DI-Gesture: Domain-Independent and Real-Time Gesture Recognition with Millimeter-Wave Signals

Yadong Li[1], Dongheng Zhang[1], Jinbo Chen[1], Jinwei Wan[2], Dong Zhang[1], Yang Hu[1], Qibin Sun[1], Yan Chen[1]
[1] School of Cyber Science and Technology, University of Science and Technology of China, Hefei, China
[2] China Nanhu Academy of Electronics and Information Technology, Jiaxing, China
Email: {yadongli, jinbochen}@mail.ustc.edu.cn, {dongheng, eeyhu, qibinsun, eecyan}@ustc.edu.cn
wan3137@163.com, zdtop@iat.ustc.edu.cn

***Abstract*—Human gesture recognition using millimeter wave (mmWave) signals provides attractive applications including smart home and in-car interface. While existing works achieve promising performance under controlled settings, practical applications are still limited due to the need of intensive data collection, extra training efforts when adapting to new domains (i.e. environments, persons and locations) and poor performance for real-time recognition. In this paper, we propose DI-Gesture, a domain-independent and real-time mmWave gesture recognition system. Specifically, we first derive the signal variation corresponding to human gestures with spatial-temporal processing. To enhance the robustness of the system and reduce data collecting efforts, we design a data augmentation framework based on the correlation between signal patterns and gesture variations. Furthermore, we propose a dynamic window mechanism to perform gesture segmentation automatically and accurately, thus enabling real-time recognition. Finally, we build a lightweight neural network to extract spatial-temporal information from the data for gesture classification. Extensive experimental results show DI-Gesture achieves an average accuracy of 97.92%, 99.18% and 98.76% for new users, environments and locations, respectively. In real-time scenario, the accuracy of DI-Gesture reaches over 97% with an average inference time of 2.87ms, which demonstrates the superior robustness and effectiveness of our system.**

***Index Terms*—Gesture recognition, mmWave sensing, data augmentation, neural network**

I. Introduction

Human gesture recognition plays an important role in human-computer interface systems, which provides users with a more natural and convenient way to interact and control machines and devices. For instance, in smart homes, people can control household Internet of Things (IoT) devices with gestures in a contactless way, which provides entertaining user experiences.

Traditional approaches for gesture recognition are based on cameras [1] or wearable sensors [2]. Although these techniques have achieved impressive recognition accuracy, their deployments in real-world applications still remain challenges. Camera-based solutions have to deal with illumination variations and privacy issues while wearable sensors require physical contact between the human body and device, which is uncomfortable and not suitable for long-term use. To resolve these challenges, the recent wireless sensing technique has demonstrated its ability for contactless sensing, including vital sign monitoring [3], gait recognition [4] and pose estimation [5]. Compared with traditional sensing methods, the wireless sensing technique is more privacy-friendly and robust under different illumination conditions. In the past years, gesture recognition based on different wireless mediums, including WiFi [6–8], acoustic signals [9] and millimeter-wave [10] has been investigated. Among these mediums, millimeter-wave draws lots of attention due to its significant advantages. First of all, the fine-grained range and angle resolution of mmWave radar enable sensing of subtle motion. Secondly, the high frequency band of the mmWave signal enables a strong anti-interference ability.Finally, it is easy to be embedded in portable devices due to the small size of mmWave radar chip [11]. Hence, many research efforts have been made to exploit mmWave signals for gesture recognition [10–15]. For instance, deep-soli [11] achieves fine-grained gesture recognition with a compact mmWave radar and deep neural network. RadarNet [12] designs an efficient neural network and collects a large-scale dataset including over 4 million samples to train a robust model. Liu. et al. [16] extract dynamic variation of gestures from mmWave signals and design a lightweight CNN to recognize gestures in long-range scenarios. Beyond recognizing simple gestures, mmASL [17] extracts frequency features from 60GHz mmWave signals and achieves American sign language recognition with a multi-task deep neural network.

