mmWave Radar-Based Unsupervised Gesture Recognition via Image-Aligned Heterogeneous Domain Transfer

Qihua Feng, Kunpeng Cheng, and Chunhui Duan, *Member, IEEE*

***Abstract*—Human Gesture Recognition (HGR) using mmWave radar has become increasingly promising due to its exceptional contactless perception sensitivity. Conventional approaches predominantly rely on supervised models to learn radar signals, thus incurring substantial costs associated with annotation. To address this limitation, certain works embrace transfer learning to effectively transfer knowledge from labeled source domain to unlabeled target domain, achieving unsupervised recognition in the target domain. However, existing transfer-based methods still necessitate large-scale labeled source domain radar data, thereby constraining their practical applicability. To this end, we propose a novel unsupervised solution for mmWave-based HGR by transferring public image gestures to radar data, eliminating the need for acquiring labeled radar data in source domain. We aim to establish heterogeneous alignment between images and radar signals, facilitating cross-domain transfer. Initially, we mitigate the negative impact of data heterogeneity by employing sophisticated signal processing techniques to convert raw radar signals into gesture trajectories. Subsequently, we introduce an Adversarial-Contrastive Domain Transfer Model (ACDTM) to achieve fine-grained alignment. ACDTM not only confuses the source and target domains by adversarial learning, enabling the acquisition of domain-invariant features, but also designs a robust similarity matrix to facilitate intra-class alignment through contrastive learning. Additionally, ACDTM conducts adversarial self-training on target domain with pseudo-labeled distribution. Our experimental findings substantiate that the unsupervised accuracy achieves about 80∼92% on different mmWave gesture datasets, outperforming existing unsupervised HGR schemes by large margins.**

***Index Terms*—Gesture recognition, mmWave radar sensing, unsupervised learning, heterogeneous domain transfer.**

I. Introduction

With the advancement of digital services, Human Gesture Recognition (HGR) has found extensive applications in various fields, including smart living [1] and intelligent healthcare [2]. Although traditional computer vision-based HGR has reached a mature stage [3], [4], it often encounters performance limitations under poor lighting conditions and raises concerns about privacy [5], [6], [7], [8], [9]. In recent years, the rapid evolution of wireless sensing technologies [10] has effectively addressed these limitations associated with vision-based approaches. Wireless sensing techniques utilize emitted electromagnetic signals to model motion gestures, thereby introducing a new paradigm of perception. Among these techniques, millimeter-wave (mmWave) radar sensor technology plays a vital role. Compared to low-frequency signals like WiFi and UWB, mmWave signals offer high frequency and large bandwidth, enabling enhanced sensitivity and fine-grained perception capabilities [6]. Consequently, a growing body of research focuses on leveraging mmWave sensing for HGR [1], [5], [7], [11], [12], leading to significant breakthroughs in this field.

Typical mmWave radar-based HGR methods involve the collection of data, followed by the construction of Deep Neural Networks (DNN) to learn and identify gesture patterns from radar data [1], [7], [11], [12], [13]. However, these methods heavily rely on supervised learning, which incurs significant costs due to the need for manual labeling. The labeling process is further complicated by the challenge of comprehending Radio-Frequency (RF) signals, making it more difficult for humans. To address this issue, some works employ transfer learning techniques to alleviate the burden of manual annotation [5], [14]. For example, UDARF [5] proposes leveraging a large amount of labeled signal data from Environment A (referred to as the source domain) and transferring its knowledge to unlabeled signal gestures in Environment B (the target domain). The source and target domains may be related (e.g., having similar categories), but they often exhibit significant distribution differences. In UDARF, pseudo labeling-based domain adaptation techniques [15] are employed to mitigate distribution discrepancies, enabling unsupervised learning in target domain. However, existing transfer-based works necessitate a substantial amount of labeled source domain signal data. Acquiring large-scale annotated and related signal data poses significant challenges in the context of wireless sensing. Consequently, a straightforward homogeneous transfer between radar data encounters limitations in terms of feasibility in practical settings.

In contrast, domain transfer techniques have demonstrated significant effectiveness in the field of computer vision [15], [16], [17], [18], [19] primarily due to the availability of abundant open image datasets in practical scenarios. For example, there are numerous open-source gesture image datasets, including numerical and alphabetical gestures [20], [21], [22]. If we

Received 2 April 2025; revised 24 August 2025; accepted 22 September 2025. Date of publication 25 September 2025; date of current version 4 February 2026. This work was supported by the Beijing Institute of Technology Research Fund Program for Young Scholars. Recommended for acceptance by M. Furkan Keskin. *(Corresponding author: Chunhui Duan.)*

The authors are with the Beijing Institute of Technology, Beijing 100081, China (e-mail: duanch@bit.edu.cn).

Code is available at https://github.com/onlinehuazai/mmGesture.

Digital Object Identifier 10.1109/TMC.2025.3614353

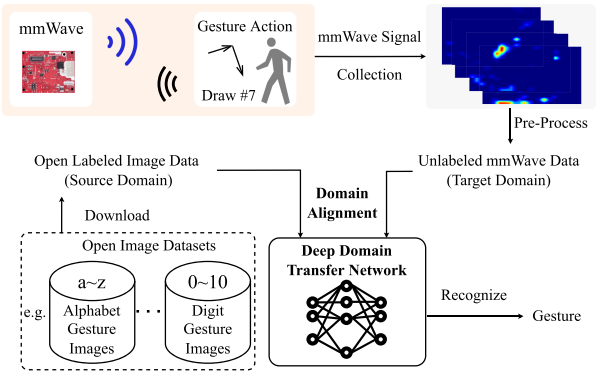


Fig. 1. The illustration of the proposed approach framework. Our study transfers knowledge of open image datasets to unlabeled mmWave data. By aligning source and target domains, we can achieve unsupervised mmWave gesture recognition.

can transfer the knowledge from these heterogeneous image gestures to radar gesture signals, it would significantly reduce the need for collecting source domain radar data. Moreover, unsupervised domain adaptation does not require downstream fine-tuning like in self-supervised learning. Consequently, our objective, as depicted in Fig. 1, is to align radar data with images and achieve unsupervised recognition of radar gestures through heterogeneous domain transfer learning. However, this type of heterogeneous transfer presents two main challenges at the data level and in terms of model adaptation:

- Image and radar signals are completely heterogeneous data types, differing significantly in terms of data representation, dimensions, and meanings. Therefore, it is essential to analyze and process the heterogeneous relationship between the data, mitigating the negative effects of heterogeneity and facilitating successful transfer.
- There exists a substantial distribution discrepancy between radar signals and image data, making it easy for models to differentiate between the domains based on their distributions. This discrepancy poses a challenge in aligning the source and target domains.

To address the aforementioned challenges, we propose a novel mmWave-based unsupervised gesture recognition method that focuses on aligning information between the heterogeneous image and radar domains. *Firstly*, we establish a bridge, gesture trajectory images, between images and radar signals to effectively mitigate the negative effects of heterogeneous transfer at the data level. These trajectory images are constructed by applying point cloud filtering techniques within and between frames of radar signal data. *Secondly*, we employ an Adversarial-Contrastive Domain Transfer Model (ACDTM) to achieve fine-grained alignment between image and radar data. Inspired by the concept of an adversarial minimax game, ACDTM aims to confuse the source and target domains, ultimately aligning the trajectory images of the target domain with the data distribution of the source domain. Furthermore, ACDTM utilizes the ratio test method to construct a sample-level similarity matrix between the target and source domains. This matrix enhances intra-class alignment through the computation of the contrastive loss on the similarity matrix. To fully exploit unlabeled samples and enhance model robustness, ACDTM employs Adversarial Self-Training on target domain with pseudo-labeled distribution. The experimental results demonstrate that our method successfully transfers information from heterogeneous images to radar signals, achieving a promising unsupervised performance.

Overall, our work contributes in the following three aspects:

1) We propose a novel framework that utilizes heterogeneous domain transfer for unsupervised mmWave gesture recognition. By aligning with open-source images, we tackle the challenge of collecting large-scale signal data and transfer heterogeneous image knowledge to radar data.
2) To bridge the gap between the target domain radar signals and the source domain images at the data level, we use a simple yet effective concept of gesture trajectory images. Furthermore, we explore an ACDTM to achieve fine-grained alignment, mitigating the distribution differences between the source and target domains.
3) Experimental results demonstrate the effectiveness of our approach. Without fine-tuning, our unsupervised mmWave gesture recognition achieves an accuracy of about $80 \sim 92\%$, surpassing existing unsupervised HGR schemes by substantial margins.

