Facilitating Radar-Based Gesture Recognition With Self-Supervised Learning

Zhiyao Sheng[1], Huatao Xu[2], Qian Zhang[1], Dong Wang[1]
[1] Shanghai Jiao Tong University, Shanghai, China
[2] Nanyang Technological University, Singapore, Singapore
szyjadey@gmail.com, huatao001@e.ntu.edu.sg, {qwert3472, wangdong}@sjtu.edu.cn

***Abstract*—With deep learning, millimeter-wave radar-based gesture recognition applications have achieved satisfactory results. However, most existing approaches highly rely on high-quality labeled data, and they suffer from severe over-fitting when labeled data are scarce. To end this, we present RadarAE, a novel representation learning framework for radar sensing applications. RadarAE learns sophisticated representations from massive low-cost unlabeled radar data, which enables accurate gesture recognition with few labeled data. To achieve this goal, we first meticulously observe the characteristics of raw radar data and extract an effective feature, Spatio-Temporal Motion Map (STMM). Then we borrow the key principle of Masked Autoencoders (MAE), a self-supervised learning technique for images, and propose an MAE-like model to learn useful representations from STMM. To adapt RadarAE to radar sensing applications, we present a series of customization techniques, including data augmentation, optimized model structure, and adaptive pre-training method. With the learned high-level representations, gesture recognition models can achieve superior performance in few-shot scenarios. Experiment results show that our model can achieve 79.1%, 92.1%, 97.8%, and 99.5% recognition accuracy in the 1, 2, 4, and 8-shot scenarios, respectively, where x-shot refers to the number of labeled samples for each gesture type. The source codes and dataset are made publicly available[1].**

***Index Terms*—Human-Computer Interaction, Millimeter-Wave Radar, Gesture recognition, Self-Supervised Learning**

I. Introduction

Gesture recognition is a critical technology for many Human-Computer Interaction (HCI) applications. Traditional gesture recognition solutions are mainly based on cameras [1], [2], wearable sensors [3]–[5]. Although camera-based solutions can obtain good performance, images are likely to contain sensitive information, leading to privacy concerns [6]. Wearable sensor-based technologies require users to wear or hold certain sensors, which are not user-friendly. In comparison, millimeter-wave radar-based methods facilitate device-free sensing with no privacy concern and have attracted much attention [7]–[12].

With the development of deep learning, many radar-based works utilize deep learning models to process data [7]–[12] and achieve high performance in certain cases. However, most of these methods require to collect and annotate a large number of training data, which is very costly. To this end, we adopt the key principle of self-supervised learning [13]–[16] and propose a novel representations learning framework, RadarAE. Instead of collecting large amounts of labeled data, RadarAE learns to extract high-level representations from radar data through a pre-training task performed on an unlabeled dataset. Compared with labeled data, unlabeled data are much easier to collect because users are not constrained to perform specific gestures during the collection process. In addition, millimeter-wave radars are easy to deploy and do not raise privacy concerns, making them perfect for collecting massive unlabeled data. For instance, we can install the radar in a gym to collect unlabeled data for workout activity recognition automatically when people do exercises. The downstream gesture classifier models can directly take the extracted representations as input to estimate the final results and achieve impressive performance with a small number of labeled data.

To introduce the principle of self-supervised learning into radar sensing, the first challenge comes from the huge data discrepancy between radar data and other data types adopted by traditional self-supervised learning approaches. Existing self-supervised learning techniques focus on processing text, videos, images, etc. [13]–[16], which are informative and embedded with abundant contextual relations. Different from them, many commercial radars devices transmit frequency modulated continuous wave (FMCW) signals, from which we can obtain Range-Doppler Map (RDM) and Range-Angle Map (RAM) that reveal reflectors' positions and velocities. However, RDM and RAM are information-sparse and lack temporal relations, which are different from text or images. As a result, existing self-supervised approaches can not be applied to radar data.

To address this challenge, we rethink the input of radar sensing models and propose a new feature based on RDM and RAM. By locating the user and cropping the user-related part from RDM and RAM, we effectively eliminate position-related patterns of radar data. Subsequently, we aggregate the information about velocities and directions of reflectors embedded in sequences of cropped RDM and RAM to form an informative and compact feature, Spatio-Temporal Motion Map (STMM), which is an image-like 3-D tensor.

Our framework's design is therefore motivated by learning useful representations from STMM. By meticulously observing the properties of STMM, we find that STMM resembles image data because it carries rich contextual relations along the time and distance domain. To this end, we treat STMM as images and borrow the key principle of MAE [14] to process

[1]https://github.com/Ela-Boska/RadarAE

STMM, which is an emerging self-supervised learning model in computer vision. However, due to the data discrepancy, the original MAE model fails to work with STMM. This paper thus devises a set of techniques to adapt RadarAE to radar sensing applications. Specifically, we adjust the pre-training strategy (see Section III-C1) to make full use of spatio-temporal relations in STMM. In addition, we propose a lightweight model structure together with the weight-sharing mechanism, which eases over-fitting caused by limited training data. Furthermore, we design an efficient data augmentation approach to enrich the diversity of the collected dataset without destructing the physical meaning of STMM.

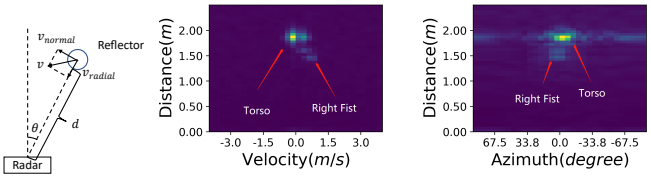
To show the effectiveness of RadarAE in enhancing the performance of radar sensing models with few labeled samples, we collect a dataset for the gesture recognition task and conduct extensive experiments. RadarAE achieves 79.1% recognition accuracy in the one-shot scenario and consistently outperforms other supervised and self-supervised learning approaches in all few-shot scenarios. Our main contributions are summarized as follows:

1) To the best of our knowledge, RadarAE is the first work that introduces the principle of self-supervised learning into radar sensing. Compared with previous works, RadarAE requires much less labeled data, which is a concrete step toward practical radar-based gesture recognition systems. In addition, the methodology of RadarAE is also applicable in wireless sensing applications based on other signals embedded with rich spatio-temporal features (e.g., Wi-Fi [17] , RFID [18] and ultrasonic wave [19]).
2) We design a compact and informative feature STMM from raw radar data and present a powerful representation learning framework based on STMM. Furthermore, a series of improvements in pre-training strategy, model structure, and data augmentation are proposed to make RadarAE adapt to radar data.
3) We implement a prototype system of RadarAE and evaluate it with massive experiments. The results show that the RadarAE-powered model significantly outperforms state-of-the-art techniques. The source codes and dataset are made available to the public.