However, the existing methods are limited in three aspects: (i) Labor intensive data collection. To ensure robustness of the deep learning model, researchers have to collect sufficient training data to prevent overfitting, which is tedious and impractical. (ii) Domain dependence. Deep learning based approaches achieve high accuracy when training and testing the model under familiar domains. However, model retraining or extra-training efforts are still required when adapting to new domains since the propagation of mmWave signals is subject to change upon the variations of environment setup, relative locations and gesture speed of individuals. (iii) Off-line recognition and poor performance in real-time scenarios. Most of existing works focus on off-line gesture recognition, assuming that all gesture samples are well segmented before passing into the classifier. However, in practical scenarios, the system operates in real-time, which is more difficult than the segmented classification task since the radar continuously receives signals with unknown gesture boundaries. RadarNet [12] performs the unsegmented recognition task using the sliding window approach. However, its performance is susceptible

to the selection of sliding window size, especially when people perform gestures with variant speeds.

In this paper, we propose DI-Gesture, a real-time mmWave gesture recognition system that can generalize to gestures performed by new users, at new locations or in new environments with high accuracy and low latency. The main contributions of this paper can be summarized as follows.

(1) We extract spatial-temporal changes of gesture patterns while reducing the influence of environment and user discrepancy. We further design a data augmentation framework for mmWave signals based on the relationship between DRAI representations and gesture variations to ease the pain of data collection and improve the robustness of the classifier. To the best of our knowledge, we are the first to address the domain dependence problem of mmWave gesture recognition in various domains (i.e. environment, person and location).

(2) We present a dynamic window mechanism based on different characteristics between motion frame and static frame to detect the starting and ending of gestures, which enables our system to achieve accurate real-time gesture classification.

(3) We implement DI-Gesture on a commodity mmWave radar and conduct extensive evaluations. The experiment results demonstrate the impressive performance of DI-Gesture in terms of recognition accuracy under cross-domain settings and real-time scenarios.

(4) We collect and label a comprehensive mmWave gesture dataset from various domains, consisting of 24050 samples from 25 volunteers, 5 locations and 6 environments, which has been public to the research community[1]. We believe that this dataset would facilitate future research of mmWave gesture recognition.

II. System Design

A. Signal Processing

As shown in Fig. 1, DI-Gesture employs the Frequency Modulated Continuous Wave (FMCW) radar to obtain the Dynamic Range Angle Image (DRAI) for gesture recognition. Specifically, we first perform 3D-FFT on raw signals to derive the ranges, velocities and angles of gestures. Then, we conduct noise elimination to filter environmental interference and improve the robustness of the classifier.

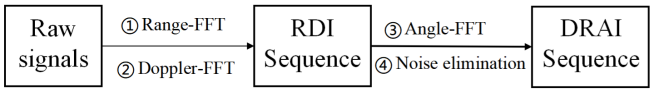


Fig. 1. The calculation process of DRAI.

(1) Range-FFT: The radar continuously transmits FMCW signals (i.e. chirps) , which will be reflected after hitting the detected object and received by receive antennas. Then the mixer on the radar board will mix the received chirp with the transmitting chirp to obtain an intermediate frequency (IF) signal. The relationship between the frequency of IF signal f and distance d between radar and object can be denoted as

$$f = S \cdot \tau = \frac{S \cdot 2d}{c} \Rightarrow d = \frac{fc}{2S}, \tag{1}$$

where S is the slope of the chirp signal, c is the speed of the signal. Therefore, the range of the detected object can be computed using FFT.

(2) Doppler-FFT: To derive the moving speed of the targeted object, the radar transmits a frame that consists of N chirps. The velocity of the object v can be derived from the phase difference $\Delta\phi$ caused by the doppler effect between two adjacent chirps as

$$\Delta\phi = \frac{4\pi v T_c}{\lambda} \Rightarrow v = \frac{\lambda \Delta\phi}{4\pi T_c}, \tag{2}$$

where λ is the wavelength of the signal, T_c is the time interval between two adjacent chirps. According to Eq. 2, we perform FFT among N chirps to extract doppler information and obtain Range Doppler Image (RDI).

