m^3ASL: ASL Gesture Recognition with Moving mmWave Radar

Guiyun Fan, Rong Ding, Xiaocheng Wang, Yichen Zhu, Haiming Jin*
Shanghai Jiao Tong University
Email: {fgy726, dingrong, curryjam_cg, zyc_ieee, jinhaiming}@sjtu.edu.cn

IEEE INFOCOM 2025 - IEEE Conference on Computer Communications | 979-8-3315-4305-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/INFOCOM55648.2025.11044456

***Abstract*—This paper tries to answer the question: "Can we enable moving mmWave radar to recognize ASL gestures?" A positive answer would help facilitate the non-contact interaction between the millions of deaf or hard of hearing ASL users and radar-equipped mobile robots. The challenges, however, include how to deal with the pros and cons of radar motion, and how to exploit the multi-scale nature of radar data. This paper gives an affirmative answer by proposing m^{3}ASL, a method which achieves ASL recognition with moving mmWave radar for the first time. Specifically, m^{3}ASL generates accurate radar point cloud in a motion-ware manner: it constructs extended phase sequences for accurate angle estimation by leveraging radar motion, corrects motion-incurred phase errors via self-supervision, and calibrates motion-incurred point cloud distortion via careful spatial transformation and speed compensation. Furthermore, m^{3}ASL extracts features from multi-scale radar data accordingly with augmented graph and temporal convolution neural network, and complementarily fuses the extracted features with our proposed symmetric double cross attention neural network structure to infer ASL gestures. We conduct extensive experiments via both simulation and real-world system. The experimental results show that m^{3}ASL achieves *close to 93% recognition accuracy* on 30 representative ASL gestures, *surpassing SOTA by at least 30%*.**

I. Introduction

Nowadays, it is imperative to empower moving mmWave radar with the ability of recognizing American Sign Language (ASL) gestures. Such strong necessity originates from the fact that millions of deaf or hard of hearing people around the globe (e.g., US, Canada, Philippines) rely on ASL gestures as their primary means of communication [1], as hearing degradation is usually accompanied by loss of speech quality. For them, the common method of using human voice to enter control commands to mobile robots (e.g., delivery or cleaning robots) is infeasible, but one that leverages ASL gestures recognized in a non-contact manner (as in Fig. 1) is more desirable. Among the diverse sensors commonly equipped by mobile robots, we envision that mmWave radar is potentially a satisfactory candidate for ASL gesture recognition. On the one hand, mmWave radar is more resilient to environmental lighting conditions and incurs less privacy concerns than optical sensors (e.g., camera). On the other hand, mmWave radar is equipped with an innate capability of extracting the spatial coordinates and speeds of the reflection points on human body, which are necessary ingredients to recognize ASL gestures.

Though promising, enabling ASL gesture recognition with moving mmWave radar faces the following challenges.

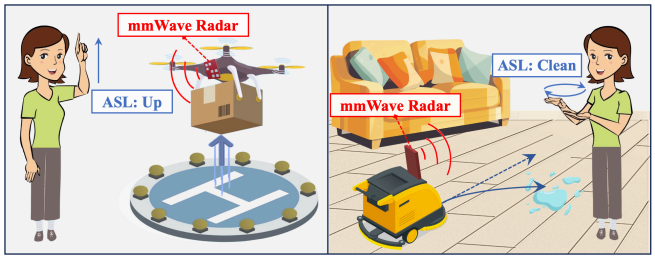

Figure 1: Examples of mobile robot control based on ASL gesture recognition with moving mmWave radar: instruct an aerial delivery robot to raise its altitude with ASL gesture UP (left), and trigger a ground cleaning robot with ASL gesture CLEAN (right).

Challenge C1: Avoiding the cons and exploiting the pros of radar motion. In nature, radar motion is a double-edged sword. On the one hand, radar motion causes severe distortion to the spatial coordinates and speeds of radar point cloud in the radar's own coordinate system. Existing mmWave-based ASL recognition method [2] is specifically tailored for stationary mmWave device, and is thus inapplicable under such motion-incurred distortion. On the other hand, however, we realize that radar motion is beneficial as well, because it naturally introduces an extended number of virtual antennas (VAs) along the radar moving direction, unveiling an attractive opportunity to construct longer phase sequences for angle estimation and achieve higher angle estimation accuracy than conventional method [3]. Therefore, one has to carefully avoid the cons and exploit the pros of radar motion to obtain accurate radar point cloud for ASL gesture recognition.

Challenge C2: Exploiting the multi-scale nature of radar data. In practice, radar data exists in multiple modalities with different scales of focuses. Specifically, radar point cloud contains concrete spatial coordinates and speeds, but only focuses on a few discrete reflection points. In contrast, Range-Angle-Doppler map (RADM) focuses on a wider scale of the radar's entire field of view with the shapes and sizes of its peaks conveying richer gesture semantics than point cloud, but is inevitably more noisy. Thus, it is critical to carefully take into account the unique properties of each radar data modality during feature extraction, and complementarily fuse the features extracted from different modalities to exploit the multi-scale nature of radar data for ASL gesture recognition.

Proposed Method. This paper proposes m^{3}ASL, a method[1] that achieves ASL gesture recognition with moving mmWave

*Corresponding author

[1]The m^3 in the name m^{3}ASL comes from moving and mmWave.

radar by addressing the above challenges. Specifically, m^{3}ASL consists of the following key components.

Motion-Aware Point Cloud Generation (for C1). This component deals with the motion of the radar to generate accurate point cloud via three key designs. (i) This component seeks to select equal-spaced VAs along the radar moving direction, and constructs long phase sequences for angle estimation by carefully stitching the peak phases of the Range-Doppler maps (RDMs) of the selected VAs. Such construction leverages radar motion to introduce multiplicative extension of the length of constructed phase sequence, and thus notably boosts angle estimation accuracy compared with conventional method [3]. (ii) In practice, due to inaccuracy of radar motion estimation, the selected VAs are not perfectly equal-spaced, resulting in non-identical difference between consecutive elements in the constructed phase sequence. Therefore, this component corrects such phase errors in a self-supervised manner via finding the correction terms that maximize the concentration of the frequency domain of the corrected phase sequence. (iii) Moreover, this component calibrates the point cloud distortion incurred by radar motion via carefully transforming the coordinates of the points in different frames to the same reference system, and compensates the speed of each point with the radar moving speed to obtain its actual radial speed.

Multi-Scale Fusing Gesture Inference (for C2). This component exploits the multi-scale nature of radar data with a parallel neural network structure to extract features from point cloud and RADM accordingly, and complementarily fuses the extracted features for ASL gesture inference. Specifically, the point cloud branch of this component exploits the inherent dynamic graphical structure of radar point cloud and augments dynamic graph convolution [4] to extract both spatial and temporal features from radar point cloud; the RADM branch of this component augments temporal convolution [5] to effectively extract features from complex-valued RADM. Furthermore, the fusion module of this component extends cross attention [6] to a symmetric double cross attention structure to enable mutual complementary fusion of the features extracted from both point cloud and RADM.

Contributions. We present the contributions of this paper as follows in terms of functionality, technique, and experiment.