II. Background

A. Radar Data Processing

We implement RadaAE with IWR1642 from Texas Instruments, a millimeter-wave radar that transmits FMCW signals. The frequency of FMCW signals changes periodically, and each period is defined as a chirp. For a single receiving antenna, each frame contains multiple chirps. We can infer the reflectors' distances, radial velocities, and azimuths (defined in Fig. 1a) by performing FFT over each chirp, chirps in a frame, and frames of multiple receiving antennas [20], which are called Range FFT, Doppler FFT, and Angle FFT in this paper, respectively. By combining Range FFT and Doppler FFT, we obtain the RDM which reflects the distances and velocities of



(a) Setup (b) RDM (c) RAM

Fig. 1: (a) The definition of the azimuth, radial velocity and distance, which are denoted as θ, v_{radial} and d, respectively. (b) a RDM sample (c) a RAM sample.

reflectors. Similarly, we obtain RAM by performing Range FFT and Angle FFT, which contains information on the distances and directions of reflectors. The size of RDM and RAM is $[N_D, N_R]$ and $[N_A, N_R]$ where N_D, N_A, and N_R represent the lengths of transformed axes after Doppler, Angle, and Range FFT, respectively. In our case, N_D, N_A and N_R equal to 32, 32 and 60.

Fig. 1 illustrates two examples of RDM and RAM when the user pulls the fist back. For RDM, x-axis (velocity axis) and y-axis (distance axis) represent the radial velocity and the distance. When a reflector reflects the radar signal, it would raise a response at the corresponding velocity index and distance index in RDM. Similarly, the x-axis (angle axis) and y-axis (distance axis) of RAM represent the azimuth and the distance. A reflector would induce a response at the corresponding angle index and distance index in RAM. For RDM and RAM, the amplitude of the response represents the intensity of reflected signals. Specifically, the angle index of i corresponds to the azimuth of $(i - N_A/2)\Delta A$, whereas the velocity index of j corresponds to the radial velocity of $(j - N_D/2)\Delta V$; the distance index of k corresponds to the distance of $k\Delta R$. The angle resolution ΔA, velocity resolution ΔV and distance resolution ΔR are some configurable parameters of the FFT.

In Fig. 1, we can distinguish the torso and the right fist of the user. The right fist is on the right side of the torso and has a positive radial velocity because it is being pulled back. Although RDM and RAM contain information about reflectors' motion states and positions, the features of RDM and RAM are very sparse.

B. Self-supervised Learning

Deep learning techniques have demonstrated their effectiveness in numerous applications. However, establishing a successful deep learning model depends on plenty of labeled data. Some pioneer works [13]–[15] propose self-supervised learning approaches to learn general representations from low-cost unlabeled images, videos, and texts. These works suggest that a wide variety of downstream tasks can benefit from the learned representations. One common methodology of self-supervised learning techniques is reconstructing a proportion of the input from the other part, which enables generating labels from data themselves. MAE [14] is an effective self-supervised learning model processing images. It first separates an image into a sequence of "patches". Then it masks

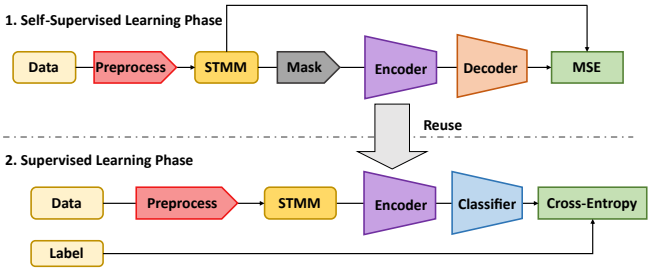


Fig. 2: RadarAE overview.

some patches and trains a deep learning model to recover masked patches from the left ones. Compared with other self-supervised learning techniques (e.g., BERT [13]), MAE is capable of utilizing contextual relations along two dimensions, which provide a feasible solution to extract the rich contextual relations of radar data in space and time domains.

III. RadarAE Design

A. Overview

Fig. 2 illustrates the overview of RadarAE. RadarAE consists of two phases: the self-supervised learning and supervised learning phases. RadarAE first extracts the informative image-like features (i.e., STMM), from raw data. In the self-supervised learning phase, we mask a portion of STMM input and feed them to a pre-training model composed of an encoder and a decoder. The pre-training model is trained to reconstruct the masked parts from the other parts. Through the reconstruction task, the encoder gains comprehensive understanding (i.e., rich spatio-temporal relations embedded in STMM) of STMM and generates high-level representations from unlabeled data.

In the supervised learning phase, we make use of the pre-trained RadarAE encoder to process labeled data. Then a gesture classifier is trained to recognize gesture types based on the extracted representations. By leveraging prior knowledge learned in the self-supervised learning phase, the downstream models (i.e. the gesture classifier) can achieve superior performance with a small amount of labeled data.