The rest of this paper is organized as follows. Section II introduces the related work. In Section III, the overview of our system is presented. We elaborate on method details of the proposed approach in Section IV. In Section V, we describe the implementation and evaluation of our system. Some potential improvements and research directions are explored in Section VI. Finally, Section VII summarizes the conclusion.

II. Related Work

Unsupervised Domain Transfer: Models trained on annotated source domains are often sensitive to domain shifts, primarily manifested in poor generalization performance when directly applied to another target domain due to significant distribution biases [15], [19]. Unsupervised domain adaptation techniques have emerged to minimize the distribution disparities or distances between the labeled source domain and the unlabeled target domain, facilitating the transfer of knowledge from source domain to target domain. Unsupervised domain adaptation finds extensive applications in deep learning, primarily encompassing three types: discrepancy-based, adversarial-based, and pseudo labeling-based methods. Discrepancy-based methods establish distance functions between the source and target domains to measure the disparities at corresponding feature embeddings. For example, Long et al. [16] define the sum of Multiple Kernel variant of Maximum Mean Discrepancies (MK-MMD) between FC layers as the distance. Chen et al. [23] propose higher-order statistics as distance function, and further extend it into reproducing kernel Hilbert spaces. Adversarial-based methods align global distribution by minimax game, which fools discriminator to confuse domains and learns domain-invariant features. For instance, DANN [17] proposes GRL to train domain adversarial networks. CDAN [18] considers aligning conditional distributions in discriminator to improve discriminability. Since target

domain is unlabeled, to take into account class-specific adaptation during the transfer process, some works [24], [25], [26] predict pseudo-labels to target domain in the training stage.

Video-based Radar Data Synthesis: Several notable works, such as Vid2Doppler [27], Midas [28], Midas++ [29], and SBRF [30], synthesize radar signals from videos, offering innovative solutions to mitigate mmWave data scarcity. The general pipeline of video-based synthesis is: First, they extract skeletal information from video keyframes and reconstruct 3D human meshes. Then, according to the human meshes, they compute radial velocity, visibility, and Radar Cross-Section (RCS) to obtain preliminary simulated signals by physical modeling. Finally, they refine simulated radar data using real data by neural networks (e.g., U-Net [31], Transformer [32]). However, existing methods face several challenges: (1) They still require real data for training, such as loss computation between simulated and real signals during refinement stage; (2) High computational costs and need for synchronized camera-radar data collection. The process relies on robust computer vision models for mesh estimation and large-parameter models (e.g., Transformer) for refinement, resulting in high computational complexity. Our approach reduces model training overhead by converting radar gesture signals into low-dimensional trajectory pictures and achieves unsupervised recognition by aligning with open-source gesture images. Additionally, since we only have radar data without paired video data, video-based radar data synthesis falls outside the scope of this paper.

mmWave Radar Sensing: With the development of the Internet of Things (IoT), mmWave sensing starts impacting life, which can sense human in a contactless way [6]. For example, the works [33], [34] utilize mmWave radar to track and localize human. [35], [36], [37] propose human vital signs measurement using mmWave, such as respiration and heartbeat signal. Our study pays attention to gesture recognition using mmWave, which has been explored by many works. For instance, RFWash [13] utilizes radar to monitor the Alcohol-Based Hand Rub and employ BiLSTM [38] to learn gestures. mmASL [7] and mHomeGes [1] extract Doppler information from mmWave signals and employ CNN-based model to recognize gesture signs. Li et al. [12] propose a series of coupling data augmentations for mmWave signals and learns gestures with a CNN-RNN-based [8] deep model. Pantomime [11] proposes a hybrid deep model (combinations of the PointNet++ [39] and LSTM) to achieve accurate gesture recognition with sparse point clouds. These mmWave-based gesture recognition works [1], [7], [11], [12], [13] have achieved excellent performance and promoted its development, but they are supervised learning which causes extra labeling overhead. UDARF [5] proposes unsupervised gesture recognition by transferring source knowledge to the unlabeled target domain, using pseudo labeling-based domain adaptation techniques. However, their source and target domains are homogeneous, and they require labeled mmWave radar data in the source domain, which makes it hard to collect related large-scale labeled radar data in wireless sensing. Our study aims to cope with this challenge by using am open image gesture dataset to replace radar data of source domain. We explore heterogeneous domain transfer from images to radar signals, achieving mmWave-based unsupervised gesture recognition with unlabeled target domain radar signals.

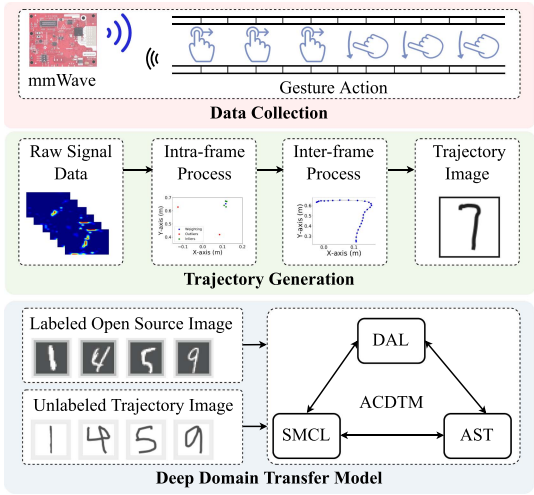


Fig. 2. System overview of our proposed method.

III. System Overview

The architecture of our proposed approach is depicted in Fig. 2, which consists of two main modules: mmWave trajectory generation and deep domain transfer model.

mmWave Trajectory Generation: After collecting the gesture data using mmWave radar, we obtain the raw radar signal data. The mmWave trajectory generation module aims to process this raw data through signal processing techniques to reconstruct the gesture trajectory. Each gesture data corresponds to T frames, which are processed sequentially within frames and between frames to generate a trajectory image. Within t-th frame data $Frame_t$, we perform Fast Fourier Transform (FFT) and noise reduction operations to obtain point cloud data. By accumulating the point clouds from T frames and applying filtering techniques, we can obtain a stable trajectory image that represents the gesture.

Deep Domain Transfer Model: Upon obtaining the trajectory image, we can transfer the knowledge learned from open gesture images to our trajectory images. The open gesture images, referred to as source domain, come with labels, while the mmWave trajectory image serves as the target domain without labels. Nevertheless, due to the nature of radar electromagnetic signals, our gesture trajectory maps differ significantly from real gesture images. For instance, trajectory maps consist of discrete points, and the resulting connected-line trajectories are less standardized compared to open-source gesture images. Additionally, radar-specific characteristics lead to sparse and potentially inaccurate point localization, often causing distorted or incomplete trajectories. To better align unlabeled trajectory maps with public images, we construct an ACDTM to reduce the discrepancy in data distribution for unsupervised mmWave gesture recognition. First, we employ Domain Adversarial Learning

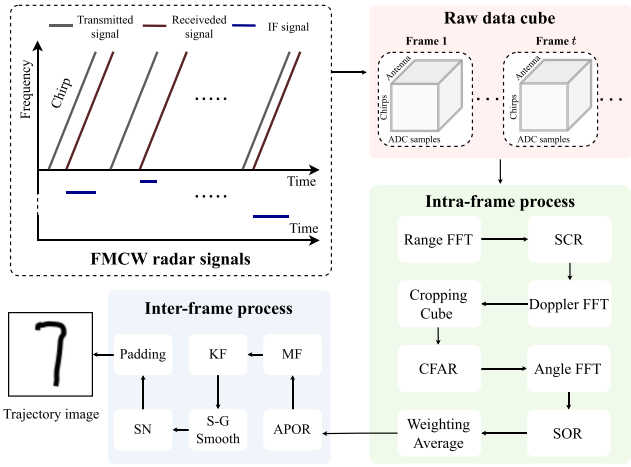


Fig. 3. The sketch of trajectory generation using radar signals.

(DAL) to globally align the distributions of source and target domains, learning domain-invariant features. Second, local intra-class alignment further reduces discrepancies in the conditional data distributions, and we propose a Similarity-Matrix-based Contrastive Learning (SMCL) module. Specifically, we construct similarity matrices between source and target domains based on ratio test techniques, using contrastive learning to minimize intra-class distances while maximizing inter-class separation. Furthermore, to fully leverage unlabeled target domain samples, pseudo-labeling combined with self-training is commonly adopted. However, pseudo-labels may introduce noise. Thus, ACDTM employs Adversarial Self-Training (AST) on the target domain using pseudo-label distributions to enhance model robustness. In summary, our method comprehensively aligns source and target domain distributions both globally and locally, while AST improves robustness.