- *Functionality.* This paper is the first one that achieves ASL recognition with moving mmWave radar: existing mmWave-based ASL recognition method is specifically tailored for stationary mmWave device, while this paper addresses a more challenging problem with moving mmWave radar; existing methods on gesture recognition with moving wireless devices only focus on coarse-grained gestures (e.g., lateral raise, pushing forward), while the ASL gestures that this paper recognizes are more fine-grained.
- *Technique.* Technically, the method m^{3}ASL proposed in this paper consists of (i) a point cloud generation component that generates accurate radar point cloud by carefully avoiding the cons and exploiting the pros of radar motion, and (ii) an inference component that extracts features from multi-scale radar data accordingly and complementarily fuses the extracted features to infer ASL gestures.
- *Experiment.* This paper performs extensive experiments to evaluate the performance of m^{3}ASL using both simulation and real-world system with a wide variety of environments, humans, and radar trajectories. The experimental results demonstrate that m^{3}ASL achieves close to 93% recognition accuracy on 30 representative ASL gestures, surpassing the state-of-the-art (SOTA) methods by at least 30%.

II. Related Work

Non-contact gesture recognition is considered an effective complement to voice-based command entry to mobile devices because of its resilience to voice frequency noise, and has been widely studied in recent years. Among all non-contact gesture recognition methods, those that leverage radio frequency (RF) signals (e.g., WiFi, mmWave) are more resilient to environmental lighting conditions and incur less privacy concerns than vision-based methods [7], and typically enjoy longer sensing distances than ultrasound-based methods [8–11]. The rest of this section primarily discusses existing representative RF-based gesture recognition methods [2, 12–25].

Gesture Recognition with Stationary RF Sensing Device. WiFi and mmWave are currently the most widely used RF signals for gesture recognition. Among existing representative WiFi- and mmWave-based methods, [12–19] share the common objective of recognizing coarse-grained arm-level gestures (e.g., lateral raise, pushing forward) but have different focuses: [12–16] use gesture-affected WiFi CSI as inputs, and achieve WiFi spectrogram leakage suppression [12], quick adaptation to new gesture classes [13], as well as multi-human gesture recognition [14, 15]; [17–19] use mmWave point cloud as inputs, and achieve continuous real-time gesture recognition [17], environment independence [18], as well as light-weight gesture inference [19]. Clearly, the above methods [12–19] specifically designed to recognize coarse-grained gestures lack the ability of recognizing ASL gestures which involve fine-grained palm- and finger-level body movements. Note that there exist RF-based methods [2, 20–23] capable of realizing ASL or ASL-like fine-grained gestures. However, these methods, as well as the ones discussed above are specifically tailored for stationary RF sensing device, and inevitably become inapplicable when sensing device is in motion, because of motion-incurred RF data distortion.

Gesture Recognition with Moving RF Sensing Device. There already exist works [24, 25] seeking to enable moving RF sensing device to recognize human gestures. However, neither [24] nor [25] could recognize gestures as fine-grained as ASL ones: [24] is designed for IR-UWB radar with one transmitting and one receiving antenna, which is unable to perform angle estimation and can only recognize simple gestures distinguishable in range; [25] only eliminates the negatives of radar motion without utilizing it to break the accuracy limit of angle estimation posed by the layout of radar antenna array, making [25] unable to recognize fine-grained ASL gestures.

Miscellaneous Related Work. There also exist other works that are related to this paper beyond the above two cate-

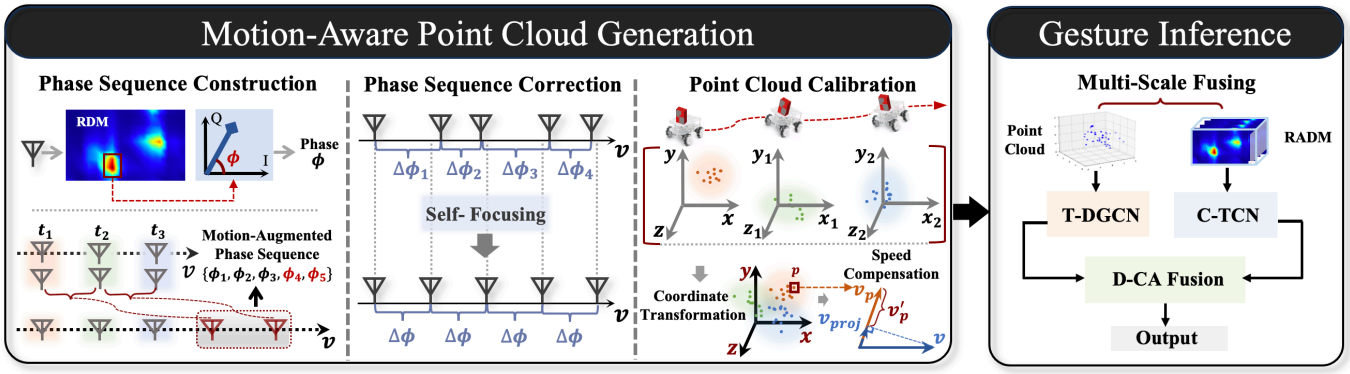

Figure 2: m^{3}ASL overview (motion-aware point cloud generation & multi-scale fusing gesture inference).

gories. [26] proposes a method to recognize ASL gestures with wearable devices, which is a contact-based method that involves installing sensors on human body. Besides, RF signals have been widely utilized to perform other human sensing tasks, such as respiration sensing [27], heartbeat waveform reconstruction [28], facial expression recognition [29], as well as many others, which are orthogonal to the task of ASL gesture recognition studied in this paper.

III. m^{3}ASL Overview

As shown in Fig. 2, m^{3}ASL contains a motion-aware phase sequence construction component and a multi-scale fusing gesture inference component, which are introduced as follows.

Motion-Aware Point Cloud Generation. This component generates accurate point cloud for ASL gesture recognition by exploiting the pros and avoiding the cons of radar motion. Specifically, this component takes as inputs the intermediate frequency (IF) signal and estimated radar ego-motion, and contains a (i) *motion-augmented phase sequence construction* method that exploits radar motion to construct long phase sequences for accurate angle estimation, a (ii) *self-focused phase sequence correction* method that corrects the phase errors in the constructed phase sequences incurred by the discrepancy between the estimated and actual radar speed, and a (iii) *motion-compensated point cloud calibration* method that compensates the point cloud distortion caused by radar motion.

Multi-Scale Fusing Gesture Inference. This component contains a multi-scale fusing neural network to fuse point cloud and RADM which are two complementary radar data modalities with different scales of focuses. Specifically, this component adopts a parallel *temporal dynamic graph convolutional network (T-DGCN)* and *complex-valued temporal convolutional network (C-TCN)* structure to extract high-level features from these two modalities. T-DGCN extends DGCN [4] by enabling it to extract not only spatial but also temporal features from point cloud, and C-TCN extends TCN [5] by enabling it to effectively extract features from complex-valued RADM. Furthermore, this component employs a *double cross attention (D-CA) fusion* module that extends cross attention [6] to a symmetric structure to enable mutual complementary fusion of the features extracted by T-DGCN and C-TCN.

IV. Motion-Aware Point Cloud Generation

Accurate point cloud is no doubt a fundamental ingredient of correctly recognizing ASL gestures. However, the low angle estimation accuracy and severe point cloud distortion incurred by radar motion collectively deteriorate point cloud accuracy. To address these two issues, we propose a motion-aware point cloud generation method, which not only fully exploits the radar motion to enhance angle estimation accuracy via motion-augmented phase sequence construction (Sec. IV-A) and self-focused phase sequence correction (Sec. IV-B), but also carefully compensates radar motion-incurred point cloud distortion via motion-compensated point cloud calibration (Sec. IV-C). We describe the details of these three components as follows.