B. Preprocessing

As shown in Fig. 1, RDM and RAM are pretty sparse that most elements are close to zero, which carries little information about user activities. Moreover, they do not contain the essential temporal relations for gesture recognition. Driven by this analysis, we propose a compact and informative feature, STMM. We observe that the pixels with high values caused by the user motion gather around the position of the target user. Thus, we first find the distance and direction of the user by selecting the location with the most significant response in RAM. However, the response intensities within RAM fluctuate significantly, leading to inconsistent results. Therefore, we estimate the user position based on the average intensities of the first N_{pre} (i.e. 40) frames of RAM of a data sample:

$$i_u = \arg\max_i \sum_{t=1}^{N_{pre}} \sum_{j=1}^{N_R} RAM_{i,j}^{(t)} \tag{1}$$

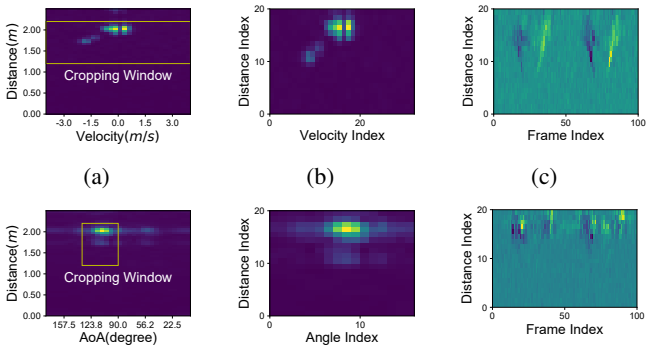


Fig. 3: The procedures to extract RA and RV. (a) RDM; (b) HRDM; (c) 100 frames of continuous RV; (d) RAM; (e) HRAM; (f) 100 frames of continuous RA.

$$j_u = \arg\max_j \sum_{t=1}^{N_{pre}} \sum_{i=1}^{N_A} RAM_{i,j}^{(t)} \tag{2}$$

Where i_u and j_u are the angle and distance index of the target user. We then crop the user-related responses from the original RDM and RAM via windows with the size of $N_D \times Y$ and $W_A \times Y$ to obtain the human-centered RDM (HRDM) and human-centered RAM (HRAM), as shown in Fig. 3. The cropping area in RDM is $\{RDM_{i,j} | j \in [j_u - 0.8Y, j_u + 0.2Y)\}$ and the cropping area in RAM is $\{RAM_{i,j} | i \in [i_u - 0.5W_A, i_u + 0.5W_A), j \in [j_u - 0.8Y, j_u + 0.2Y)\}$ where i_u and j_u are the angle index and distance index of the user. The W_A and Y are two configurable parameters. The cropped distance indices range from $[j_u - 0.8Y, j_u + 0.2Y)$ because users make gestures with their hands in front of their body. The HRDM and HRAM can adapt to different user locations and avoid the noised responses induced by other objects.

Based on HRDM and HRAM, we compute the overall velocity and azimuth at each distance index by aggregating all corresponding velocities and azimuths. We name the two processed features Range-wise Velocity (RV) and Range-wise Angle (RA) in our paper, which are defined as follows:

$$RV_j^{(t)} = \frac{\sum_{i=1}^{N_D} v_i HRDM_{i,j}^{(t)}}{c_1 + \sum_{i=1}^{N_D} HRDM_{i,j}^{(t)}} \tag{3}$$

$$RA_j^{(t)} = \frac{\sum_{i=1}^{W_A} \theta_i HRAM_{i,j}^{(t)}}{c_2 + \sum_{i=1}^{W_A} HRAM_{i,j}^{(t)}} \tag{4}$$

where RV, $RA \in \mathbb{R}^Y$, and c_1, c_2 are two constants to ensure that RV_j and RA_j are close to 0 when there is no reflector at the distance index j (i.e., $\sum_{i=1}^{W_A} HRAM_{(i,j)}^{(t)} \to 0$ and $\sum_{i=1}^{N_D} HRDM_{i,j}^{(t)} \to 0$). v_i and θ_i are the radial velocity with velocity index of i and azimuth with angle index of i (with respect to the target user), which are formulated as

$$v_i = (i - N_D/2)\Delta V, \quad \theta_i = (i - W_A/2)\Delta A$$

We then concatenate RV and RA of T continuous frames and create a feature with the size of $T \times Y \times 2$, namely

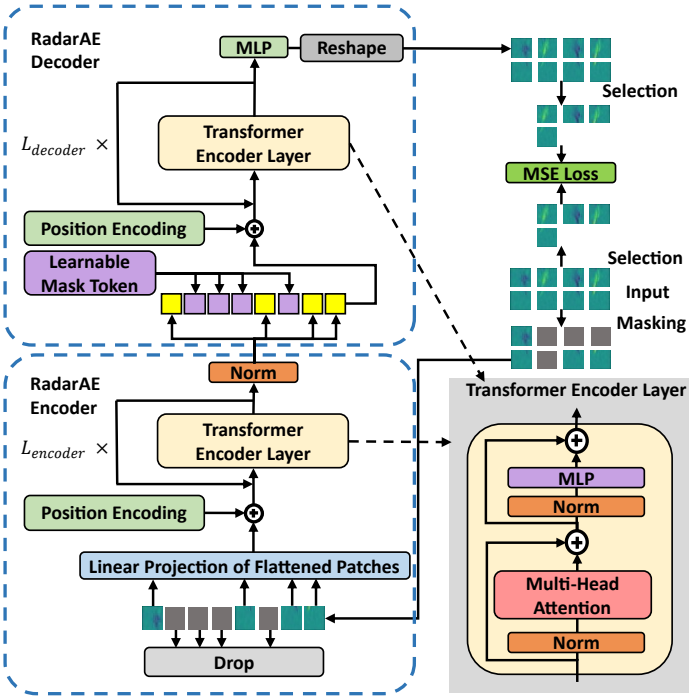


Fig. 4: Workflow of self-supervised learning.

Spatio Temporal Motion Map (STMM). The three dimensions of STMM represent the dimension of time, the dimension of distance, and the dimension of RV and RA. If we interpret STMM as an image with only two channels, its first and second channel consists of T continuous frames of RV and RA, as Fig. 3c and 3f show. If a sample contains l frames ($l < T$), we pad $T - l$ frames with zeros to the first dimension and have the unified size of STMM. Compared with raw radar data, STMM integrates the information of both velocities and azimuths at different times and distances, which are more compact and beneficial to representation learning of the follow-up models.

C. Self-Supervised Learning Phase

To learn the underlying patterns of radar data, we propose to make full use of unlabeled data with a pre-training model. Different from the text data which only contains temporal relations, STMM is a 2D image-like feature embedded with rich spatio-temporal relations. Thus, we treat STMM as images and borrow the training approach of MAE [14]. As shown in Fig. 4, some random patches of the input data are masked and the encoder and decoder learn to reconstruct the masked data from the visible ones. With such a reconstruction task, the models are trained to capture the relations between the visible patches at different times and distances.