In Section IV, we provide the methodological details of our system. Specifically, Section IV-A depicts the mmWave trajectory generation algorithm, and our deep domain transfer model is given in Section IV-B. Unless specified otherwise, our study focuses on typical digit and letter gestures.

IV. Method

A. Trajectory Generation

We employ a commercial off-the-shelf TI IWR1843BOOST radar [40] to collect mmWave data, which has been widely utilized in mmWave radar sensing tasks [6]. The radar continuously transmits Frequency Modulated Continuous Wave (FMCW) signals, and our study employs a series of processing techniques to extract the corresponding trajectory image from the collected signals. The signal processing is described in Fig. 3, containing intra-frame and inter-frame processes.

1) Intra-Frame Process: The trajectory diagram is formed by connecting the coordinate points from each moment, with each raw data frame at every moment corresponding to a coordinate point. Thus, it is necessary to capture the signals relevant to gesture movements in each frame and process them into a coordinate point. Illustrated in Fig. 3, the intra-frame procedure encompasses range FFT, Static Clutter Removal (SCR), Doppler FFT, cropping cube, Constant False Alarm Rate (CFAR) [41], angle FFT, Statistical Outlier Removal (SOR), and weighting average. Hereafter, we offer an elaborate elucidation of the intra-frame process.

Range FFT: The FMCW radar operates by periodically transmitting chirp signals and receiving the reflected signals from objects through receive antennas. The mixer in radar system produces Intermediate Frequency (IF) signal by combining received chirps with transmitting chirps. The frequency f_{IF} of IF signal corresponds to the distance d of the object and can be mathematically expressed as:

$$f_{IF} = \frac{S \times 2d}{c} \Rightarrow d = \frac{f_{IF} \times c}{2S}, \tag{1}$$

where S is the slope of the chirp signal, and c is the speed of light. Hence, using FFT on Analog-to-Digital Converter (ADC) samples dimension to estimate f_{IF} can obtain range information of object.

SCR: In the environment, various static objects such as furniture and walls are present. However, since our focus is on capturing dynamic gesture actions, it is necessary to eliminate purely static objects from signals. Considering these clutter objects remain stationary with respect to slow time, but gesture actions are with large variance, so we suppress the clutter reflections through high-pass filtering in the slow time direction [42]. Specifically, we calculate the average values of the signals on slow time and subtract them from the data.

Doppler FFT: The moving speed of the object can be determined using the Doppler effect. By analyzing the phase shift $\Delta\phi_1$ between consecutive chirps, we can calculate velocity v. Assume that the interval between two consecutive chirps is T_c, the relationship between $\Delta\phi_1$ and v is described as:

$$\Delta\phi_1 = \frac{4\pi v T_c}{\lambda} \Rightarrow v = \frac{\lambda \Delta\phi_1}{4\pi T_c}, \tag{2}$$

where λ is the wavelength of the mmWave signal. Therefore, performing FFT across chirps to estimate phase shift which can be transformed to Doppler velocity. This process allows us to obtain the Range-Doppler Map (RDMap).

Cropping Cubes: In actual scenes, the gesture of a person performs close to radar sensor [12], so we select a specific range bins in ADC samples dimension. For example, the ADC sample is 256 in our study, and we select the previous A_S range bins. Similarly, the actual speed of hand is not very fast, so we crop too large Doppler bins and remain the velocity values from $-\frac{V_D}{2}$ to $\frac{V_D}{2}$ relative to the Doppler bins. The A_S and V_D are hyper-parameters, and we give an example of cropped RDMap in Fig. 4(a).

CFAR: Environment noise remains in the cropped RDMap due to multipath interference, which confuses the target gesture signals. CFAR is a typical technique to detect targets against environment noise [6], adaptively selecting a noise threshold [12] to detect the object cells (e.g., hand). We apply CFAR along Doppler and range dimensions, and values below the threshold are considered clutter and are removed from the cropped

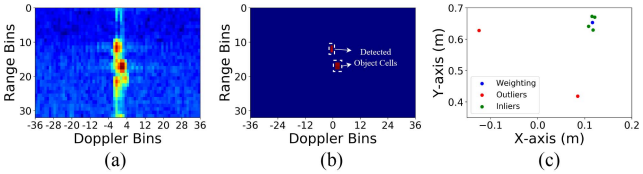


Fig. 4. Examples of intra-frame process. (a): cropped RDMap ($A_S = 32, V_D = 36$); (b): detected object cells after CFAR; (c): points of a frame in the Cartesian coordinate system (red points are outliers, green points are filter inliers after SOR, and blue point is the final weighting average result).

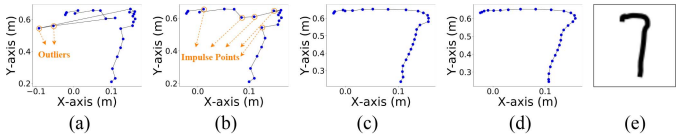


Fig. 5. Examples of inter-frame process. (a) original trajectory; (b) after APOR; (c) after MF and interpolation; (d) after KF and S-G smooth; (e) final trajectory image.

RDMap. This process is shown in Fig. 4(b), where the remaining candidate cells are related to action.

Angle FFT: For the remaining candidate cells, we already possess distance information d, and acquiring angle information enables us to obtain precise coordinate details. The Angle of Arrival (AOA) can be conducted at receiver array with multiple elements [6]. According to the phase changes $\Delta\phi_2$ between adjacent receiving antennas, the AoA θ can be derived from $\Delta\phi_2$:

$$\Delta\phi_2 = \frac{2\pi l \sin\theta}{\lambda} \Rightarrow \theta = \sin^{-1}\left(\frac{\lambda\Delta\phi_2}{2\pi l}\right), \tag{3}$$

where l is the distance between adjacent receiving antennas. Therefore, alongside the antenna dimension performing FFT, we can obtain angle information. According to the range d and angle θ of detected object cells, we can generate the plane coordinate point by converting them into the Cartesian coordinate system, which can be calculated as $(d\sin\theta, d\cos\theta)$. As shown in Fig. 4(c), after angle FFT, there are six points (red and green points) in the Cartesian coordinate system that are converted from Fig. 4(b).

SOR: Due to the nature of electromagnetic signals, noise may persist within the candidate point cloud. This type of noise, termed outliers, typically consists of isolated points that are significantly distant from normal points. To eliminate outliers, we utilize SOR method [43] for detection and removal. The SOR performances k-Nearset Neighbor for each point and calculates the average distance to its neighboring points. When the average distance of a point is larger than a threshold τ_{intra}, this point is be classified as outliers. The threshold τ_{intra} is defined as:

$$\tau_{intra} = \mu_{intra} + \beta_1 \times \sigma_{intra}, \tag{4}$$

where μ_{intra} and σ_{intra} are mean and standard deviation of corresponding average distances of all points, respectively. β_1 is the weight of the standard deviation σ, and we set it to 1in our study. As shown in Fig. 4(c), two outliers (red points) are detected by using SOR.

Weighting Average: For the remaining valid points, we need to convert them into a single coordinate point to represent the trajectory coordinates of this frame. The most straightforward approach is to take the mean; however, this overlooks the significance of individual points. To allocate importance to different points, we employ a weighted sum based on the energy of the point cloud to derive the final coordinate, where points with higher energy hold greater weight. Suppose there are N^t_{intra} valid points of t-th frame, and their energy values are $\{E^t_1, \ldots, E^t_{N_{intra}}\}$, so the final point coordinate (px_t, py_t) of t-th frame is defined as:

$$px_t = \sum_{i=1}^{N^t_{intra}} \frac{\exp^{E^t_i}}{\sum_{j=1}^{N^t_{intra}} \exp^{E^t_j}} d^t_i \sin\theta^t_i, \tag{5}$$

$$py_t = \sum_{i=1}^{N^t_{intra}} \frac{\exp^{E^t_i}}{\sum_{j=1}^{N^t_{intra}} \exp^{E^t_j}} d^t_i \cos\theta^t_i, \tag{6}$$

where d^t_i and θ^t_i are range and angle of i-th point in the t-th frame, respectively. As shown in Fig. 4(c), the blue point is the final result after weighting average. For ease of notation, we designate (px_t, py_t) as p_t, where $1 \le t \le T$.