A. Motion-Augmented Phase Sequence Construction

Intuition. We delve into the reason of the low angle estimation accuracy of existing methods which invariably perform angle estimation from both the azimuth and elevation perspectives. Specifically, to estimate the azimuth (or elevation) of a point, existing methods extract its phase from the range-doppler map (RDM) of each virtual antenna (VA) along the azimuth (or elevation) direction, construct a sequence of the extracted phases, and then perform FFT over the constructed phase sequence to obtain the estimated angle. The azimuth and elevation resolutions are thus limited by the few number of VAs that an mmWave radar has along the two directions respectively, which degrades the angle estimation accuracy.

Fortunately, we realize that *radar motion naturally introduces a number of VAs along its moving direction in different time instances*. Such extended number of VAs could enable the construction of a phase sequence long enough to accurately estimate the *point-motion angle*, defined as the included angle between $\overrightarrow{op}$ and the radar moving direction with o denoting the origin of the radar reference system in the current frame and p denoting the point to be estimated. Following such intuition, instead of seeking to boost the azimuth and elevation resolutions, we propose to enhance the angle estimation accuracy from a new perspective via accurately estimating the point-motion angle. We present our method of constructing the phase sequence to estimate such angle in detail as follows.

Design Details. We design a motion-augmented phase sequence construction method, which takes as input one frame

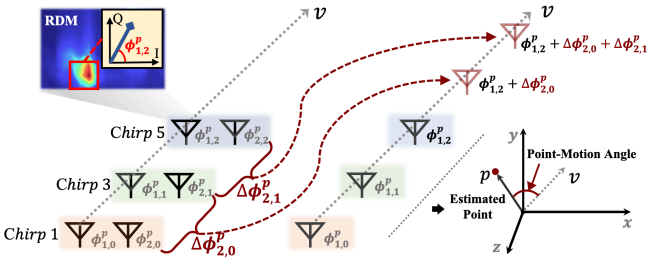

Figure 3: Motion-augmented phase sequence construction.

of IF signals of all VAs and outputs a phase vector for computing the point-motion angle of each point in the point cloud. Specifically, we let N denote the total number of VAs of our mmWave radar and index the VAs from 1 to N; we let K denote the number of chirps in a frame and index the chirps from 1 to K in chronological order; we let $s_{n,k}$ denote the IF signal that corresponds to VA n and chirp k. The detailed process of the method is described as follows.

- *Step 1 (Range-FFT).* For each IF signal $s_{n,k}$ with $n \in [1, N]$ and $k \in [1, K]$, the method calculates its range-FFT $r_{n,k}$.
- *Step 2 (Motion-Based Chirp Rearranging).* Given inter-chirp spacing δ and range-FFT sequence length M, the method constructs range-FFT sequence $\mathbf{r}_{n,m} = \{r_{n,k} : 1 + m\delta \leq k \leq K - (M - m)\delta\}$ for each $n \in [1, N]$ and $m \in [0, M]$.
- *Step 3 (Doppler-FFT).* The method calculates the doppler-FFT of each range-FFT sequence $\mathbf{r}_{n,m}$ and obtains the corresponding RDM $d_{n,m}$.
- *Step 4 (Peak Extraction).* The method extracts the peaks in the RDMs constructed in step 3 using the CFAR algorithm [3], assigns a common index[2] p to the group of $N(M+1)$ peaks located at the same position of the RDMs, and sets P as the largest peak index.
- *Step 5 (Phase Stitching).* For each peak p extracted in step 4, the method obtains its phase $\phi^p_{n,m}$ that corresponds to each RDM $d_{n,m}$, calculates a phase difference $\Delta\phi^p_{n,m} = \phi^p_{n,m+1} - \phi^p_{n,m}$ for each $n \in [2, N]$ and $m \in [0, M-1]$, constructs a phase difference sequence $\{\Delta\phi^p_{2,0}, \cdots, \Delta\phi^p_{2,M-1}, \cdots, \Delta\phi^p_{N,0}, \cdots, \Delta\phi^p_{N,M-1}\}$, and sets $\Delta\phi^p_q$ as the qth element in the phase difference sequence. The method stitches the phase differences into a phase sequence $\Psi^p = \{\psi^p_i : 1 \leq i \leq NM+1\}$ with

$$\psi^p_i = \begin{cases} \phi^p_{1,i-1}, & \text{if } i \in [1, M+1] \\ \phi^p_{1,M} + \sum_{q=1}^{i-M-1} \Delta\phi^p_q, & \text{if } i \in [M+2, NM+1] \end{cases}.$$

The method outputs Ψ^p as the constructed phase sequence.

Toy Example. We use the toy example shown in Fig. 3 with $N = \delta = M = 2$ to intuitively illustrate the above process. Under such setting, step 2 constructs 6 range-FFT sequences, namely $r_{1,1}$, $r_{1,2}$, $r_{1,3}$, $r_{2,1}$, $r_{2,2}$, and $r_{2,3}$, which leads step 3 to construct 6 RDMs $d_{1,1}$, $d_{1,2}$, $d_{1,3}$, $d_{2,1}$, $d_{2,2}$, and $d_{2,3}$. For each detected peak p, step 5 constructs a phase difference sequence as $\{\Delta\phi^p_{2,0}, \Delta\phi^p_{2,1}\}$ and the output phase sequence as $\{\phi^p_{1,0}, \phi^p_{1,1}, \phi^p_{1,2}, \phi^p_{1,2} + \Delta\phi^p_{2,0}, \phi^p_{1,2} + \Delta\phi^p_{2,0} + \Delta\phi^p_{2,1}\}$.

Rationales and Implications of Key Designs. Given that the duration of a frame is only around 0.1s, the radar can be approximately regarded as moving with some constant speed v along a straight line in one frame. Thus, step 2 essentially extends each VA to $M+1$ equal-spaced VAs along the moving direction with an inter-VA distance of $v\delta\tau$ where τ denotes the duration of a chirp. Note that we set the inter-chirp spacing δ such that the inter-VA distance is less than but close to half the wavelength of the mmWave signal, and set the range-FFT sequence length M such that the changes of the positions and speeds of the reflection points are negligible in $M+1$ chirps.

The output phase sequence has $NM+1$ elements with the same theoretical inter-element phase difference. Such length is irrelevant with the VA layout, but is positively correlated with the total number of VAs (i.e., N). We let N_a and N_e denote the number of VAs along the azimuth and elevation directions, which exactly equal to the lengths of the phase sequences that existing methods use to estimate the two angles. N is greater than both N_a and N_b, and could be even close to their product under the VA layout of common mmWave radars (e.g., TI IWR6843AOP [30]). Therefore, our motion-augmented phase sequence construction method introduces multiplicative increase in the length of the phase sequence for angle estimation compared with existing methods.

B. Self-Focused Phase Sequence Correction

In practice, minor variation of the radar moving speed v in a frame leads to non-identical inter-VA distance $v\delta\tau$, and further causes the constructed phase sequence Ψ^p to have non-identical inter-element phase difference. Thus, Ψ^p needs to be carefully corrected before being used to estimate the motion-point angle. Inspired by existing method [31] for addressing SAR image degradation, we propose to *correct the errors of Ψ^p in a self-focused manner via minimizing the entropy in the frequency domain of the corrected phase sequence*. We present the details of our method as follows.