(3) Angle-FFT: The Angle of Arrival (AoA) can be computed by cascading multiple RDIs obtained from different antennas according to phase changes between adjacent receiving antennas. The relationship between the phase differences $\Delta\phi$ and AoA θ can be derived as

$$\Delta\phi = \frac{2\pi l \sin\theta}{\lambda}, \tag{3}$$

where l is the distance between adjacent receiving antennas. After the Angle-FFT, we have obtained the range-doppler-angle matrix for further processing.

(4) Noise Elimination: Since moving targets and static clutters can be discriminated based on doppler frequency, we simply set doppler frequency lower than a velocity threshold as 0 to remove static clutter. To eliminate multipath reflections, we sum the averaged range doppler matrix along the range dimension to obtain the signal intensity of each doppler bin and experimentally set a threshold. Finally, only doppler bins whose signal intensity is higher than the threshold will be counted when summing the range-doppler-angle matrix along the doppler dimension into 2D matrix to obtain DRAI.

Fig. 2 shows a series of Range Angle Images (RAIs, i.e. directly summing the range-doppler-angle matrix along the doppler dimension without noise elimination) and DRAI when users push at different locations. From Fig. 2(a) and (b) we have two key observations: Firstly, different gestures result in different dynamic patterns in RAI and DRAI. For example, when users perform push, the brightest spot moves vertically which denotes distance changes of hands. Another observation is that compared with RAI, features in DRAI are clearer after static removal and noise elimination.

B. Data Augmentation

Since the accuracy and robustness of neural networks are highly dependent on the quantity and quality of training data, we design a data augmentation framework for mmWave signals to enrich the training data which makes it contain

[1]https://github.com/DI-HGR/cross_domain_gesture_dataset

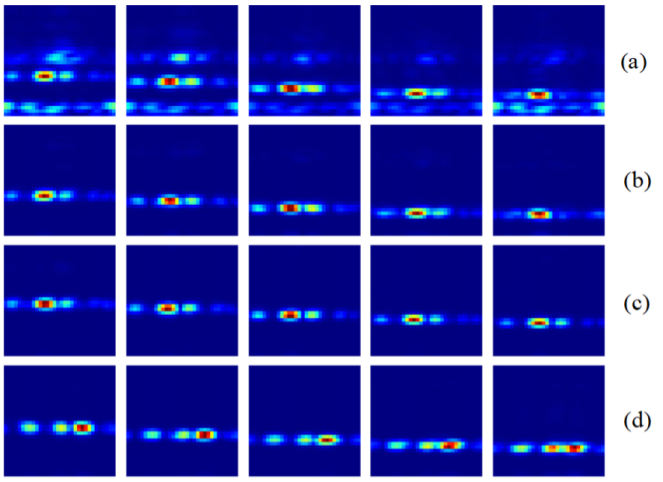


Fig. 2. Examples of push at different locations: (a) RAI of push at 60cm; (b) DRAI of push at 60cm; (c) DRAI of push at 80cm; (d) DRAI of push at 30°. Columns represent time series of 5 frames. In RAI and DRAI, pixel color, horizontal axis and vertical axis correspond to doppler power, AoA and range, respectively.

sufficient variations of gestures. The intuition behind the data augmentation is that DRAI representations vary with different gesture properties. After analysis of various practical scenarios of gesture executions, we summarize four factors which have a significant influence on DRAI data as follows:

1) Different Distances: Due to the fine-grained range information of DRAI, gestures at different distances lead to variations in DRAI. To measure the impact of the distance between radar and user, we perform push standing at 60cm and 80cm in front of the radar, respectively. Fig. 2(b) and (c) show the DRAI sequences, and we can observe vertical offset along range axis in DRAI sequences which results from different distances to the radar. Therefore, we can synthesize DRAIs of gestures performed at different distances by vertically translating all DRAIs in one sequence.