2) Inter-Frame Process: Upon acquiring the point coordinates for each frame, we aggregate them onto a two-dimensional plane. This consolidation allows us to depict the general gesture trajectory movement. Nevertheless, due to the sparsity and instability of radar point clouds, some points may still deviate from the trajectory. To bolster the robustness of the trajectory image, additional processing steps are necessary. These procedures encompass Adjacent-Point-based Outlier Removal (APOR), Median Filtering (MF), Kalman Filtering (KF), Savitzky-Golay (S-G) smoothing, Scale Normalization (SN), and padding, as illustrated in Fig. 3. Below, we detail the intra-frame process.

APOR: Due to the susceptibility of electromagnetic signals to multipath interference, points of a frame may significantly deviate from the trajectory. For instance, as depicted in Fig. 5(a), two outlier points are present in the trajectory of gesture 7, with these outliers exhibiting significant temporal deviations. To eliminate outliers in the trajectory, we utilize trajectory trends and distances to filter out these points. Anomalies typically exhibit trend shifts temporally and are notably distant from adjacent trajectory points. Thus, we first calculate the distances between adjacent points, and the adjacent distance of point p_t can be defined as:

$$adj\_dis[t] = \begin{cases} dis(p_t, p_{t+1}), & \text{if } t = 1, \\ dis(p_t, p_{t+1}) + dis(p_t, p_{t-1}), & \text{if } 1 < t < T, \\ dis(p_t, p_{t-1}), & \text{if } t = T, \end{cases} \tag{7}$$

where $dis(\cdot,\cdot)$ is distance function, such as Manhattan distance. Then, we compute the mean μ_{inter} and variance σ_{inter} of the $adj\_dis[1:T]$, which can be defined as:

$$\mu_{inter} = \frac{1}{T}\sum_{t=1}^{T} adj\_dis[t], \tag{8}$$

$$\sigma_{inter} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(adj\_dis[t] - \mu_{inter})}. \quad (9)$$

Finally, a threshold τ_{inter} is calculated based on the μ_{inter} and σ_{inter}, which can be defined as:

$$\tau_{inter} = \mu_{inter} + \beta_2 \times \sigma_{inter}, \quad (10)$$

where β_2 is the weight of σ_{inter}, and we set it to 1.5 in our study. APOR identifies points whose adjacent distances exceed τ_{inter} as outliers. As shown in Fig. 5(b), the two outliers are removed by using APOR.

MF: Some outlier points in the trajectory sequence may manifest as spikes but do not exhibit significantly large adjacent distances, making it challenging to filter out these points solely relying on APOR. For example, as depicted in Fig. 5, these spike points, resembling impulse shapes, can significantly impact the robustness of the trajectory. To address these impulse points, we apply MF to the trajectory sequence. MF is a nonlinear signal processing method grounded in ranking statistics theory. The fundamental principle of MF involves replacing the value of a point in the trajectory sequence with the median value of its surrounding neighborhood. This approach ensures that the neighborhood values closely approximate the true value, making it effective in filtering out impulse noise [44]. Furthermore, post MF, we perform a simple linear interpolation on the point cloud. Illustrated in Fig. 5(c) is an example after applying MF and interpolation, showcasing the removal of impulse points.

KF & S-G Smooth: To enhance the clarity of trajectory, the KF is commonly utilized in trajectory tracking tasks. The KF is a recursive state estimation algorithm employed to estimate the state of a dynamic system from a series of incomplete and noisy measurements. This dynamic system is subject to noise, often assumed to be white noise. To refine the estimated state, KF leverages measurements that are related to the state but are also subject to disturbances. Due to the favorable properties of the KF, we apply it to our trajectory coordinate points. Additionally, to enhance the smoothness of the trajectory, we employ the common S-G filtering technique [45]. S-G filtering is a digital signal processing method for smoothing sequential data, estimating smoothed values by locally fitting a polynomial to data points within a sliding window. Illustrated in Fig. 5(d) is the result after KF and S-G smooth. Assuming there are N_{inter} points after S-G smoothing, the coordinates of the points are represented as $[(\hat{px}_1, \hat{py}_1), \ldots, (\hat{px}_{N_{inter}}, \hat{py}_{N_{inter}})]$. Hence, the width W_1 and height H_1 of the trajectory can be defined as:

$$W_1 = \max\left(\hat{px}_{1:N_{inter}}\right) - \min\left(\hat{px}_{1:N_{inter}}\right), \quad (11)$$

$$H_1 = \max\left(\hat{py}_{1:N_{inter}}\right) - \min\left(\hat{py}_{1:N_{inter}}\right). \quad (12)$$

SN & Padding: After describing the trajectory using points and line segments, the next step involves transforming it into an image. To maintain the scale invariance of the gesture, we perform proportional normalization and scaling of the trajectory's height and width, and then pad it into an image of dimensions $W_2 \times H_2$. For instance, for gesture 1, where the width is significantly smaller than the height, resizing the gesture image directly to $W_2 \times H_2$ would result in severe distortion. To address this, we first calculate the ratio of height to width, normalize the length and width accordingly, and then scale them based on this ratio. Assuming the height is greater than the width and needs to be scaled to H_{edge}, the scaled width W_{edge} is expressed as:

$$W_{edge} = \frac{W_1}{H_1} \times H_{edge}. \quad (13)$$

Therefore, after the proportional scaling, the image dimensions become $W_{edge} \times H_{edge}$, where the coordinate point $(\hat{px}_{1:N_{inter}}, \hat{py}_{1:N_{inter}})$ is transformed into $(\hat{px}^{img}_{1:N_{inter}}, \hat{py}^{img}_{1:N_{inter}})$, with the calculation method as follows:

$$\hat{px}^{img}_i = \frac{\hat{px}_i - min(\hat{px}_{1:N_{inter}})}{W_1} \times W_{edge}, \quad 1 \le i \le N_{inter}, \quad (14)$$

$$\hat{py}^{img}_i = \frac{\hat{py}_i - min(\hat{py}_{1:N_{inter}})}{H_1} \times H_{edge}, \quad 1 \le i \le N_{inter}. \quad (15)$$

We connect all the coordinates in the order of their temporal relations, resulting in the $W_{edge} \times H_{edge}$ trajectory image. Finally, we fill the $W_{edge} \times H_{edge}$ image into a $W_2 \times H_2$ image, padding blank pixels around the perimeter. As shown in Fig. 5(e), this is an example, gesture 7, of the final trajectory.

B. Domain Transfer Model ACDTM

After obtaining gesture trajectory images, one approach to recognize these images is to use a pre-trained model on open images such as MNIST [20]. However, directly transferring the model to target domain leads to unsatisfactory performance, and our experiments show only about 33% recognition accuracy. The primary reason for this performance drop is the significant distribution difference between the source and target domains. One notable difference lies in pixel-level distributions between MNIST and trajectory images. MNIST gestures have high pixel resolution, whereas mmWave point clouds are sparse, thus the final trajectory images are not as standardized as MNIST images. Moreover, the pixel values in MNIST range from 0 to 255, whereas trajectory images lack detailed pixel values. Instead, the same pixel values are used to indicate the presence or absence of a trajectory.

To reduce the distribution difference between the source and target domains, domain transfer is a common method, which can effectively improve the unsupervised recognition accuracy of the target domain. In the problem of unsupervised domain transfer, we define the source domain with $\mathcal{D}^s = \{x_i^s, y_i\}_{i=1}^{|\mathcal{D}^s|}$, where x_i^s represents source samples, y_i is corresponding label, $|\mathcal{D}^s|$ is the total number of source samples. The distribution of source domain is P_s. Similar, the unlabeled target domain is defined as $\mathcal{D}^t = \{x_j^t\}_{j=1}^{|\mathcal{D}^t|}$, where $\mathcal{D}^t \sim P_t$ and $P_s \neq P_t$. In recent years, DAL has achieved remarkable performance in unsupervised domain adaptation [15], [17], [18]. DAL usually builds an adversarial target about the domain discriminator to encourage domain confusion and domain invariant features are learned through adversarial training. Our study also adopts the framework of DAL and introduces the idea of conditional adversarial to improve the transfer process. In addition, intra-class alignment can further