We design a self-focused phase sequence correction method, which takes as input the constructed phase sequence Ψ^p for each peak p extracted by step 4 in Sec. IV-A, and computes a correction term φ^p_i for each element ψ^p_i of Ψ^p by solving the following optimization program, named as the phase entropy minimization (PEM) problem.

$$\textbf{PEM}: \min_{\{\varphi^p_i\}} = -\sum_{p=1}^{P}\sum_{l=0}^{NM} \frac{|A_{p,l}|^2}{Z} \ln \frac{|A_{p,l}|^2}{Z}, \tag{1}$$

where $Z = \sum_{p=1}^{P}\sum_{l=0}^{NM} |A_{p,l}|^2$, and

$$A_{p,l} = \frac{1}{NM+1} \sum_{i=1}^{NM+1} e^{-j\psi^p_i} e^{-j\varphi^p_i} e^{j\frac{2\pi}{NM+1}il} \tag{2}$$

is the lth element of the FFT of the signal sequence $\{e^{-j\psi^p_i} : 1 \leq i \leq NM+1\}$.

Having constructed **PEM**, the method proceeds as follows: it solves **PEM** using gradient descent until convergence, obtains the correction terms $\{\varphi^p_i : 1 \leq i \leq NM+1, 1 \leq p \leq P\}$,

[2]As each index p has a 1-to-1 mapping to a point in the point cloud, we use p to denote both the index of a peak and its corresponding point.

and outputs the corrected phase sequence $\Psi_c^p = \{\psi_i^p + \varphi_i^p : 1 \leq i \leq NM+1\}$ for each peak $p \in [1, P]$. Clearly, this method returns the correction terms that minimize the entropy or equivalently maximize the concentration of the frequency domain of the corrected phase sequence Ψ_c^p, which facilitates the unsupervised correction of the phase errors in Ψ^p incurred by non-uniform radar motion.

C. Motion-Compensated Point Cloud Calibration

Radar motion not only incurs translation and rotation to the radar reference system across frames, but also makes the estimated speed of each point a superimposition of its true speed and the speed of the radar. In other words, radar motion causes distortion to the point cloud from the radar's view, which blurs the semantics of the point cloud. Therefore, we propose a motion-compensated point cloud calibration method which carefully *maps the coordinates of the points in each frame to the same reference system*, and *calibrates the speed of each point by compensating it with the radar's speed*. Specifically, this method takes as input a frame of IF signals and proceeds according to the following steps.

- *Step 1 (Coordinate Calculation).* The method calculates the point-motion angle θ_p of each point p of the input frame by performing FFT over the corrected phase sequence Ψ_c^p, and uses conventional point cloud generation method to obtain its range r_p, radial speed value v_p, and elevation β_p. Based on (θ_p, β_p, r_p), the method calculates the coordinates (x_p, y_p, z_p) of point p.
- *Step 2 (Coordinate Transformation).* Given the translation vector $\mathbf{T}$ and rotation matrix $\mathbf{R}$ of the radar from the initial frame to the current frame, the method calculates the coordinates of each point p in the initial radar reference system as $(x_p^{ini}, y_p^{ini}, z_p^{ini}) = \mathbf{R}(x_p, y_p, z_p) + \mathbf{T}$.
- *Step 3 (Speed Compensation).* The method projects the radar's speed to the radial direction in the current radar reference system, obtains the projected speed value v_{proj}, and subtracts v_{proj} from the radial speed v_p to obtain $v_p' = v_p - v_{proj}$. The method outputs the motion-compensated point cloud with each point p represented by $(x_p^{ini}, y_p^{ini}, z_p^{ini}, v_p')$.

V. Multi-Scale Fusing Gesture Inference

A. Overall Framework

We propose a multi-scale fusing neural network (MSFNN) for gesture inference, as shown in Fig. 2. The input of MSFNN includes both the point cloud and RADM, which have different scales of focuses. That is, the point cloud contains concrete spatial coordinates and speeds, but only focuses on small-scale reflection points; the RADM focuses on a wider scale (i.e., the radar's entire field of view) with the shapes and sizes of its peaks conveying richer gesture semantics than the point cloud, but is inevitably more noisy. To fully exploit such complementary nature of point cloud and RADM, our MSFNN contains a temporal dynamic graph convolutional network (T-DGCN) module and a complex-valued temporal convolutional network (C-TCN) module to extract high-level features from the two modalities respectively, as well as a double cross attention (D-CA) fusion module to fuse the extracted features and infer gesture labels. We provide the details of the three modules in Sec. V-B, V-C, and V-D, respectively.

B. T-DGCN Module

Our design philosophy behind the T-DGCN module is to exploit the inherent dynamic graphical structure of radar point cloud to perform spatial-temporal fusion of the input point features to obtain the high-level representation of gesture semantics. To realize such philosophy, we augment existing edge convolution network by integrating its input with temporal information, obtain a novel temporal edge convolution network (T-EdgeConv), and cascade several T-EdgeConvs as the core of T-DGCN as illustrated in Fig. 4. We describe the detailed design of the T-DGCN module as follows.

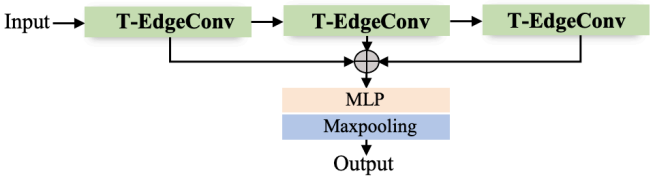

Figure 4: Structure of T-DGCN module with 3 T-EdgeConvs.

The input layer of T-DGCN aggregates all frames of the motion-compensated point clouds of a gesture into an aggregated point cloud, as each single frame is overly sparse to portray gesture semantics. We let $\mathcal{Q} = \{p_1, \cdots, p_Q\}$ denote the aggregated point cloud which contains Q points with each point $p_i = (x_{p_i}, y_{p_i}, z_{p_i}, v_{p_i}, t_{p_i})$ represented by its 5-D feature including its 3-D spatial coordinates $(x_{p_i}, y_{p_i}, z_{p_i})$, radial speed v_{p_i}, and the index t_{p_i} of the frame it belongs to. Then, the input layer constructs a point cloud graph with each node i corresponding to point p_i in Q, and connects a node i and another node j with an edge (i, j), if p_j is among p_i's top-H nearest neighbors in the 5-D feature space where H is a predefined empirically chosen hyper-parameter.

Then, the first T-EdgeConv layer computes the feature $e_{i,j,u}$ of each edge (i, j) in each channel u as

$$e_{i,j,u} = \text{ReLU}\big(\alpha_{m,1}(p_j - p_i) + \alpha_{m,2} p_i\big), \tag{3}$$

where $\alpha_{m,1}$ and $\alpha_{m,2}$ are learnable parameters. Then, the first T-EdgeConv layer updates the feature $p_{i,u}$ of each node i for each channel u as the element-wise maximum among the features of the edges incident to it, i.e.,

$$p_{i,u} = \max_{j:(i,j)\in\mathcal{E}} e_{i,j,u}, \tag{4}$$

where $\mathcal{E}$ is the edge set of the constructed point cloud graph.