1) Masking: First, $STMM \in \mathbb{R}^{T\times Y\times 2}$ is divided into a sequence of flattened patches $X \in \mathbb{R}^{N_P\times H_{patch}}$ where $N_P = 2TY/H_{patch}, H_{patch} = P_T \times P_Y \times 2$ and $P_T \times P_Y$ represents the patch size. After that, $r_m N_P$ randomly selected patches are masked where r_m is called the masking ratio. In RadarAE, we choose to mask a high proportion of patches to create a challenging training task. The models can infer the missed patches by simply copying visible patches if a small masking ratio is used, which undermines the quality of learned features. In contrast, the models are forced to learn the spatio-temporal relations with a high masking ratio.

Since the STMM are different from images, we customize the masking strategy to make it work better in our scenario. Specifically, we reduce the patch size from 16×16 to 3×10 for the following two reasons. First, the size of STMM (72×20) is much smaller than that of images (typically 224×224). RadarAE would fail to learn the inter-patch correlations if a large patch size is used as STMM is partitioned into few patches. Second, as STMM depicts the movements of the target user, the inter-time relations in STMM contain more information than the inter-distance relations. Therefore, we set the patch size along the time axis to be smaller than that along the distance axis for better extraction of inter-time relations.

2) Pre-training Model: The structure of the pre-training model is illustrated in Fig. 4. First, each patch is mapped into H_e dimensions with a linear projection. subsequently, positional encoding [21] with the size of $N_p \times H_e$ is added to the input sequence, which introduces information about the patches' positions in STMM. This is because the transformer encoder layer has no prior knowledge about the position of each patch. After that, the masked patches are directly discarded. The processed features $\mathbf{Z}_0$ are formulated as:

$$\mathbf{Z}_0 = \text{Discard}(XW + PE), \quad \mathbf{Z}_0 \in \mathbb{R}^{(1-r_m)N_p\times H_e}$$
$$W \in \mathbb{R}^{H_{patch}\times H_e}, \quad PE \in \mathbb{R}^{N_p\times H_e} \tag{5}$$

The processed features $\mathbf{Z}_0$ are subsequently fed into $L_{encoder}$ stacked transformer encoder layers [21]. The transformer encoder layer is composed of a multiheaded self-attention layer (MSA) [21] and a MLP block. The MSA effectively extracts the contextual relations of the input sequence with position encoding. The MLP contains two fully-connected layers with one GELU activation layer [22]. The output size of the MLP is identical to H_e while the hidden size is $4H_e$. Before the MSA and MLP, we normalize the feature with Layer Normalization (LN) [23]. In addition, a residual connection is appended after each MSA and MLP. The processing of the l-th transformer encoder layer can be expressed as:

$$\mathbf{a}_l = \text{MSA}(\text{LN}(\mathbf{Z}_{l-1})) + \mathbf{Z}_{l-1} \tag{6}$$

$$\mathbf{Z}_l = \text{MLP}(\text{LN}(\mathbf{a}_l)) + \mathbf{a}_l \tag{7}$$

After the last transformer encoder layer, LN is applied to produce the extracted representations y:

$$y = \text{LN}(\mathbf{Z}_{L_{encoder}}) \tag{8}$$

3) Reconstruction: We utilize a decoder to reconstruct masked STMM from the extracted representations. Different from the encoder, the decoder replaces the discarded patches with a learnable mask token (see Fig. 4). It contains $L_{decoder}$ transformer encoder layers with embedding size H_d. If $H_d \neq H_e$, a linear projection is necessary before feeding the extracted representations into the decoder. After the transformer

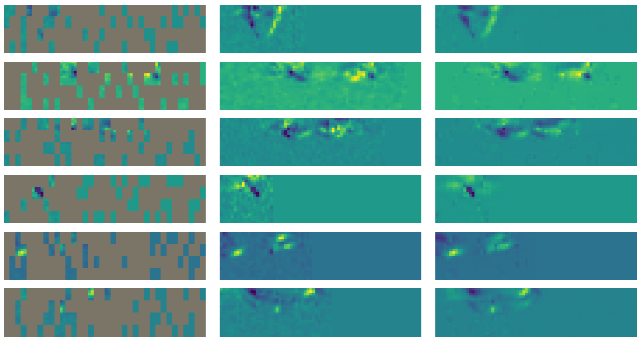


Fig. 5: Some reconstruction results with data from the testing set. We only show the first channel of STMM for illustration. In each row, we plot the masked input (left), original input (middle), and reconstructed input (right).

encoder layers, an MLP with a reshape operation outputs the reconstructed STMM X' from the processed features. To measure the difference between the missing patches and reconstructed patches, we adopt the Mean Square Error (MSE) function to calculate the pre-training loss:

$$loss = \text{MSE}(\text{Select}(X), \text{Select}(X')) \tag{9}$$

where Select represents selecting the patches at the masked positions as Fig. 4 shows.

Some reconstruction examples are given in Fig. 5. RadarAE shows an impressive capacity for reconstructing masked patches. Despite the fact that the majority of patches are invisible for RadarAE, the reconstructed inputs have similar patterns to the ground truth. These reconstruction results verify that RadarAE develops a sophisticated understanding of radar data via self-supervised learning.

D. Supervised Learning Phase

In the supervised learning phase, RadarAE utilizes the learned encoder to facilitate gesture recognition. The workflow of supervised learning is demonstrated in Fig. 6. This procedure trains a classifier model which learns to map the extracted representations to target labels. We input the STMM into the pre-trained RadarAE encoder to extract general representations without masking (i.e., the "Discard" operation in (5) is removed). A classifier model then processes the representations and determines the user gesture of the input data. The cross-entropy loss function is applied to optimize the classifier model. We design a classifier composed of two transformer encoder layers and two fully-connected layers. The embedding size of the transformer encoder layers is equal to H_e. After the transformer encoder layers, the processed features of different patches are concatenated and fed to a two-layer MLP to predict the gesture type.

E. Data Augmentation

For the radar sensing scenario, we devise a customized data augmentation method, namely *random temporal cropping*

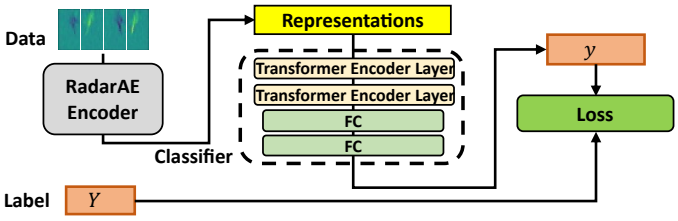


Fig. 6: Workflow of supervised learning.