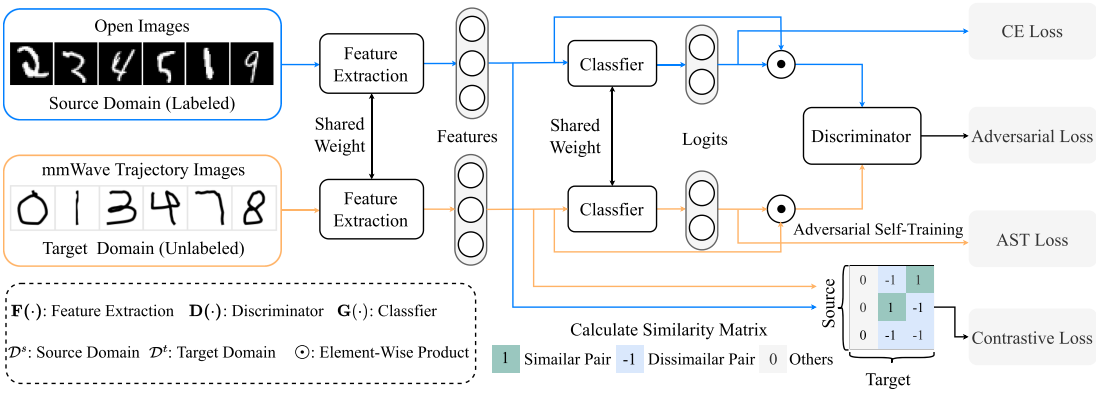


Fig. 6. The framework of the proposed ACDTM.

reduce the difference in the conditional distribution of the data. Therefore, ACDTM constructs the similarity matrix between the source domain and target domain and then uses contrastive learning to learn a more general feature space. To fully exploit unlabeled samples, it is common to assign pseudo-labels and employ self-training on the target domain. However, pseudo-labels may introduce additional noise, so ACDTM employs AST on target domain with pseudo-labeled distribution to enhance model robustness. Fig. 6 shows the framework of the proposed ACDTM and we detail it below.

1) Minimax Adversarial Learning: DAL is a two-player game: one player is domain discriminator $\mathbf{D}(\cdot)$ which is trained to distinguish the feature comes from source or target domains; another player is feature extraction $\mathbf{F}(\cdot)$ which is trained to confuse $\mathbf{D}(\cdot)$, enabling $\mathbf{D}(\cdot)$ unable to distinguish source and target domains, namely learning domain-invariant feature. Common Domain Adversarial Neural Network [17] (DANN) is formulated a minimax optimization problem by three competitive loss terms: (a) $\varepsilon(\mathbf{F},\mathbf{G})$ is minimized on $\mathbf{F}(\cdot)$ and classifier $\mathbf{G}(\cdot)$ of source domain; (b) $\varepsilon(\mathbf{D},\mathbf{F})$ is minimized over $\mathbf{D}$ across the source and target domains; (c) $\varepsilon(\mathbf{D},\mathbf{F})$ is maximized over $\mathbf{F}$ across the source and target domains. Therefore, the minimax game of DAL can be expressed as:

$$\min_{\mathbf{F},\mathbf{G}} \varepsilon\left(\mathbf{F},\mathbf{G}\right), \quad \max_{\mathbf{F}} \varepsilon\left(\mathbf{D},\mathbf{F}\right), \tag{16}$$

$$\min_{\mathbf{D}} \varepsilon\left(\mathbf{D},\mathbf{F}\right), \tag{17}$$

$$\varepsilon\left(\mathbf{F},\mathbf{G}\right) = \mathbb{E}_{(x_i^s,y_i^s)\sim\mathcal{D}^s} \mathcal{L}\left(\mathbf{G}\left(\mathbf{F}\left(x_i^s\right)\right), y_i^s\right), \tag{18}$$

$$\varepsilon\left(\mathbf{D},\mathbf{F}\right) = -\mathbb{E}_{x_i^s\sim\mathcal{D}^s} \log\left[\mathbf{D}\left(f_i^s\right)\right] - \mathbb{E}_{x_j^t\sim\mathcal{D}^t} \log\left[1-\mathbf{D}\left(f_j^t\right)\right], \tag{19}$$

where $\mathcal{L}(\cdot,\cdot)$ is Cross-Entropy (CE) loss, and f_i^s and f_j^t are feature representations through $\mathbf{F}(\cdot)$ of source and target samples, respectively. We can simplify Eq. (16) with a negative sign, which can be rewritten as:

$$\min_{\mathbf{F},\mathbf{G}} \varepsilon\left(\mathbf{F},\mathbf{G}\right) - \varepsilon\left(\mathbf{D},\mathbf{F}\right). \tag{20}$$

Although DANN achieves excellent unsupervised classification performance by adapting the feature representation, it may be insufficient only to adapt feature representation due to the nature of multi-class classification. CDAN [18] shows that the prediction of $\mathbf{G}(\cdot)$ also conveys discriminative information, so we simultaneously adapt domain variances in both feature representation and classifier prediction. The above minimax optimization problem can be defined by conditioning classifier prediction on feature representation, which is expressed as:

$$\min_{\mathbf{F},\mathbf{G}} \varepsilon\left(\mathbf{F},\mathbf{G}\right) - \varepsilon\left(\mathbf{D},\mathbf{F},\mathbf{G}\right), \tag{21}$$

$$\min_{\mathbf{D}} \varepsilon\left(\mathbf{D},\mathbf{F},\mathbf{G}\right), \tag{22}$$

$$\varepsilon\left(\mathbf{D},\mathbf{F},\mathbf{G}\right) = -\mathbb{E}_{x_i^s\sim\mathcal{D}^s} \log\left[\mathbf{D}\left(f_i^s, g_i^s\right)\right] - \mathbb{E}_{x_j^t\sim\mathcal{D}^t} \log\left[1-\mathbf{D}\left(f_j^t, g_j^t\right)\right], \tag{23}$$

where g is the classifier predicted logits. To capture interactions between f and g, we use Randomized Multilinear Conditioning [18] (RMC) to embed (f,g) into reproducing Hilbert spaces, which can be defined as:

$$\mathrm{RMC}(f,g) = \frac{1}{\sqrt{d_r}}\left(\mathbf{R}_f f\right) \odot \left(\mathbf{R}_g g\right), \tag{24}$$

where $\odot$ is element-wise product, $\mathbf{R}_f$ and $\mathbf{R}_g$ are learnable matrices, and d_r is resulting dimension. So Eq. (23), called adversarial loss, can be rewritten as:

$$\varepsilon\left(\mathbf{D},\mathbf{F},\mathbf{G}\right) = -\mathbb{E}_{x_i^s\sim\mathcal{D}^s} \log\left[\mathbf{D}\left(\mathrm{RMC}\left(f_i^s, g_i^s\right)\right)\right] - \mathbb{E}_{x_j^t\sim\mathcal{D}^t} \log\left[1-\mathbf{D}\left(\mathrm{RMC}\left(f_j^t, g_j^t\right)\right)\right]. \tag{25}$$

The loss function ℓ_{DAL} of proposed DAL can be defined as:

$$\max_{\mathbf{D}} \min_{\mathbf{F},\mathbf{G}} \ell_{DAL} = \max_{\mathbf{D}} \min_{\mathbf{F},\mathbf{G}} \mathbb{E}_{(x_i^s,y_i^s)\sim\mathcal{D}^s} \mathcal{L}\left(\mathbf{G}(\mathbf{F}(x_i^s)), y_i^s\right) + \alpha\left(\mathbb{E}_{x_i^s\sim\mathcal{D}^s} \log\left[\mathbf{D}\left(\mathrm{RMC}\left(f_i^s, g_i^s\right)\right)\right] + \mathbb{E}_{x_j^t\sim\mathcal{D}^t} \log\left[1-\mathbf{D}\left(\mathrm{RMC}\left(f_j^t, g_j^t\right)\right)\right]\right), \tag{26}$$

where α is a weight hyper-parameter. In the training stage, we adopt Gradient Reversal Layer (GRL) [17] strategy to achieve end-to-end DAL with a single feed-forward network and standard back-propagation.

2) Similarity-Matrix-Based Contrastive Learning: To enable intra-class alignment, ACDTM explores sample-level SMCL to enhance transfer process. ACDTM adopts a k-Nearest-Neighbor (kNN) based ratio test technique to construct a sample-level similarity matrix, yielding multiple positives and negatives to compute contrastive loss. Next, we detail the construction of similarity matrix and its contrastive loss.