To explicitly emphasize temporal information, each subsequent T-EdgeConv layer concatenates the feature of each node i output by its previous layer with the temporal information t_{p_i} to form its input feature, and conducts similar operations as defined by Equation (3) and (4). Note that graph construction is carried out by all T-EdgeConv layers, and the graph constructed by a T-EdgeConv layer is different from that constructed by its previous layer because their input point features are different. The outputs of all cascaded T-EdgeConv layers are concatenated and passed through an MLP and a

max-pooling layer to form the final output of T-DGCN. Our empirical implementation of the T-DGCN module contains 3 T-EdgeConv layer as shown in Fig. 4.

C. C-TCN Module

Our primary design philosophy behind the C-TCN module is to extend traditional TCN [5] to effecitvely extract gesture-related features from the complex-valued elements of the RADM. To realize such philosophy, C-TCN takes as input the RADMs that correspond to all frames of the motion-compensated point clouds of a gesture, and cascades several layers of complex-valued temporal convolution (C-TemConv) as its core. Specifically, C-TemConv introduces a complex-valued convolution kernel with exponentially expanding receptive fields, and carries out causal complex-valued convolution between the kernel and the elements of the RADM within the receptive fields. C-TCN passes the output of the last C-TemConv layer through a max-pooling layer, a flatten layer, and an MLP to obtain its final feature extraction output.

Note that the aforementioned complex-valued convolution kernel helps extract features from the phases of the elements of the input complex-valued RADM, which also convey valuable gesture-related semantics. Such phase features can not be extracted, if one simply uses a real-valued convolution kernel that convolves with the real and imagery part of the input separately. Furthermore, the causality of C-TemConv preserves the causal relationship of the RADM sequence of a gesture, and C-TemConv's dilated receptive field facilitates the processing of long RADM sequences.

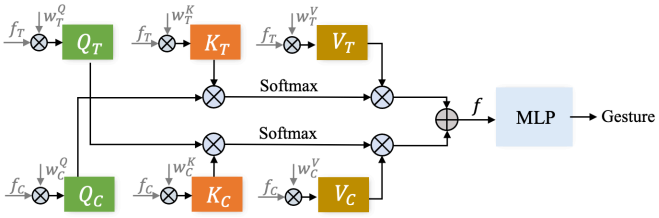

Figure 5: Structure of D-CA fusion module.

D. D-CA Fusion Module

Our primary design philosophy behind the D-CA fusion module is to extend the conventional cross attention [6] fusion structure to a symmetric double cross attention structure as illustrated in Fig. 5 to facilitate the complimentary mutual fusion of the features extracted by T-DGCN and C-TCN. Superficially, we let f_T and f_C denote the outputs of T-DGCN and C-TCN, respectively. The fusion output f is calculated as

$$f = \text{softmax}\left(\frac{Q_T K_C}{\sqrt{d_C}}\right) V_C \oplus \text{softmax}\left(\frac{Q_C K_T}{\sqrt{d_T}}\right) V_T,$$

where d_C and d_T denote respectively the length of the vectors K_C and K_T, $\oplus$ denotes element-wise addition, $Q_T = f_T w_T^Q$, $K_C = f_C w_C^K$, $V_C = f_C w_C^V$, $Q_C = f_C w_C^Q$, $K_T = f_T w_T^K$, and $V_T = f_T w_T^V$ with w_T^Q, w_C^K, w_C^V, w_C^Q, w_T^K, and w_T^V denoting learnable parameters. Then, the D-CA fusion module feeds the fused feature f into an MLP to obtain the ASL gesture recognition result. Clearly, such double cross attention structure enjoys the desirable property of fully exploiting the extracted feature of each modality to enrich that of the other for effective ASL gesture recognition.

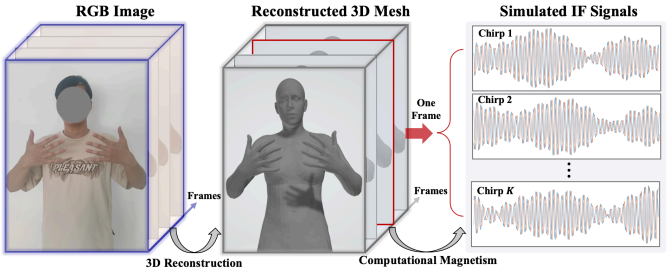

Figure 6: Workflow of our simulation software.

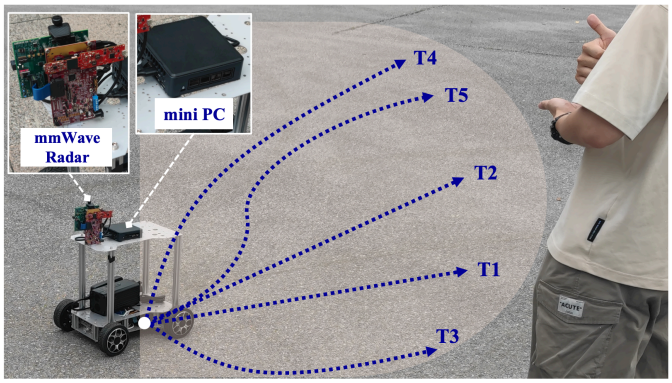

Figure 7: Real-world system and trajectory types.

VI. Implementation and Evaluation

A. Simulation and Real-World System Implementation

Simulation. We build a simulation software to generate a large volume of mmWave data of human gestures under diverse mobile platform trajectories, which are used as the training data of MSFNN. The workflow of the simulation software is shown in Fig. 6 and its key implementation choices are described as follows. The software takes as input the videos of human performing gestures in front of a camera, and converts the 2-D human image in every frame of the video to a 3-D model using a SOTA 3-D reconstruction method [32]. The constructed 3-D human model represents the mesh of a human using a number of stitched small triangle facets. Next, the software computes the mmWave signals that returns to the radar after being reflected by the 3-D human model, when the radar transmits signals at locations along its trajectory based on computational magnetism methods [33]. Finally, the software performs mixing of the transmitted and received signals to obtain the IF signal inputs of m^{3}ASL. Clearly, such simulation-based approach is much more efficient than collecting training data with a real-world system.

Real-World System. To test the performance of m^{3}ASL, we implement a real-world system with a TI IWR6843AOP [30] mmWave radar mounted on the WheelTec [34] mobile robot as shown in Fig. 7. The radar has 3 transmitting antennas and 4 receiving antennas with 60GHz initial frequency, 5mm wavelength, and 3.96GHz bandwidth. We collect the radar's raw IF signals using the DCA1000EVM [35] data acquisition

E1: Open area | E2: Roadside | E3: Corner | E4: Lobby | E5: Room | E6: Corridor

Figure 8: Real-world testing environments (with decreasing openness from E1 to E6).

board. Furthermore, we use an INTEL NUC mini PC [36] to control the data collection of the mmWave radar, and run our motion-aware point cloud generation algorithm and MSFNN to provide in-situ gesture recognition results. The MSFNN model is trained offline on a server equipped with an Intel Xeon E5-2630 v3 CPU and an NVIDIA RTX 2080Ti GPU.

B. Experimental Methodology

Dataset Collection. We describe the key parameters of the simulated and real-world datasets as follows.

