPS'19)*, 2019.

[43] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, 1997.

[44] J. Guo, M. Jin, Y. He, W. Wang, and Y. Liu, "Dancing waltz with ghosts: measuring sub-mm-level 2d rotor orbit with a single mmwave radar," in *Proceedings of the 20th International Conference on Information Processing in Sensor Networks (IPSN'21)*. Nashville, USA: ACM/IEEE, 2021.