Similarity Matrix Construction: For each target domain sample x_j^t, ACDTM uses metric function $sim(\cdot,\cdot)$ to calculate and rank similarity with source domain samples, and the $sim(\cdot,\cdot)$ we adopt common normalized inverse Euclidean distance [46]:

$$sim(f_a, f_b) = \frac{1}{1 + ||f_a - f_b||^2}, \tag{27}$$

where f_a and f_b are feature representations of samples through $\mathbf{F}(\cdot)$, respectively. Then, x_j^t is defined as a most common class (denoted as $\hat{y}_j$) based on majority voting resulting, so the elementary similarity matrix can be defined as:

$$\mathbf{A}_{ij} = \begin{cases} 1, & \text{if } y_i = \hat{y}_j \\ -1, & \text{if } y_i \neq \hat{y}_j. \end{cases} \tag{28}$$

However, the elementary similarity matrix also is easy to introduce noise, so we need to filter possible noise labels for matrix $\mathbf{A}$. ACDTM adopts the typical neighborhood similarity ratio test [47] based filtering method. Assume the set of like source domain sample of x_j^t is $N_j^l = \{x_i^s | y_i = \hat{y}_j\}$, and the set of unlike samples is $N_j^u = \{x_i^s | y_i \neq \hat{y}_j\}$. For x_j^t, we calculate its label predict confidence Ω_j by using the ratio of sum similarity metric between like and unlike sets:

$$\Omega_j = \frac{\sum_{x_i^s \in N_j^l} sim(\mathbf{F}(x_j^t), \mathbf{F}(x_i^s))}{\sum_{x_i^s \in N_j^u} sim(\mathbf{F}(x_j^t), \mathbf{F}(x_i^s))}. \tag{29}$$

According to predict confidences, we select top μ target domain samples that are receivable, and the corresponding values in $\mathbf{A}$ of the rest of the sample are set to 0 which is regarded as noisy. In matrix $\mathbf{A}$, the "+1" and "-1" respectively represent the positions of positive and negative sample pairs, which can be used to calculate the following contrastive loss. During the training stage, we adopt balanced sampling for source domain, for example, the source domain has K class, thus we sample $K \times num$ samples in each batch, where num is samples of each class. This has the benefit of preventing classes from being undersampled on source domain, which may make computing the similarity matrix invalid. Since target domain is unlabeled, a random sampling strategy is adopted.

Contrastive Loss: Contrastive learning is widely used in self-supervised learning [48], [49], which aims to pull similar samples closer while pushing away dissimilar samples. Self-supervised learning uses the augmented versions of a sample to serve as positive pairs and other samples are negative pairs. Since the matrix $\mathbf{A}$ has obtained the positive and negative pairs, we can calculate contrastive loss to promote intra-class alignment. That is:

$$\Psi^+ = \sum_{\{x_j^t | A_{ij}=1\}} \exp^{sim\left(\mathbf{F}(x_i^s), \mathbf{F}(x_j^t)\right)}, \tag{30}$$

Algorithm 1: ACDTM Training.

input : Labeled source domain $\mathcal{D}^s$, unlabeled target domain $\mathcal{D}^t$, training epochs $epoch$, source batch size B_S, target batch size B_T, feature extraction $\mathbf{F}$, classifier $\mathbf{G}$, discriminator $\mathbf{D}$
output: Model parameters $\Theta_{\mathbf{F}}$, $\Theta_{\mathbf{G}}$, $\Theta_{\mathbf{D}}$

Initial model parameters $\Theta_{\mathbf{F}}$, $\Theta_{\mathbf{G}}$, and $\Theta_{\mathbf{D}}$ for $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{D}$, respectively;
for *ep* ← *1 to epoch* **do**
  Balanced sample $\{(x_i^s, y_i)\}_{i=1}^{B_S}$ form $\mathcal{D}^s$;
  Random sample $\{x_j^t\}_{j=1}^{B_T}$ form $\mathcal{D}^t$;
  for $i \leftarrow 1$ *to* B_S **do**
    $f_i^s \leftarrow \mathbf{F}(x_i^s)$, $g_i^s \leftarrow \mathbf{G}(f_i^s)$;
  end
  for $j \leftarrow 1$ *to* B_T **do**
    $f_j^t \leftarrow \mathbf{F}(x_j^t)$, $g_j^t \leftarrow \mathbf{G}(f_j^t)$;
    $\hat{y}_j \leftarrow \text{kNN}(f_j^t, f_{i:B_S}^s)$;
    for $i \leftarrow 1$ *to* B_S **do**
      if $y_i = \hat{y}_j$ **then** $\mathbf{A}_{ij} = 1$;
      else $\mathbf{A}_{ij} = -1$;
    end
    Calculate Ω_j by Eq. (29);
  end
  `// Update A by ratio test`
  $\mathbf{A} \leftarrow \text{Update}(\mathbf{A}, \Omega, \mu)$;
  Conduct AST on unlabeled target domain and calculate ℓ_{AST} by Eq. (36);
  Calculate total loss ℓ_{total} by Eq. (37);
  `// Update model parameters by gradient descent optimizer`
  $\Theta_{\mathbf{F}} \leftarrow \text{optimizer}(\Theta_{\mathbf{F}}, \ell_{total})$;
  $\Theta_{\mathbf{G}} \leftarrow \text{optimizer}(\Theta_{\mathbf{G}}, \ell_{total})$;
  $\Theta_{\mathbf{D}} \leftarrow \text{optimizer}(\Theta_{\mathbf{D}}, \ell_{total}, \text{GRL})$;
end
return $\Theta_{\mathbf{F}}$, $\Theta_{\mathbf{G}}$, $\Theta_{\mathbf{D}}$

$$\Psi^- = \sum_{\{x_j^t | A_{ij}=-1\}} \exp^{sim\left(\mathbf{F}(x_i^s), \mathbf{F}(x_j^t)\right)}, \tag{31}$$

$$\min_{\mathbf{F}} \ell_{CL} = \min_{\mathbf{F}} -\mathbb{E}_{(x_i^s, x_j^t) \sim (\mathcal{D}^s, \mathcal{D}^t)} \log \frac{\Psi^+}{\Psi^+ + \Psi^-}. \tag{32}$$

Our experiments show that after adding contrastive loss, the recognition performance improves by large margins.

3) Adversarial Self-Training: To fully exploit the information in the unlabeled target domain, self-training is a common approach [15], [50]. This method involves assigning pseudo-labels to samples with high confidence and then proceeding with supervised learning [51]. For unlabeled target domain samples, its self-training loss ℓ_{st} can be defined as:

$$g_j^t = \mathbf{G}\left(\mathbf{F}\left(x_j^t\right)\right), \quad \hat{g_j^t} = \arg\max\left(g_j^t\right), \tag{33}$$

$$\ell_{st} = \min_{\mathbf{F},\mathbf{G}} \mathbb{E}_{x_j^t \sim \mathcal{D}^t} \mathbb{1}\left(\max(g_j^t) \geq \tau_c\right) \mathcal{L}\left(g_j^t, \hat{g_j^t}\right), \tag{34}$$

where hyper-parameter τ_c is confidence threshold, such as 0.9. However, erroneous pseudo-labels can introduce noise and hinder the model's learning capabilities. The work [52] demonstrates that adversarially training the model when data contain some incorrect pseudo-labels leads to a tighter generalization error bound compared to standard self-training methods. Therefore, to mitigate the disruption caused by incorrect pseudo-labels, we implement AST in the target domain.

Adversarial training [53] is a classifier regularization technique utilized to enhance the robustness of a model. During adversarial training, samples are perturbed with small disturbances, such as perturbations along the gradient ascent direction [53], to help the model improve its generalization. Our AST generates adversarial perturbations r_{adv} by adding perturbations along the gradient ascent direction and normalizing them, which can be defined as:

$$\delta = \nabla_{x^t}\left(\ell_{st}, x^t\right), \quad r_{adv} = \zeta \frac{\delta}{||\delta||_2}, \tag{35}$$

where ∇ is gradient operator, ζ is perturbation weight, and $||\cdot||$ is L_2 norm. Hence, our loss function of AST can be defined as:

$$\ell_{AST} = \min_{\mathbf{F},\mathbf{G}} \mathbb{E}_{x_j^t \sim \mathcal{D}^t} \mathcal{L}\left(\mathbf{G}\left(\mathbf{F}\left(x_j^t + r_{adv}\right)\right), \hat{g_j^t}\right). \tag{36}$$

In summary, the total loss of ACDTM is:

$$\max_{\mathbf{D}} \min_{\mathbf{F},\mathbf{G}} \ell_{DAL} + \beta \ell_{CL} + \ell_{AST}, \tag{37}$$

where β is the weight of contrastive loss. The architecture of $\mathbf{F}$ can be classical Convolutional Neural Networks (CNN), such as LeNet [20] and ResNet [54]. The architectures of $\mathbf{D}$ and $\mathbf{G}$ are "Fully Connected (FC) layer $\rightarrow$ BN [55] $\rightarrow$ ReLU [56] $\rightarrow$ FC $\rightarrow$ BN $\rightarrow$ ReLU $\rightarrow$ FC $\rightarrow$ Sigmoid [57]" and "FC $\rightarrow$ BN $\rightarrow$ ReLU $\rightarrow$ FC", respectively. The pseudo-code of our ACDTM training process is presented in Algorithm 1. During the inference stage, simply input the test samples into the trained feature extraction and classifier.