- *Gestures*. Both datasets contain data of human performing 30 representative ASL gestures that correspond to the word set {BAD, BUNNY, BUSY, FINE, FINISH, FOOD, FORGET, GO, GOOD, HAPPY, HELLO, HELP, LEFT, LIKE, MORE, NEED, NO, NOT, PLAYGROUND, PLEASE, RIGHT, SAD, SAME, TEMPERATURE, THANKS, TIME, WANT, WEATHER, WRONG, YES}. These ASL gestures cover movements of diverse body parts, such as arms, palms, and fingers.
- *Trajectories*. Both datasets are collected under 5 representative types of mobile platform trajectories as shown in Fig. 8. Specifically, T1 is a straight line with constant speed; T2 is a straight line with accelerating speed; T3 is a clockwise arc with constant angular velocity; T4 is a counter-clockwise arc with constant angular velocity; T5 is a curve whose angular velocity has a constant absolute value but an opposite sign before and after half of the trajectory. Furthermore, the trajectories have $[1.5\text{m}, 5\text{m}]$ initial radar-human distance, $[-60°, 60°]$ initial included angle between the human facing direction and human-to-radar direction, $[0°, 360°]$ initial radar orientation, and $[0.2\text{m/s}, 0.8\text{m/s}]$ radar moving speed. These 4 parameters are abbreviated as distance, angle, orientation, and speed in Fig. 10, as well as captions in Fig. 13-16 for simplification of presentation. Note that the simulated dataset has finer-grained step sizes for these 4 parameters than the real-world datasets due to the convenience to alter the parameters in software.
- *Environments*. As shown in Fig. 8, we select 6 representative environments to test the performance of m^{3}ASL. Specifically, E1 is an open area with no obstacles within 10m around the human; E2 is by the roadside with trees and road signs within 5m behind the human and no obstacles on either side; E3 is an outdoor corner where half wall is within 2m behind the human and the other half is a road; E4 is an indoor lobby where a wall is within 2m behind the human; E5 is an indoor room where walls, tables, and chairs are within 1m behind and beside the human; E6 is a narrow indoor corridor. From E1 to E6, the openness level of the environment reduces and the difference between the simulated and real environment increases.
- *Humans*. We collect data from 2 humans in our simulation software and 7 humans using our real-world system. As shown by our experimental results in Sec. VI-C, 2 humans is enough to generate the training data of MSFNN that ensures satisfactory real-world testing performance.

Baseline Methods. We compare m^{3}ASL with the following 5 baseline methods with the first 2 being generic image and point cloud processing method respectively, and the last 3 being SOTA mmWave-based gesture recognition methods.

- *P3D ResNet (PR)* [37]. This method treats an RADM as a multi-channel image and maps a sequence of multiple input RADMs to a corresponding gesture class.
- *PointNet++ (PN)* [38]. This method aggregates the multi-frame point clouds of a gesture into a single frame, and maps the aggregated point cloud to a gesture class.
- *mmASL* [2]. This method is originally designed to use the spatial spectrograms of mmWave radio as input and employ an extensible multi-task deep learning model to classify ASL gestures. As m^{3}ASL utilizes mmWave radar whose data format is different from that of mmWave radio, we replace the input of this method with the range-angle map (RAM), which has similar semantics as spatial spectrogram.
- *Tesla-Rapture (TR)* [19]. This method uses a message passing graph convolution network to map input radar point cloud to a corresponding gesture class.
- *IndexPen (IP)* [23]. This method adopts a CNN-LSTM concatenated neural network structure to map the RDM and RAM joint input to a corresponding gesture class.

Miscellaneous Details. The radar ego-motion utilized by the motion-aware point cloud generation method of m^{3}ASL is obtained using a SOTA method EmoRI [39]. We train our MSFNN model with 0.0001 learning rate and 50 epochs. Adam optimizer is utilized to optimize model parameters. The running time of executing MSFNN on the mini PC for each testing sample is approximately 8.3ms. By jointly considering the frame duration, platform moving speed, and wavelength of the mmWave signal, we set the range-FFT sequence length during motion-augmented phase sequence construction as $M = 7$ in our experiment. Together with the fact that our TI IWR6843AOP mmWave radar has $N = 12$ VAs, the phase sequence constructed for estimating the point-motion angle has $NM + 1 = 85$ elements.

C. Experimental Results

Simulation Results. We first illustrate the performance of m^{3}ASL using the simulated dataset. From Fig. 9, we can

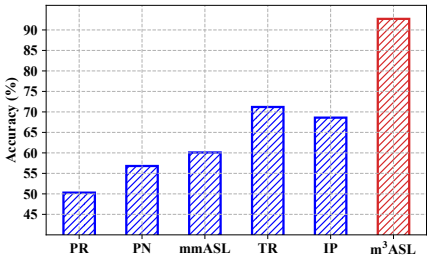

Figure 9: Overall performance in simulation.

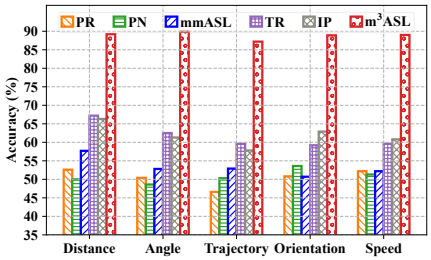

Figure 10: Generalization in simulation.

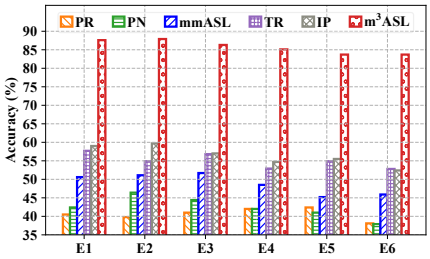

Figure 11: Real-world test w.r.t. environment.

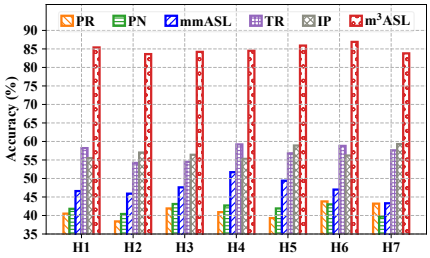

Figure 12: Real-world test w.r.t. human.

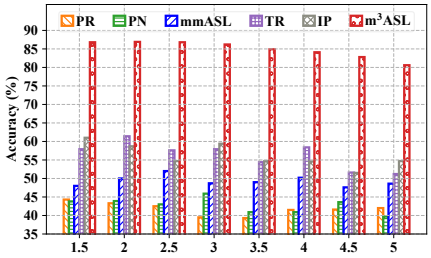

Figure 13: Real-world test w.r.t. distance (m).

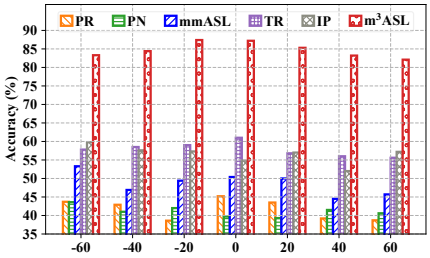

Figure 14: Real-world test w.r.t. angle (°).

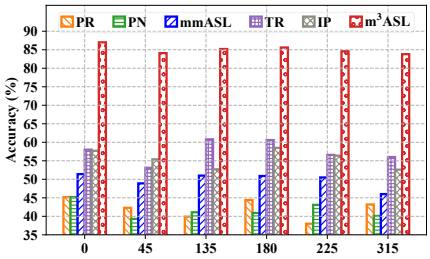

Figure 15: Real-world test w.r.t. orientation (°).