Algorithm 1 RTC

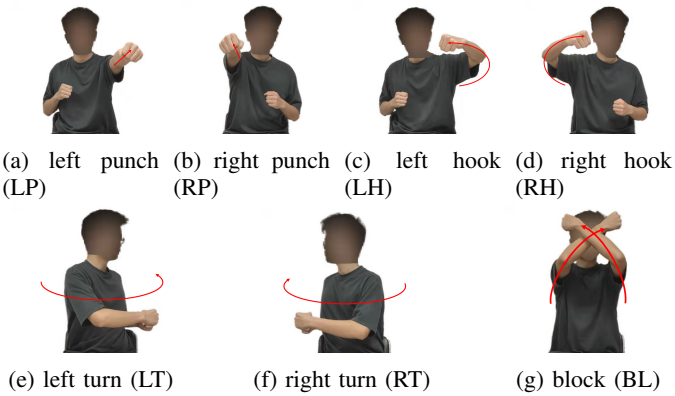
Input: STMM $X \in \mathbb{R}^{T \times Y \times 2}$, number of valid frames l
Output: STMM after data augmentation $X' \in \mathbb{R}^{T \times Y \times 2}$

1: $l' = Uniform[1, l]$;
2: $start = Uniform[0, l - l']$
3: $X' = Zeros([T, Y, 2])$
4: **for** $i = 1, ..., l'$ **do**
5: $\quad X'_i = X_{i+start}$;
6: **end for**

(RTC). The details of RTC are demonstrated in Algorithm 1. This algorithm randomly crops a segment of the original STMM sample to form a new STMM sample. The $Uniform[a, b]$ in line 1 and line 2 represents the uniform discrete distribution with the upper bounding and lower bounding set to a and b. The l' and $start$ stand for the length and starting index of the cropping. The $Zeros(size)$ in line 3 is a function that returns a tensor of the specified size that is filled with zeros. The codes in lines 4-6 copy the cropped segment to the created tensor. Zero-padding is adopted to neutralize the reduction in length caused by cropping. RTC is computationally cheap and can be applied online during training to produce distinct input in each epoch, which greatly enhances the generalizability of RadarAE.

F. Lightweight Model and Weight-Sharing

There are 12 and 4 transformer encoder layers in the original MAE's encoder and decoder, respectively. As a result, the MAE model is composed of 93 million parameters. Such a heavy model can not accommodate to radar sensing due to the scarce training data and limited computational capacity of edge devices. Thus, we carefully design a lightweight model with smaller linear projection dimensions ($H_e = 300$, $H_d = 150$) and less transformer encoder layers ($L_{encoder} = 3$, $L_{decoder} = 2$). In addition, we apply the cross-layer weight-sharing mechanism in the pre-training model. In each sub-model (i.e., encoder and decoder), the multiple transformer encoder layers share the same parameters, which significantly reduces the number of trainable parameters of RadarAE. The number of trainable parameters of our pre-training model is only 2.6 million, which is only 3% of that of MAE. This lightweight model structure can avoid over-fitting and is affordable for resource-constrained devices.



(a) left punch (LP) (b) right punch (RP) (c) left hook (LH) (d) right hook (RH)

(e) left turn (LT) (f) right turn (RT) (g) block (BL)

Fig. 7: The gesture set.

IV. Implementation

A. Device and Data Processing Configuration

We collect data using IWR1642, a COTS millimeter-wave radar device with 2 transmitting antennas and 4 receiving antennas, which yields 8 virtual antennas [24]. The radar is configured to work with a frame rate of 25 Hz. We adopt an angle resolution (ΔA) of $\pi/32$, a velocity resolution (ΔV) of $0.254m/s$ and a distance resolution (ΔR) of $0.05m$. The range of azimuth, velocity and distance are set to $[-\pi/2, \pi/2]$, $[-4.06m/s, 4.06m/s]$ and $[0m, 3m]$, respectively. The size of HRDM and HRAM are set to 32×20 and 8×20 (i.e., $W_A = 8, Y = 20$). The number of frames in STMM Y is set to 72, resulting in STMM with the size of $72 \times 20 \times 2$.

B. Data Collection

RadarAE is evaluated with a gesture recognition task. We recruit 8 volunteers (4 males and 4 females) to collect radar data. One feasible scenario of gesture recognition is somatosensory games. Thus, we design 7 gestures typically used in somatosensory games, as shown in Fig. 7. All participants are required to collect 16-20 data samples for each gesture. During data collection, volunteers conduct gestures toward the radar in a 1.2m×1.2m area whose center is 1.8m in front of the radar. The whole dataset contains 1094 data samples.

C. Models for Comparison

We implement a series of baseline models for comparison. By default, the models take STMM as input.

RadarBERT is a comparing model adopting the methodology of another self-supervised learning technique, Bidirectional Encoder Representations from Transformers (BERT) [13]. Compared with RadarAE, RadarBERT employs a trivial decoder (MLP) and randomly masks 15% of frames in STMM (11 out of 72). Besides, the RadarBERT encoder no longer drops masked frames. The masked frames have an 80% chance of being set to zeros, a 10% chance of being replaced with random numbers, and a 10% chance of remaining unchanged.

RadarAE-S has the same model structure with RadarAE but it does not utilize a pre-trained encoder. The encoder of RadarAE-S is randomly initialized and learnable.

DeepCNN is a deep CNN-based model that consists of convolutional layers, fully-connected layers, batch normalization layers and max-pooling layers.

DeepLSTM is a model consisting of 2 bidirectional LSTM layers followed by an MLP. The STMM is separated into 72 frames along the time domain.

DeepSoli [10] leverages a CNN to extract features from a sequence of RDM and capture the temporal relations with LSTM layers. In order to improve its recognition accuracy, we implement a two-modal CNN network to extract features from HRDM and HRAM.

RFWash [7] implements a deep learning model with the structure of CNN+BiLSTM+CTC [25] which takes a sequence of RDM as input. The original CNN is also replaced with a two-modal CNN processing HRAM and HRDM.