V. Experiments

In this Section, we comprehensively evaluate our proposed approach. First, we describe our collected dataset and experimental setting in Section V-A. Second, we introduce the baselines in Section V-B. Third, overall performance of our approach is presented in Section V-C. Finally, Section V-D and V-E give ablation and parameter studies, respectively.

A. Datasets and Settings

Datasets: We invite a group of 23 volunteers to collect radar gesture data. The ages of these 23 volunteers range between 20 and 35, with their heights and weights detailed in Fig. 7. We collected two typical mmWave gesture datasets, consisting of 10 digit gestures and 26 alphabet gestures. Each individual performs standard hand gestures in the air, such as "Draw-0" to "Draw-9" and "Draw-a" to "Draw-z", and data collection is conducted with the consent of all participants. In our experiments, the subjects' hands were positioned 40$\sim$ 100cm from the radar at angles ranging from -30 to 30 degrees. Our data collection activities occur across three distinct environments, including a meeting room, office, and corridor, as depicted in Fig. 8. Each volunteer performs each hand gesture more than 5 times. We refer to the collected digit gesture dataset as *mmDigit* and the collected alphabet gesture dataset as *mmLetter*. mmDigit comprises 6,196 samples, totaling over 200,000 frames, with a dataset size exceeding 300 GB. mmLetter comprises 6,500 samples, with a dataset size exceeding 300 GB. A summary of these two datasets is provided in Table I. To comprehensively evaluate our approach's performance within domain and across domains, we assess the unsupervised recognition accuracy under three conditions: in-domain, cross-environment, and cross-user. The division of training and testing sets under the three scenarios is as follows:

- *In-domain:* All samples are split into training and testing sets in a 7:3 ratio, without distinguishing between users and environments.

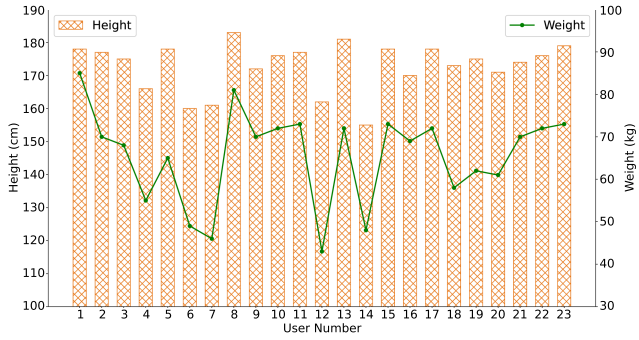
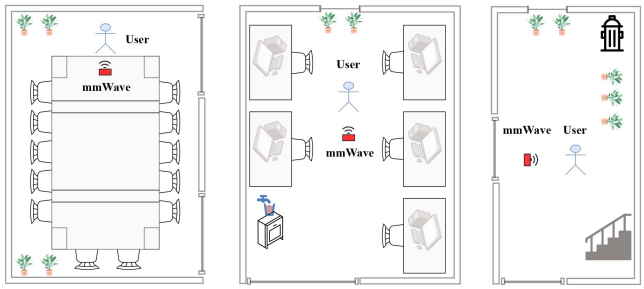


Fig. 7. Users' heights and weights.



(a) Meeting room (b) Office (c) Corridor

Fig. 8. Experiment environments.

TABLE I
Dataset Descriptions (Room1: Meeting Room. Room2: Office. Room3: Corridor).

Dataset	Environments	User number	Samples	Total samples	Memory size
mmDigit	Room1	1$\sim$4,6$\sim$9,11$\sim$16,18,20,23	2096	6196	> 300GB
	Room2	1,5$\sim$7,10$\sim$14,16$\sim$23	2000		
	Room3	1$\sim$3,5,6,8$\sim$10,12,13,17$\sim$20	2100		
mmLetter	Room1	1$\sim$5,7$\sim$9,12$\sim$18,19,22	2080	6500	> 300GB
	Room2	2$\sim$6,9$\sim$15,18$\sim$23	2340		
	Room3	1,5$\sim$8,10$\sim$13,15$\sim$17,20$\sim$23	2080		

TABLE II
IWR 1843BOOST RADAR PARAMETERS IN OUR SETTINGS

Radar Parameters	Values	Radar Parameters	Values
Physical Transmitters	2	Sample Rate	5000 ksps
Physical Receivers	4	Chirp Duration	72 us
Frequency Slope	68.654 MHZ	Chirps in Frame	128
Frames	48	ADC Samples	256

- *Cross-environment:* Data from two environments are used as the training set, while data from the remaining environment is utilized as the testing set. This cross-environment experiment can be conducted three times for each combination of environments, with the results averaged over the three trials.
- *Cross-user:* Fourteen users' data is randomly selected for the training set, while the data from the remaining users is designated as the testing set. This user randomization process is repeated five times, and the experimental results are averaged.

The mmWave radar samples in the target domain are unlabeled. For the source domain data, we leverage the publicly accessible MNIST and EMNIST [22] datasets.

Settings: The radar operates within the frequency range of 77 GHz to 81 GHz. For the experiment, we employed two transmitting antennas and four receiving antennas. The specific parameters of the radar are listed in Table II. In the gesture trajectory generation algorithm, we set the parameters $H_{edge} = 24$, $W_2 = 28$, $H_2 = 28$, $A_S = 32$, $V_D = 36$. For our deep learning framework, we opt for PyTorch [58]. The NVIDIA RTX2080 Ti GPU is used as the hardware device. We utilized the Stochastic Gradient Descent (SGD) optimizer with momentum of 0.9, and the training epoch is set to 100. The learning rate is adjusted using the cosine warm-up strategy [59], [60], [61], [62], with a weight decay of 1e-3. In the case of balanced sampling in the source domain, the number of samples per class is set to 20, namely $num = 20$. mmDigit has 10 classes, namely $K = 10$. Similarly, $K = 26$ on mmLetter. The batch sizes of mmDigit and mmLetter are 200 and 520, respectively.

B. Baseline

In the current landscape of radar sensing, unsupervised domain adaptation is typically performed within homogenous domain settings, such as transferring radar signal data from environment A to environment B, necessitating labeling of data in environment A. To facilitate a fair comparison, we select typical unsupervised domain adaptation works in wireless sensing, UDARF [5] and SALIENCE [63], and utilize their deep domain transfer models on our dataset. Additionally, as the pioneering work advocating the use of heterogeneous images for unsupervised mmWave gesture recognition, we also compared our approach to several common unsupervised domain adaptation algorithms that proposed in the text or image domains [16], [17], [18], [64], [65], [66], [67], such as DANN [17], JAN [65] and MCD [64]. These compared methods vary solely in the deep domain transfer model, while other steps follow our method. The brief introductions of these baselines are as:

Fig. 9. Trajectory visualization examples of mmDigit.

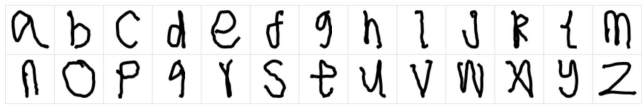


Fig. 10. Trajectory visualization examples of mmLetter.

- *Only Source:* Directly applying a model trained on the source domain data to the downstream target dataset.
- *UDARF [5]:* This approach transfers radar gesture signals from environment A to environment B through a pseudo-label-based domain adaptation strategy.
- *SALIENCE [63]:* This method focuses on wearable sensor-based activity recognition, aligning sensor data via domain adaptation.
- *DAN [16]:* This scheme utilizes a multiple kernel variant of Maximum Mean Discrepancies (MMD) to align feature representations.
- *DANN [17]:* This work performs domain adversarial learning on the feature spaces.
- *JAN [65]:* This method learns an adaptation model by aligning joint distributions of the network across domains.
- *CDAN [18]:* This approach builds upon feature adversarial training, incorporating additional conditional adversarial learning.
- *MCD [64]:* This methodology delves into task-specific decision boundaries to harmonize the distributions of the source and target domains.
- *DWL [66]:* This strategy dynamically adjusts the learning losses concerning alignment and discriminability by introducing the measures of alignment and discriminability.
- *CAF [67]:* This technique mitigates the global domain disparities while preserving the local semantic coherence for cross-domain transfer in a collaborative fashion.

All the baselines are unsupervised methods without subsequent fine-tuning processes. To comprehensively evaluate these approaches, we compare in-domain, cross-environment, and cross-user recognition accuracies of each method.