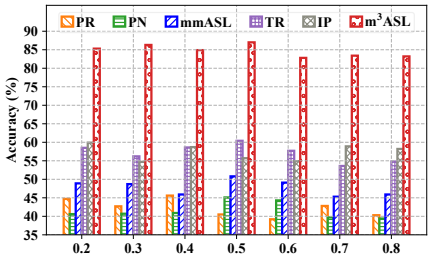

Figure 16: Real-world test w.r.t. speed (m/s).

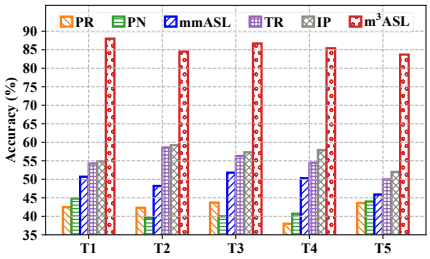

Figure 17: Real-world test w.r.t. trajectory type.

observe that m^{3}ASL could achieve close to 93% recognition accuracy in our experiments, which significantly outperforms those of the baselines. Such result indicates that m^{3}ASL can better extract and exploit the gesture features than baselines when the radar is moving.

As illustrated in Fig. 10, we then evaluate the generalization ability of m^{3}ASL with respect to trajectory type as well as the other 4 trajectory parameters described in Sec. VI-B, including initial radar-human distance, initial included angle between the human facing direction and human-to-radar direction, initial radar orientation, and platform moving speed. In our experiment, we apply the leave-one-out cross-validation method, where one set of data over the above parameters is chosen for testing and the remaining ones are used for training. It can be observed from Fig. 10 that m^{3}ASL exhibits the best generalization performance among all methods, still maintaining an average accuracy of over 87% in settings that are not present in the training dataset.

Real-World Test Results. We illustrate the real-world test performance of m^{3}ASL by directly applying the MSFNN trained with our simulated dataset in real-world tests without any re-training or fine-tuning process. The results to be presented in the rest of this section validate such choice of training with only simulated dataset.

Varying Experimental Settings. We present the results of applying m^{3}ASL in various experiment settings as follows.

In Fig. 11, we illustrate the performance of m^{3}ASL in different real-world environments. From E1 to E6, the difference between the simulated and real-world environment increases, leading to a gradual decline in gesture recognition accuracy. Nevertheless, m^{3}ASL still significantly outperforms all baseline methods, especially in complex and narrow environments. Besides, m^{3}ASL maintains an average accuracy of over 83% even in E6 which has the most complex and narrowest setting among all test environments.

In Fig. 12, we illustrate the performance of m^{3}ASL with respect to 7 different humans (referred to as H1-H7) with different heights, weights, as well as gesture habits. The results in this figure show that m^{3}ASL achieves good generalization ability with respect to different humans.

In Fig. 13-17, we illustrate the performance of m^{3}ASL with respect to different trajectory parameters. It can be observed from Fig. 13 that the gesture recognition accuracy shows a decreasing trend with increasing initial radar-human distance, because longer distance usually leads to weaker received signal strength and thus less gesture-related semantic information in both the point cloud and RADM. Note that when such distance is within 1.5m to 2.5m, m^{3}ASL maintains a gesture recognition accuracy of over 85%.

Fig. 14 illustrates that the gesture recognition accuracy decreases as the initial included angle between the human facing direction and human-to-radar direction increases. This

is because gestures are more likely to be occluded at larger angles and have smaller radar cross-sections. Nevertheless, m^{3}ASL still maintains a relatively stable performance compared with baselines. From Fig. 15, we can observe that m^{3}ASL has the best performance among all methods under all initial radar orientations and 0° and 180° correspond to relatively higher gesture recognition accuracy because of the better views under these two orientations.

Fig. 16 illustrates the performance of m^{3}ASL under different radar moving speeds. On the one hand, smaller speeds lead to fewer elements in the constructed phase sequence, which degrades the angle estimation accuracy. On the other hand, smaller speeds make the radar experience less distortion from its own perspective, leading to better motion-compensated point cloud calibration results. Therefore, both excessive and insufficient radar speeds degrade the gesture recognition performance. It can be observed from Fig. 16 that a velocity of 0.5m/s exhibits the best gesture recognition accuracy. In general, m^{3}ASL outperforms all baseline methods and could achieve a relatively stable and satisfactory gesture recognition accuracy compared with baselines.

In Fig. 17, we illustrate the performance of m^{3}ASL under different trajectory types, from which we can observe that the simplest trajectory (i.e., T1) corresponds to the best gesture recognition performance, and a higher complexity of the trajectory leads to a lower gesture recognition accuracy.

Table I: Results of ablation study.

Ablated Module	Version	Accuracy
Motion-Aware Point Cloud Generation	MAPSC$^-$	57.5%
	SFPSC$^-$	71.8%
	MCPCC$^-$	74.1%
Multi-Scale Fusing Gesture Inference	T-DGCN$^-$	80.1%
	DGCN$^+$	81.4%
	C-TCN$^-$	81.7%
	TCN$^+$	82.6%
	FC$^+$	83.3%
N/A	Original	85.2%

Ablation Study. We conduct ablation study to illustrate the effectiveness of each module in m^{3}ASL. First, we perform ablation on motion-aware point cloud generation, and obtain the following ablated versions of m^{3}ASL, including MAPSC$^-$ by removing motion-augmented phase sequence construction, SFPSC$^-$ by removing self-focused phase sequence correction, and MCPCC$^-$ by removing motion-compensated point cloud calibration. The experimental results are shown in Table I. It can be observed that motion-augmented phase sequence construction plays a crucial role, as it enhances the angle estimation accuracy. Moreover, self-focused phase sequence correction contributes notably to the overall performance, as it helps reduce the angle estimation errors caused by inaccurate radar motion estimation. Additionally, motion-compensated point cloud calibration is crucial as well, because it mitigates the distortion of point clouds in the radar coordinate system due to the motion of the radar.

Then, we perform ablation on multi-scale fusing gesture inference, and obtain the following ablated versions of m^{3}ASL, including T-DGCN$^-$ by removing T-DGCN, DGCN$^+$ by replacing T-DGCN with DGCN, C-TCN$^-$ by removing C-TCN, TCN$^+$ by replacing C-TCN with real-valued TCN that adopts a real-valued convolutional kernel and treats the real and imagery part of each element in RADM as separate channels, as well as FC$^+$ by replacing the double cross attention structure in the D-CA fusion module with a simple fully connected layer. The results are shown in Table I. It can be observed that T-DGCN$^-$ and C-TCN$^-$ both perform significantly worse than the original version. Such results indicate that both point cloud and RADM are indispensable input modalities. The performance degradation of DGCN$^+$ and TCN$^+$ compared with the original version indicates that the temporal feature captured by T-DGCN and phase-related feature captured by the complex-valued convolution in C-TCN also play a positive role in boosting gesture recognition accuracy. Finally, the double cross attention structure also provides positive assistance, because it enables mutual enhancement of the features extracted by T-DGCN and C-TCN.