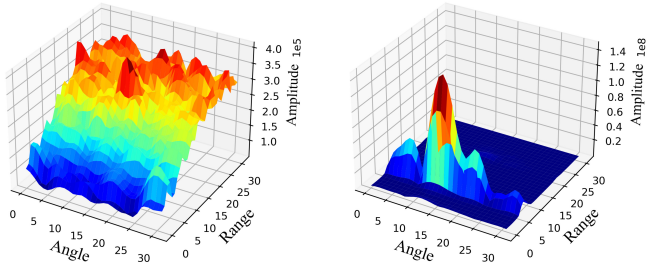
2) Different Angles: To evaluate the impact of AoA, we perform push around the radar at different angles (i.e. 60°and 120°) with 80cm away from the radar. As illustrated in Fig. 2(c) and (d), we can observe that similarly to situations of different distances, variations of AoA result in horizontal drifts of DRAI. Therefore, DRAIs of gestures performed at different angles can be generated by horizontal translation.

3) Different Speeds: It is clear that gesture samples with different speeds will have different lengths of produced DRAI sequences. Therefore, to simulate speed variations when users perform gestures, we can change the length of the DRAI sequence by downsampling and frame interpolation. To achieve this, we simply use linear frame interpolation that averagely mixes adjacent two frames to generate a new frame.

4) Different Trajectories: For the simplicity of memory and execution, we design six pair-wise gestures. Different pair-wise gestures have unique trajectories while the same pair-wise gestures have symmetry trajectories. Therefore, the DRAI sequences can be reversed to produce their pair-wise gesture data which further increase the amount of data.

C. Gesture Segmentation

To make the system work in real-time and overcome the limitation of the fixed-length sliding windows, we propose a dynamic window mechanism to adjust the window size automatically. The proposed method consists of two parts: a detection window to detect when a motion starts or ends and a recognition window to recognize whether it is a predefined gesture or a random motion. The length of the detection window is fixed while the recognition window size is exactly dependent on the gesture duration. To be specific, the first step is to distinguish whether the current frame is a motion frame (i.e. human body movement occurs in the detection range of the radar) or a static frame (i.e. no moving object). Then, a detection window is sliding along the DRAI stream to detect motion boundaries. When all frames inside the detection window are labeled as motion frames for the first time, it will be considered as the starting of a hand gesture or other unexpected motions. Once a motion start is detected, the size of the recognition window begins to increase until all frames inside the detection window are labeled as static frames, in other words, the motion ends. After that, frames belonging to the recognition window are passed into the classifier to decide whether it is a predefined gesture or not.



(a) Static frame (b) Motion frame

Fig. 3. Difference between the motion frame and static frame.

The classification of motion frame is based on the fact that energy distributions of DRAI show a great difference in different situations, as shown in Fig. 3. Specifically, there is always a series of explicit patterns (i.e. multiple high intensity peaks) in motion frames caused by gestures, while the static frames are perturbed by random noise severely. The larger differences between the energy of the peak cell and the energy of the background noise, the more likely it would be a motion frame. The detailed steps of user motion detection are as follows.

As shown in Fig. 4, assuming that the all cells c of the DRAI is set C and the signal energy of cell c is $E(c)$, the energy of the peak cell is calculated as

$$E_{peak} = \max_{c \in C} E(c) \tag{4}$$

After we find the peak cell, the rest cells surrounding the peak cell are divided into two groups: guard cells that prevent influ-

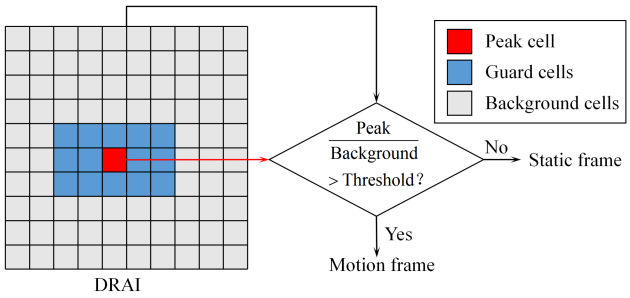


Fig. 4. Classification of motion frame and static frame

ence from motion center and background cells for estimating the average background energy of DRAI as

$$E_{bg} = \frac{\sum_{b \in B(p)} E(b)}{|B(p)|}, \tag{5}$$

where $B(p)$ represents the background cells set of the peak cell p, and $|B(p)|$ denotes the number of elements in set $B(p)$. The current frame in the DRAI sequence will be marked as a motion frame if the following equation is satisfied

$$\log(\frac{E_{peak}}{E_{bg}} + 1) > T_{motion}, \tag{6}$$

where T_{motion} is a preset threshold, and otherwise it will be marked as a static frame. In the experiments, T_{motion} is set to 1.8 empirically.