Soli [8] extracts some statistical features (e.g., mean value and variation of velocity) from a sequence of RDM and classifies the gesture with an SVM.

D. Training

We randomly shuffle the whole dataset and divide it into the unlabeled set (80%), validating set (10%), and testing set (10%). Then we randomly select a subset of the unlabeled set as the labeled set. The number of samples for each gesture type in the labeled set is set to 1, 2, 4, and 8 to simulate the few-shot scenario. Note that the unlabeled set is used for the self-supervised learning phase which does not require labeled data. Only the labels of the labeled set are leveraged in the supervised learning phase.

The training setup is illustrated in Table. I. In the self-supervised learning phase, only RadarAE and RadarBERT are pre-trained with the unlabeled set. In the supervised learning phase, all models are trained with the labeled set. To mitigate the effect of random sample selection on final results, we generate five distinct labeled sets and conduct the supervised learning procedure five times. The final result is calculated as the average accuracy of the five models on the testing set.

V. Evaluation

A. Overall Performance

Fig. 8 shows the performance comparison of RadarAE and baseline models in different few-shot scenarios. The accuracy of RadarAE are 79.1%, 92.1%, 97.8% and 99.5% in the 1, 2, 4, and 8-shot scenarios, respectively. RadarAE consistently outperforms other baseline models by large margins. We observe that the overall performance of RadarBERT is between RadarAE and other baselines. This result indicates that the pre-training method of BERT is not as effective as that of MAE in our scenario because BERT cannot make use of inter-distance relations of STMM. As expected, DeepCNN and DeepGRU perform much worse than RadarAE because they suffer from over-fitting due to the tiny number of training data. RFWash and DeepSoli achieve a worse result than DeepCNN and DeepGRU. We suspect that this is because they adopt a different input feature (a sequence of HRDM and HRAM) from STMM. STMM is more compact and informative than

TABLE I: Training Setup

Phase	Model	Trained	Selected	Evaluated	Data Augmentation	Loss Function	Epochs
Self-Supervised	RadarAE, RadarBERT	Unlabeled Set	Validating Set	N/A	RTC	MSE	3600
Supervised	All Models	Labeled Set	Validating Set	Testing Set	RTC	Cross-Entropy	700

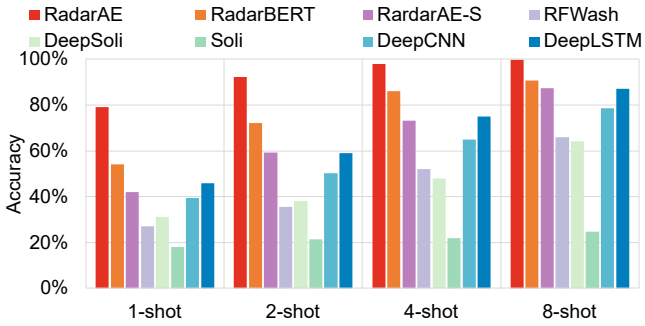
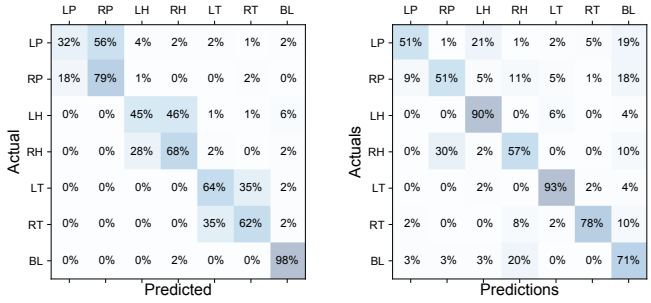


Fig. 8: The Performance comparison of RadarAE and baselines.



(a) (b)

Fig. 9: The confusion matrices of (a) RadarAE-RV and (b) RadarAE-RA.

HRDM and HRAM, which mitigates the over-fitting caused by limited labeled data. Soli only achieves 24.7% in the 8-shot scenario, probably because the adopted statistical features are not compatible with our gesture set and radar device. It is notable that the performance of RadarAE-S is close to that of baseline models like DeepLSTM, which proves that the pre-training procedure plays a pivotal role in enhancing the classifier model's performance.

Above all, RadarAE achieves outstanding results compared to state-of-the-art baselines, which verifies the effectiveness of RadarAE in extracting generic representations and facilitating radar-based gesture recognition.

B. Impact of Cropping

Based on HRDM and HRAM, we propose the novel feature STMM which is insensitive to different user positions. To verify the effectiveness of the cropping procedure, we calculate RV and RA from RDM and RAM instead of HRDM and HRAM and train a comparing model with the altered input. The results are shown in Table II. It is apparent that the performance of RadarAE deteriorates seriously due to the incapability of adapting to different user positions. The remarkable decline in accuracy proves that the proposed cropping procedure plays an essential part in precise recognition with volatile user positions.

TABLE II: Impact of Cropping on the accuracy of RadarAE

Scenario	1-shot	2-shot	4-shot	8-shot
w/ cropping	79.1%	92.1%	97.8%	99.5%
w/o cropping	35.1%	47.9%	53.9%	71.7%

C. Impact of Input Features

In this section, we verify the significance of combining RV and RA together as the input of our models. We implement two types of modified RadarAE models for comparison: a) **RadarAE-RV**, a comparing model that only leverages the first channel in STMM (i.e., T frames of continuous RV) as input. b) **RadarAE-RA** a comparing model that only leverages the second channel in STMM (i.e., T frames of continuous RA) as input. Specifically, the two models are required to reconstruct the corresponding input in the self-supervised learning phase; in the supervised learning phase, they are fed with their corresponding input to infer the gesture categories. In the 1, 2, 4, and 8-shot scenarios, the accuracy of RadarAE-RV and RadarAE-RA only achieve 51.9%, 53.5%, 62.2%, 69.4% and 49.2%, 64.0%, 69.0%, 79.8%, respectively. Fig. 9a and Fig. 9b illustrate the confusion matrices of these models in the 4-shot scenario. We find that RadarAE-RV frequently confuses symmetrical gestures, such as "left hook" and "right hook"; RadarAE-RA often makes wrong predictions for the gestures with similar moving ranges, like "right punch" and "right hook". This result indicates that RV provides information about motions along the radial direction, while RA provides information about motions along the tangential direction. Both RV and RA are indispensable for precise gesture recognition.