VII. Conclusions and Discussions

This paper proposes m^{3}ASL, a method that achieves ASL recognition with moving mmWave radar. m^{3}ASL contains a motion-aware point cloud generation component that generates accurate radar point cloud by avoiding the cons and exploiting the pros of radar motion, and a multi-scale fusing gesture inference component that extracts features from multi-scale radar data accordingly and complementarily fuses the extracted features to infer ASL gestures. Experimental results demonstrate that m^{3}ASL achieves close to 93% recognition accuracy on 30 representative ASL gestures, surpassing SOTA methods by at least 30%. In what follows, we discuss several promising directions of extending m^{3}ASL.

Applicability on Other Sign Languages. Although the current version of m^{3}ASL only recognizes ASL, we make an educated guess that m^{3}ASL can be applied to recognize other sign languages (e.g., ISL, BSL), as long as we retrain MSFNN with data of the corresponding type of sign language. Our rationale is that the gestures of different sign languages share similar scale and granularity of body movement though different in details. We would empirically validate the applicability of m^{3}ASL on other sign languages in our future work.

Feasibility of Fusing mmWave with Other Modalities. There exist recent works that fuse RF signals with other data modalities (e.g., vision [40], acoustics [41]) for various human sensing tasks by exploiting the benefits of different modalities. In our future work, we would also like to investigate whether it is feasible to fuse mmWave with data of other modalities that can be conveniently collected by mobile robots to improve the accuracy of ASL gesture recognition.

VIII. Acknowledgement

This work was supported by NSF China (No. U21A20519, 62372288, 62202298).

REFERENCES

[1] R. E. Mitchell and T. A. Young, "How Many People Use Sign Language? A National Health Survey-Based Estimate," *The Journal of Deaf Studies and Deaf Education*, vol. 28, no. 1, pp. 1–6, 2022.

[2] 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," in *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 2020.

[3] M. A. Richards, *Fundamentals of radar signal processing*. Mcgraw-hill New York, 2005.

[4] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics*, vol. 38, no. 5, pp. 1–12, 2019.

[5] S. Fang, Q. Zhang, G. Meng, S. Xiang, and C. Pan, "Gstnet: Global spatial-temporal network for traffic flow prediction," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.

[6] X. Wei, T. Zhang, Y. Li, Y. Zhang, and F. Wu, "Multi-modality cross attention network for image and sentence matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[7] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, vol. 43, pp. 1–54, 2015.

[8] J. Liu, D. Li, L. Wang, F. Zhang, and J. Xiong, "Enabling contact-free acoustic sensing under device motion," in *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 2022.

[9] S. Cao, D. Li, S. I. Lee, and J. Xiong, "Powerphone: Unleashing the acoustic sensing capability of smartphones," in *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2023.

[10] Y. Wang, J. Shen, and Y. Zheng, "Push the limit of acoustic gesture recognition," *IEEE Transactions on Mobile Computing*, vol. 21, no. 5, pp. 1798–1811, 2022.

[11] Y. Zou, Y. Wang, H. Dong, Y. Wang, Y. He, and K. Wu, "Pregesnet: Few-shot acoustic gesture recognition based on task-adaptive pretrained networks," *IEEE Transactions on Mobile Computing*, 2024.

[12] Z. Yang, Y. Zhang, K. Qian, and C. Wu, "Slnet: A spectrogram learning neural network for deep wireless sensing," in *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2023.

[13] L. Zhao, R. Xiao, L. Jianwei, and J. Han, "One is enough: Enabling one-shot device-free gesture recognition with cots wifi," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, 2024.

[14] R. H. Venkatnarayan, G. Page, and M. Shahzad, "Multi-user gesture recognition using wifi," in *Proceedings of the Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2018.

[15] J. Hu, T. Zheng, Z. Chen, H. Wang, and J. Luo, "Muse-fi: Contactless muti-person sensing exploiting near-field wi-fi channel variation," in *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2023.

[16] A. Virmani and M. Shahzad, "Position and orientation agnostic gesture recognition using wifi," in *Proceedings of the Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2017.

[17] 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," in *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 2020.

[18] 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," in *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 2022.

[19] D. Salami, R. Hasibi, S. Palipana, P. Popovski, T. Michoel, and S. Sigg, "Tesla-rapture: A lightweight gesture recognition system from mmwave radar sparse point clouds," *IEEE Transactions on Mobile Computing*, vol. 22, no. 8, pp. 4946–4960, 2022.

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

[21] C. Wu, X. Huang, J. Huang, and G. Xing, "Enabling ubiquitous wifi sensing with beamforming reports," in *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication (SIGCOMM)*, 2023.

[22] 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*, vol. 35, no. 4, 2016.

[23] H. Wei, Z. Li, A. D. Galvan, Z. Su, X. Zhang, K. Pahlavan, and E. T. Solovey, "Indexpen: Two-finger text input with millimeter-wave radar," in *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 2022.

[24] J. Ma, Z. Chang, F. Zhang, J. Xiong, B. Jin, and D. Zhang, "Mobi2sense: enabling wireless sensing under device motions," in *Proceedings of the Annual International Conference on Mobile Computing And Networking (MobiCom)*, 2022.

[25] Z. Chang, F. Zhang, J. Xiong, W. Chen, and D. Zhang, "Msense: Boosting wireless sensing capability under motion interference," in *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2024.

[26] Z. Zhu, X. Wang, A. Kapoor, Z. Zhang, T. Pan, and Z. Yu, "Eis: A wearable device for epidermal american sign language recognition," in *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 2018.

[27] B. Xie, D. Ganesan, and J. Xiong, "Embracing lora sensing with device mobility," in *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2023.

[28] S. Zhang, T. Zheng, Z. Chen, and J. Luo, "Can we obtain fine-grained heartbeat waveform via contact-free rf-sensing?" in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, 2022.

[29] X. Zhang, Y. Zhang, Z. Shi, and T. Gu, "mmfer: Millimetre-wave radar based facial expression recognition for multimedia iot applications," in *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2023.

[30] "Iwr6843," https://www.ti.com/product/IWR6843AOP.

[31] R. L. Morrison, M. N. Do, and D. C. Munson, "Sar image autofocus by sharpness optimization: A theoretical study," *IEEE Transactions on Image Processing*, vol. 16, no. 9, pp. 2309–2321, 2007.

[32] S. Goel, G. Pavlakos, J. Rajasegaran, A. Kanazawa, and J. Malik, "Humans in 4d: Reconstructing and tracking humans with transformers," in *Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[33] H. Xue, Q. Cao, C. Miao, Y. Ju, H. Hu, A. Zhang, and L. Su, "Towards generalized mmwave-based human pose estimation through signal augmentation," in *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2023.

[34] "Wheeltec mobile robot," https://wheeltec.jd.com/.

[35] "Dca1000evm," https://www.ti.com/tool/DCA1000EVM/.

[36] "Intel nuc mini pc," https://ark.intel.com/content/www/us/en/products.

[37] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[38] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2017.

[39] R. Ding, H. Jin, J. Ding, X. Wang, G. Fan, F. Zhu, X. Tian, and L. Kong, "Push the limit of single-chip mmwave radar-based egomotion estimation with moving objects in fov," in *Proceedings of the Conference on Embedded Networked Sensor Systems (SenSys)*, 2023.

[40] S. Zhang, T. Zheng, Z. Chen, J. Hu, A. Khamis, J. Liu, and J. Luo, "Ochid-fi: Occlusion-robust hand pose estimation in 3d via rf-vision," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[41] M. Li and W. Wang, "Hybrid zone: Bridging acoustic and wi-fi for enhanced gesture recognition," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, 2024.