D. Gesture Recognition

The input of the classifier is a sequence of DRAI, of which each single DRAI depicts the doppler power distribution over the spatial location during a short time interval. Hence, the consecutive DRAIs describe how the distribution changes corresponding to a particular kind of gesture. Therefore, to fully extract the inherent characteristics of DRAI data, as shown in Fig. 5, we design a neural network consisting of a frame model which employs Convolutional Neural Network (CNN) to extract spatial features from each single DRAI and a sequence model which utilizes Long Short-Term Memory (LSTM) to learn the temporal dependencies of the entire sequence.

The frame model has 3 convolutional layers with kernel size 3x3, 1 fully connected layer with 128 units and batch normalization. The number of filters of the three convolutional layers increases from 8, 16 to 32. The sequence model consisting of 1 LSTM layer with 128 hidden units and 1 fully connected layer to obtain gesture probability. We set the activation function as ReLU and the dropout rate as 0.5. The network is trained with Adam optimizer with a learning rate of 0.0001 and a batch size of 64. The loss function of our model can be expressed as

$$\begin{aligned} Loss(Y, c) &= -log(\frac{exp(Y[c])}{\sum_{j=0}^{C-1} exp(Y[j])}) \\ &= -Y[c] + log(\sum_{j=0}^{C-1} exp(Y[j])) \end{aligned} \tag{7}$$

where Y refers to the output of the last fully connected layer, and C is the number of gesture classes.

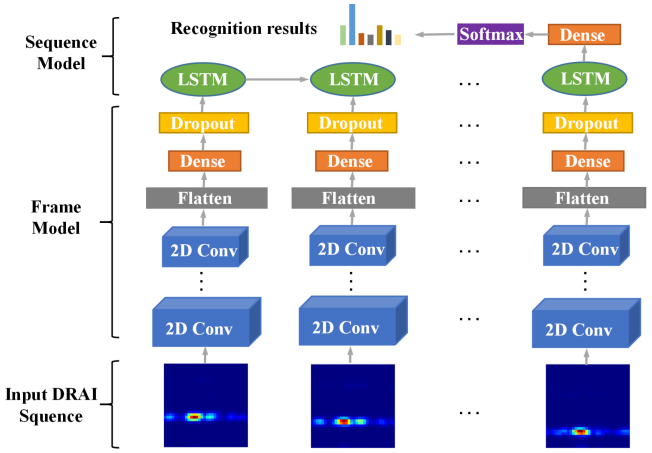


Fig. 5. Network architecture: the frame model employs CNN for spatial features extraction and the sequence model utilizes LSTM for temporal modeling.

III. Experiments

A. Data collection

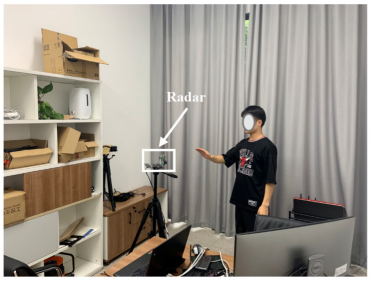


Fig. 6. Environment setup

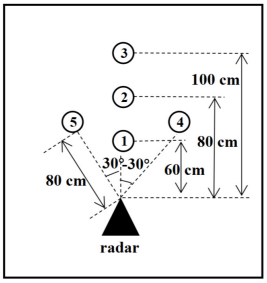


Fig. 7. Five anchor locations

To evaluate the performance of DI-Gesture, we collect gesture data from 25 volunteers, 6 environments and 5 locations. Fig. 6 shows one of the evaluation environments and Fig. 7 illustrates the setup of 5 locations in each environment. We select six common gestures as the predefined gestures, including push (PH), pull (PL), left swipe (LS) right swipe (RS), clockwise turning (CT) and anticlockwise turning (AT). To improve the robustness of the classifier and filter unexpected motions in real-time application scenarios, we also collect other human actions as negative samples (NG). In total, we have collected 24050 samples, consisting of 10650 gesture samples and 13400 negative samples.