D. Impact of Patch Size

In this part, we evaluate the performances of RadarAE models with the patch sizes of 2×10, 4×10, 3×5, 3×20, and 3×10. The results are demonstrated in Fig. 10. We find that the patch size of 3×10 achieves the best performance. For the patch size of 3×20, the performance degrades because the model cannot utilize inter-distance relations as the patch size along the distance axis equals Y. We believe that a large patch size (e.g., 3×20 and 4×10) worsens the recognition accuracy due to the insufficient temporal and inter-distance relations. However, when the patch size gets too small (e.g., 2×10 and 3×5), the model fails to extract enough information from single patches and becomes less accurate. In addition, the patch size should be smaller along the time axis, as the

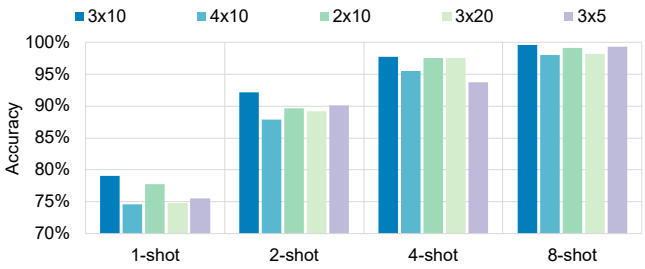


Fig. 10: Accuracy of RadarAE with different patch sizes.

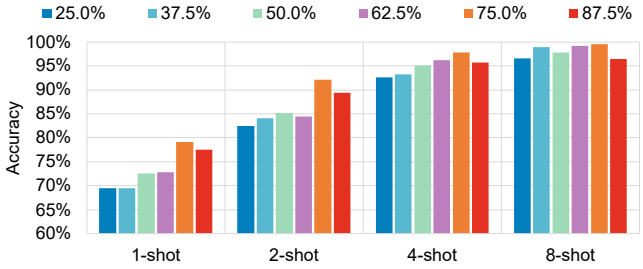


Fig. 11: Accuracy of RadarAE with different masking rates.

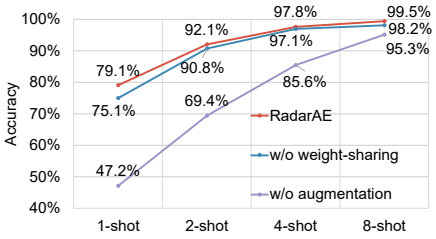


Fig. 12: The effect of weight-sharing and data augmentation on the recognition accuracy.

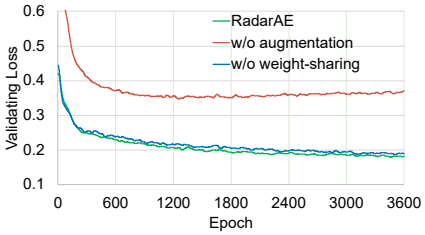


Fig. 13: The effect of weight-sharing and data augmentation on the smoothed validating loss during pre-training.

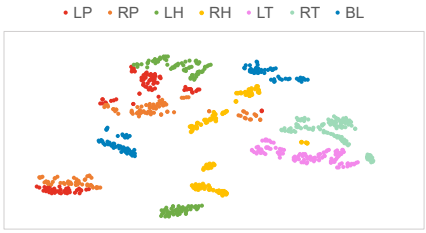


Fig. 14: Visualization of learned representations.

temporal relations between patches provide more instructive information for gesture recognition than the inter-distance relations do.

E. Impact of Different Masking Ratio

This section investigates the influence of different masking ratios. In Fig. 11 we plot the accuracy of RadarAE with different masking ratios. We find that 0.75 is the optimal choice for our scenario whereas larger or smaller masking ratios undermine the quality of self-supervised learning. The behind reason is that: a relatively high masking ratio is necessary to prevent the model from simply copying the surrounding patches to reconstruct the missing ones. Yet an extremely high masking ratio provides insufficient information to reconstruct the original STMM, making the model fail to converge. The masking ratio of 75% avoids the cheating behavior of the model and leaves adequate information for reconstruction, allowing RadarAE to extract representations from the input effectively.

F. Impact of Data Augmentation and Weight-Sharing

This subsection investigates the impact of the proposed data augmentation (RTC) and weight-sharing mechanism. For comparison, we train two RadarAE models without data augmentation or weight-sharing. The accuracy of RdarAE and comparing models are plotted in Fig. 12. The results reveal that the performance of the model without data augmentation deteriorates tremendously, probably due to the reduced diversity of training samples. It is notable that the weight-sharing mechanism also slightly improves the recognition accuracy. This is likely because the reduced amount of parameters alleviate the severe over-fitting. To have a better understanding of the effect of RTC and weight-sharing, we plot the smoothed validating loss in the self-supervised learning phase (MSE loss on the validating set) of RadarAE and comparing models in Fig. 13. The validating loss without data augmentation is much higher and even begins increasing when the epoch number exceeds 1200, implying that RTC enriches the diversity of the dataset and does not destruct the original data distributions at the same time. Besides, the validating loss without weight-sharing is slightly higher than that of RadarAE, which conforms to our previous analysis of the effect of weight-sharing.

G. Representation Visualization

To verify the effectiveness of representation learned by RadarAE, we leverage t-distributed Stochastic Neighbor Embedding (t-SNE) [26] to map the extracted representations into 2-D space. Fig. 14 shows the distribution of representations of different gesture types. It is apparent that the representations of different gestures tend to appear in different positions, which helps the classifier model distinguish different gestures. We also discover that the representations of some gestures form several distinct clusters, probably due to the high variation of users' behaviour. This phenomenon leads to a relatively low recognition accuracy in 1-shot and 2-shot scenarios. Another interesting finding is that the representations of similar gestures (*left punch*, *right punch*, *left hook*, and *right hook*) appear to be close to each other. In contrast, the representations of *right turn*, *left turn*, and *block* have obvious boundaries because these gestures have very different patterns from others. Above all, the representations learned by RadarAE are discriminative and beneficial to follow-up gesture recognition.