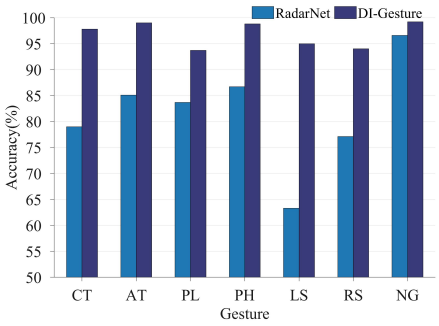


Fig. 8. Comparison of new user test

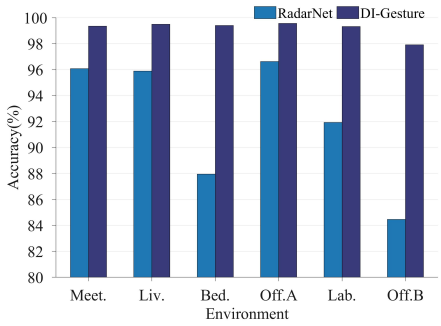


Fig. 9. Comparison of new room test

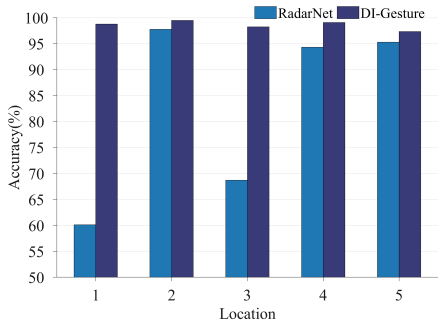


Fig. 10. Comparison of new location test

B. Device Configuration

We have implemented our gesture recognition system using TI AWR1843 mmWave radar and DCA1000 real-time data acquisition board. Each radar frame has 128 chirps and each chirp has 128 sample points. We set the chirp and frame parameters of the radar to achieve a frame rate of 20fps, a range resolution of 0.047m, and a velocity resolution of 0.039m/s. We activate 2 transmitting antennas and 4 receiving antennas to obtain an approximately angular resolution of 15°.

TABLE I
ACCURACY(%) OF IN-DOMAIN RECOGNITION

Model	Fold. 1	Fold. 2	Fold. 3	Fold.4	Fold. 5	Avg.
RadarNet [12]	96.79	98.26	98.63	98.19	97.11	97.80
DI-Gesture	**99.22**	**99.43**	**99.68**	**99.64**	**99.41**	**99.48**

C. Ability of In-Domain Recognition

To demonstrate the superior performance of DI-Gesture, we implement the state-of-the-art RadarNet proposed in [12] with the same TI-AWR1843 board for comparison. We first compare the ability of in-domain recognition with RadarNet. We take 80% data from each domain for training and then test the remaining 20% by 5-fold cross-validation. As shown in Table I, both two methods work quite well on familiar domains.

D. Ability of Cross-Domain Recognition

We now compare the overall accuracy of RadarNet and DI-Gesture on different domain factors, including environment, user and location.

1) Person variety: To evaluate the person-independent performance of DI-Gesture, we train models with gesture data from User A-G and test with the data of the resting 18 persons. The recognition accuracy for each gesture of new user test is presented in Fig. 8. As a result, RadarNet's performance on new users significantly drops to 88.64% while DI-Gesture-Lite can still preserve an impressive accuracy of 97.92%.

2) Environment diversity: To investigate the cross-environment performance of DI-Gesture, we adopt leave-one-environment-out test meaning that taking data collected in 1 room as the test set and the other 5 rooms as the training set. As Fig. 9 depicts, The recognition accuracy of DI-Gesture outperforms RadarNet in all different unseen environments.

3) Location variation: To investigate the performance of DI-Gesture at different locations, we conduct leave-one-location-out test, which denotes that one location as the test set and the rest four locations as the training set. As shown in Fig. 10, RadarNet is able to perform well at location 2, 3 and 5 but cannot generalize to location 1 and 3, This is because that location 2, 3 and 5 have different angles relative to radar while location 1 and 3 are at different ranges away from the radar. Therefore, gestures performed at location 1 and 3 show significant differences due to the fine-grained range resolution of RDI. Instead, DI-Gesture solves this problem by the data augmentation technique which can synthesize gesture data at different locations thus achieving high accuracy on different locations.

4) Analysis: From the above cross domain evaluations, we believe that DI-Gesture performs better than RadarNet for two reasons: (i) DI-Gesture successfully filters signals reflected from the human torso and static objects in the environment while preserving the inherent characteristics of gestures. (ii) The proposed data augmentation methods indeed provide more quality data to reduce the influence of gesture inconsistency in different domains and improve the robustness of the system.

E. Performance in Real-time Scenario

1) Recognition Ability: In unsegmented recognition tasks, we use two metrics for performance evaluation, including continuous recognition accuracy (CRA) and multiple prediction rate (MPR). For CRA, the recognition result is wrong when the gesture is misclassified or there is no prediction. Besides, the MPR requires the system to make only one prediction for each gesture. Suppose that P is the number of predictions that the system output, the MPR can be expressed as

$$MPR = 1 - \frac{N}{P} \tag{8}$$

To validate the real-time recognition ability of DI-Gesture, we train DI-Gesture with data collected from 6 environments, 4 locations and 20 users. Then we ask 4 familiar users standing at the rest 1 unseen location in a new environment, which is a more practical and challenging situation. The users continuously perform each predefined gesture for 10 times and

TABLE II
COMPARISON OF CONTINUOUS RECOGNITION ACCURACY (CRA) AND MULTIPLE PREDICTION RATE (MPR) FOR NEW ROOM AND NEW LOCATION TEST

Method	CRA (%)	MPR (%)
Fixed-length sliding window	95	34.25
Dynamic window mechanism	**97.08**	**2.83**

the system uses the dynamic window mechanism to segment gestures. We also implement the fixed-length sliding window approach similar to [12] and use the same classifier for a fair comparison. The results are shown in Table II. We can observe that the proposed dynamic window mechanism achieves higher CRA and much lower MPR compared with the fixed-length sliding window. We believe this is because that fixed-length sliding windows can not handle situations that people perform gestures at different speeds. To be specific, if the frame length of fast gesture is shorter than the step size of the sliding window, it is more difficult to detect. Besides, slow gestures are more often recognized as multiple gestures when their duration is larger than the window size, resulting in an increase of MPR. In contrast, the proposed dynamic window resolves this problem by accurately detecting the start and end of the gesture, and adjusting the size of the recognition window dynamically. The result also demonstrates that DI-Gesture can work well when crossing multiple different domains.

TABLE III
MODEL SIZE AND INFERENCE TIME OF DI-GESTURE

Model	Model size (MB)	Inference time (ms)
DI-Gesture	0.69	2.87

2) Computational Consumption: To demonstrate the efficiency of our system, we evaluate the model size and inference time of DI-Gesture. We implement DI-Gesture on a laptop with CPU only and measure the inference time by taking the average inference time over 1000 runs. As shown in Table III, the model size and inference time of DI-Gesture are only 0.69MB and 2.87ms, respectively. Therefore, the computational cost of DI-Gesture is small enough for real-time implementation.