VI. Related Work

Millimeter-wave radar-based gesture recognition has attracted much attention since millimeter-wave radars are low-

cost and easy to deploy compared with other wireless devices [7]–[12]. Many radar-based works [7]–[11] adopt radars transmitting FMCW signals, which are modulated with the reflectors' distance, velocity, and azimuth. Thus, some works perform FFT over different axes of FMCW signal data to generate RDM as the input of their deep learning model [7], [8], [10]. For instance, RFWash [7] processes RDM with a deep learning model with the structure of CNN+LSTM+CTC [25] to predict handwashing gesture sequence performed at a bathroom sink. One shortage of these works is that they do not utilize the information of directions of reflectors, which may lead to the incapability of recognizing symmetric gestures. Other works generate sparse 3-D point clouds from RDM for further processing [9], [11]. [9] proposes a hybrid architecture combining the PointNet++ model [27] and LSTM to infer the gesture type from the point clouds data. However, these methods are only suitable for radars with a 2-D antenna array because radars with a 1-D antenna array (including our radar device, IWR1642) cannot generate 3-D point clouds. Though these existing methods perform well, they need a large amount of labeled data. In contrast, our framework enables radar sensing models to achieve superior performance with a small number of labeled samples.

VII. Conclusion

This paper presents RadarAE, a framework pushing forward the application of radar-based gesture recognition by fusing self-supervised learning techniques. Due to the sparseness of radar data, we propose a novel input feature STMM to facilitate the self-supervised learning procedure. Meanwhile, we meticulously analyze the properties of radar data and propose an MAE-like pre-training model with a series of adaptions and enhancements to make RadarAE learn better representations from unlabeled radar data. Extensive experiment results verify that RadarAE outperforms state-of-art works by a large margin in few-shot scenarios, addressing the over-fitting caused by the small amount of labeled data. Due to its low reliance on labeled data, RadarAE is a pivotal step towards practical gesture recognition systems.

VIII. Acknowledgments

This research is supported by a grant from National Natural Science Funds of China, No. 62102245.

References

[1] D. Mehta *et al.*, "Vnect: real-time 3d human pose estimation with a single RGB camera," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 44:1–44:14, 2017.

[2] G. Gkioxari, R. B. Girshick, P. Dollár, and K. He, "Detecting and recognizing human-object interactions," in *CVPR*. Computer Vision Foundation / IEEE Computer Society, 2018, pp. 8359–8367.

[3] Y. Liu, Z. Li, Z. Liu, and K. Wu, "Real-time arm skeleton tracking and gesture inference tolerant to missing wearable sensors," in *MobiSys*. ACM, 2019, pp. 287–299.

[4] Q. Zhang, J. Jing, D. Wang, and R. Zhao, "Wearsign: Pushing the limit of sign language translation using inertial and EMG wearables," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, no. 1, pp. 35:1–35:27, 2022.

[5] Y. Yu, D. Wang, R. Zhao, and Q. Zhang, "RFID based real-time recognition of ongoing gesture with adversarial learning," in *SenSys*. ACM, 2019, pp. 298–310.

[6] K. Caine, S. Sabanovic, and M. Carter, "The effect of monitoring by cameras and robots on the privacy enhancing behaviors of older adults," in *HRI*. ACM, 2012, pp. 343–350.

[7] A. Khamis, B. Kusy, C. T. Chou, M. McLaws, and W. Hu, "Rfwash: a weakly supervised tracking of hand hygiene technique," in *SenSys*. ACM, 2020, pp. 572–584.

[8] 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 Trans. Graph.*, vol. 35, no. 4, pp. 142:1–142:19, 2016.

[9] S. Palipana, D. Salami, L. A. Leiva, and S. Sigg, "Pantomime: Mid-air gesture recognition with sparse millimeter-wave radar point clouds," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 5, no. 1, pp. 27:1–27:27, 2021.

[10] 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*. ACM, 2016, pp. 851–860.

[11] H. Liu *et al.*, "Real-time arm gesture recognition in smart home scenarios via millimeter wave sensing," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, no. 4, pp. 140:1–140:28, 2020.

[12] P. S. Santhalingam *et al.*, "mmasl: Environment-independent ASL gesture recognition using 60 ghz millimeter-wave signals," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, no. 1, pp. 26:1–26:30, 2020.

[13] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT (1)*, 2019, pp. 4171–4186.

[14] K. He *et al.*, "Masked autoencoders are scalable vision learners," *CoRR*, vol. abs/2111.06377, 2021.

[15] B. Fernando, H. Bilen, E. Gavves, and S. Gould, "Self-supervised video representation learning with odd-one-out networks," in *CVPR*. IEEE Computer Society, 2017, pp. 5729–5738.

[16] H. Xu, P. Zhou, R. Tan, M. Li, and G. Shen, "LIMU-BERT: unleashing the potential of unlabeled data for IMU sensing applications," in *SenSys*. ACM, 2021, pp. 220–233.

[17] Y. Zheng, Y. Zhang, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Zero-effort cross-domain gesture recognition with wi-fi," in *Mobisys*. ACM, 2019, pp. 313–325.

[18] C. Dian, D. Wang, Q. Zhang, R. Zhao, and Y. Yu, "Towards domain-independent complex and fine-grained gesture recognition with rfid," *Proceedings of the ACM on Human-Computer Interaction*, vol. 4, no. ISS, pp. 1–22, 2020.

[19] Q. Zhang, D. Wang, R. Zhao, Y. Yu, and J. Shen, "Sensing to hear: Speech enhancement for mobile devices using acoustic signals," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 3, pp. 1–30, 2021.

[20] A. Da'u, N. Salim, and I. Rabiu, "Multi-level attentive deep user-item representation learning for recommendation system," vol. 433, 2021, pp. 119–130.

[21] A. Vaswani *et al.*, "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008.

[22] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[23] L. J. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *CoRR*, vol. abs/1607.06450, 2016.

[24] F. Robey, S. Coutts, D. Weikle, J. McHarg, and K. Cuomo, "Mimo radar theory and experimental results," in *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004.*, vol. 1, 2004, pp. 300–304 Vol.1.

[25] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *ICML*, ser. ACM International Conference Proceeding Series, vol. 148. ACM, 2006, pp. 369–376.

[26] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[27] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *NIPS*, 2017, pp. 5099–5108.