IV. CONCLUSION

In this paper, we proposed DI-Gesture, a real-time mmWave gesture recognition system that worked well across new users, new environments and new locations. DI-Gesture outperformed the state-of-the-art in two aspects: (i) The proposed signal processing pipeline and a series of data augmentation techniques enabled an impressive cross-domain accuracy without collecting extra data or model retraining; (ii) The dynamic window mechanism helped DI-Gesture achieve a more satisfying performance when the system worked in real-time. Furthermore, we collected the first cross-domain mmWave gesture dataset consisting of 24050 gesture samples from 25 volunteers, 6 environments and 5 locations and made it public to the research community. We believe that the proposed methods and released dataset not only push the mmWave gesture recognition into real-world applications, but also can be applied to other wireless sensing tasks and inspire more researchers to investigate this ubiquitous sensing technique.

REFERENCES

[1] P. Narayana, J. R. Beveridge, and B. A. Draper, "Gesture recognition: Focus on the hands," in *CVPR*, 2018, pp. 5235–5244.

[2] F. S. Botros, A. Phinyomark, and E. J. Scheme, "Electromyography-based gesture recognition: Is it time to change focus from the forearm to the wrist?" *IEEE TII*, vol. 18, no. 1, pp. 174–184, 2022.

[3] D. Zhang, Y. Hu, Y. Chen, and B. Zeng, "Breathtrack: Tracking indoor human breath status via commodity wifi," *IEEE IoTJ*, vol. 6, no. 2, pp. 3899–3911, 2019.

[4] X. Yang, J. Liu, Y. Chen, X. Guo, and Y. Xie, "Mu-id: Multi-user identification through gaits using millimeter wave radios," in *INFOCOM*, 2020, pp. 2589–2598.

[5] M. Zhao, Y. Liu, A. Raghu, T. Li, H. Zhao, A. Torralba, and D. Katabi, "Through-wall human mesh recovery using radio signals," in *ICCV*, 2019, pp. 10 113–10 122.

[6] 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 TPAMI*, pp. 1–1, 2021.

[7] Z. Chen, L. Zhang, C. Jiang, Z. Cao, and W. Cui, "Wifi csi based passive human activity recognition using attention based blstm," *IEEE TMC*, vol. 18, no. 11, pp. 2714–2724, 2019.

[8] Y. He, Y. Chen, Y. Hu, and B. Zeng, "Wifi vision: Sensing, recognition, and detection with commodity mimo-ofdm wifi," *IEEE IoTJ*, vol. 7, no. 9, pp. 8296–8317, 2020.

[9] H. Chen, F. Li, and Y. Wang, "Echotrack: Acoustic device-free hand tracking on smart phones," in *INFOCOM*, 2017, pp. 1–9.

[10] M. Scherer, M. Magno, J. Erb, P. Mayer, M. Eggimann, and L. Benini, "Tinyradarnn: Combining spatial and temporal convolutional neural networks for embedded gesture recognition with short range radars," *IEEE IoTJ*, vol. 8, no. 13, pp. 10 336–10 346, 2021.

[11] S. Wang, J. Song, J. Lien, I. Poupyrev, and O. Hilliges, "Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum," in *UIST*, 2016, p. 851–860.

[12] E. Hayashi, J. Lien, N. Gillian, L. Giusti, D. Weber, J. Yamanaka, L. Bedal, and I. Poupyrev, "Radarnet: Efficient gesture recognition technique utilizing a miniature radar sensor," in *CHI*, 2021.

[13] S. Palipana, D. Salami, L. A. Leiva, and S. Sigg, "Pantomime: Mid-air gesture recognition with sparse millimeter-wave radar point clouds," *Proc. ACM IMUWT.*, vol. 5, no. 1, Mar. 2021.

[14] 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 IoTJ*, vol. 9, no. 5, pp. 3397–3415, 2022.

[15] R. Min, X. Wang, J. Zou, J. Gao, L. Wang, and Z. Cao, "Early gesture recognition with reliable accuracy based on high resolution iot radar sensors," *IEEE IoTJ*, pp. 1–1, 2021.

[16] 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," *Proc. ACM IMUWT.*, vol. 4, no. 4, Dec. 2020.

[17] 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," *Proc. ACM IMUWT.*, vol. 4, no. 1, Mar. 2020.