C. Overall Performance

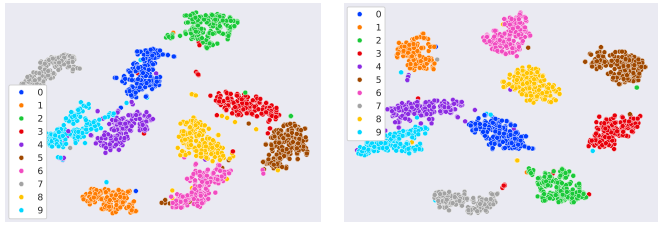
Trajectory Reconstruction Visualization: Fig. 9 and Fig. 10 depict the radar gesture trajectory examples of mmDigit and mmLetter, respectively. It is evident that the trajectory plot effectively captures the gestural trends. Due to the comparatively lower resolution of radar point clouds, the gesture trajectories may lack consistency and exhibit oscillations and fluctuations.

TABLE III
RECOGNITION ACCURACY COMPARISONS WITH BASELINES

Schemes	mmDigit			mmLetter		
	In-domain	Cross-environment	Cross-user	In-domain	Cross-environment	Cross-user
Supervised	0.978	0.970	0.969	0.883	0.879	0.880
Only Source	0.670	0.669	0.678	0.502	0.511	0.507
DAN [16]	0.798	0.785	0.773	0.676	0.662	0.648
DANN [17]	0.809	0.803	0.791	0.698	0.683	0.671
JAN [65]	0.826	0.821	0.816	0.719	0.710	0.686
UDARF [5]	0.848	0.839	0.826	0.732	0.726	0.708
MCD [64]	0.861	0.852	0.841	0.741	0.731	0.715
SALIENCE [63]	0.859	0.853	0.846	0.742	0.730	0.719
CDAN [18]	0.867	0.860	0.855	0.748	0.732	0.723
DWL [66]	0.893	0.884	0.876	0.766	0.757	0.746
CAF [67]	0.912	0.903	0.889	0.780	0.768	0.752
Ours	**0.923**	**0.918**	**0.902**	**0.792**	**0.781**	**0.765**

TABLE IV
MODEL SIZE, NUMBER OF PARAMETERS, AND FLOPS

Schemes	DANN [17]	UDARF [5]	CDAN [18]	DWL [66]	CAF [67]	Ours
Model Size	0.09MB	0.09MB	0.09MB	0.10MB	0.10MB	0.09MB
Model Parameters	0.22M	0.22M	0.22M	0.24M	0.24M	0.23M
FLOPs	0.75M	0.77M	0.76M	0.82M	0.83M	0.84M



(a) only source (b) ours

Fig. 11. t-SNE Visualization (in-domain mmDigit). (a) only source. (b) ours.

Additionally, due to variations in participants' habits or potentially sloppy gestures, some gestures may not be entirely standardized. For instance, in Fig. 9, the last row's 0, 2, 4, etc., exhibit some discrepancies. Consequently, there exist significant disparities in the distribution between the radar trajectories and the open image data. This underscores the necessity of employing our domain transfer model to align the images.

Recognition Accuracy Comparison: In Table III, we compare the accuracy of our proposed approach with the baselines. It is evident that compared to these baselines, our approach achieves better unsupervised performance, approaching the performance of fully supervised learning more closely than they do. First, on the mmDigit, our method achieves an accuracy of over 90% in unsupervised gesture recognition, while the accuracy of the model trained only on the source domain is around 67%. This indicates that our ACDTM significantly improves unsupervised recognition accuracy through domain transfer. Similarly, on the mmLetter, compared to using only the source domain, we can improve unsupervised accuracy by over 26% through domain transfer. Second, our approach outperforms current baselines by significant margins. For example, compared to UDARF, we surpass them by over 7.5%, 7.9%, and 7.6% on in-domain, cross-environment, and cross-user scenarios on mmDigit, respectively. Additionally, compared to unsupervised domain adaptation models in the image and text domains, our method also outperforms 1% accuracy. Third, our method demonstrates similar effectiveness across in-domain, cross-environment, and cross-user scenarios, indicating robustness to different environments and users. By effectively processing the signals and transforming radar data into images, our model becomes less sensitive to variations in environments and users. In summary, our ACDTM exhibits better generalization performance.

Model Computational Costs: Our method aligns open image datasets with radar gesture trajectories through Unsupervised Domain Adaptation (UDA), transferring knowledge from images to radar gestures. Current UDA techniques for radar gesture recognition primarily focus on homogeneous radar data transfer [5]. Therefore, when comparing with these methods, we only adopt their model architectures (e.g., loss functions) while using our trajectory maps and public images as input. Additionally, we compare against advanced UDA models from computer vision [17], [18], [66], [67]. Since both our source and target domains use 28×28 gesture images as input, shallow CNNs (e.g., LeNet [20]) suffice, keeping all models' size, parameters, and computations minimal. Therefore, our approach requires only 0.09 MB model size, 0.23 M parameters, and 0.84M FLOPs computations. Current UDA techniques mainly modify loss functions or adaptation strategies - for instance, DWL [66] dynamically adjusts learning losses for alignment and discriminability, while CAF [67] collaboratively mitigates global domain disparities while preserving local semantic coherence. Consequently, model parameters and computations primarily depend on the backbone. Our adversarial and contrastive losses do not increase model parameters but slightly increase FLOPs. By maintaining the same backbone, our method shows comparable - sometimes even lower - computational overhead than baselines, as shown in Table IV. However, our approach achieves 1%-2% higher accuracy compared to state-of-the-art computer vision models. Overall, our innovative heterogeneous domain transfer between images and radar data reduces annotation costs, improves unsupervised accuracy, and maintains low computational requirements.

Confusion Matrix and t-SNE Visualizations: To showcase the advantages of unsupervised domain transfer, we contrast our approach with the only source domain that does not undergo domain transfer, analyzing confusion matrix results and feature space visualization. We employed t-SNE [46] for visualizing the feature spaces in the in-domain setting on mmDigit, results are shown in Fig. 11. It can be observed that the boundaries between classes in the only source are not as distinct as in our method, and the intra-class distribution is not as compact as in our approach. Our method exhibits clearer boundaries between classes, with greater distances between categories, resulting in a more distinct separation between different classes. Additionally, we present the confusion matrix under the cross-user scenario, as shown in Fig. 12. It can be observed that, compared to the only source, the accuracy significantly improves after applying our ACDTM. For instance, the accuracy of "Draw-0" increases from 84.29 to 95.95, "Draw-3" from 86.19 to 91.67, "Draw-5" from

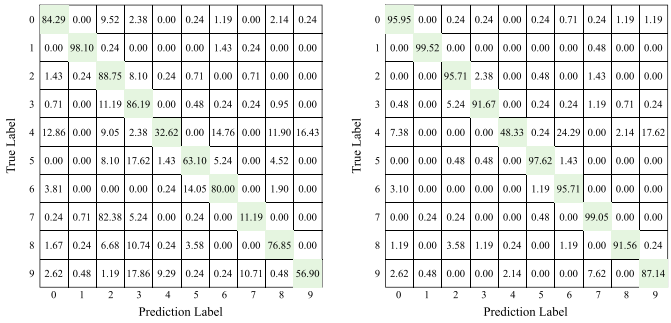


Fig. 12. Confusion matrix (%) results.

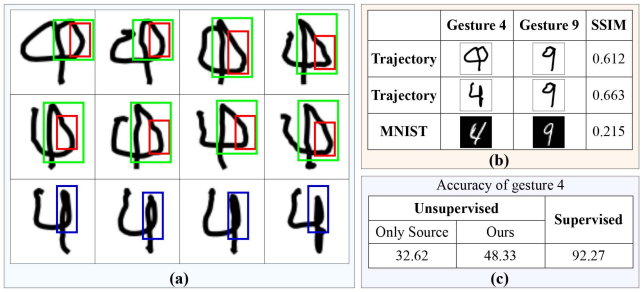


Fig. 13. Analysis and investigation for relatively low recognition accuracy of digit gesture "4".

63.10 to 97.62, "Draw-7" from 11.19 to 99.05, "Draw-8" from 76.85 to 91.56, and "Draw-9" from 56.90 to 87.14. It is noted that the accuracy of "Draw-4" is relatively low, at 32.62 in the only source, which our domain transfer boosts to 48.33. When writing the number 4 in the air, it is written in one stroke, so there is an extra curve in the top right corner compared to a normal 4, making it easy to be mistaken for 6 or 9, resulting in lower recognition accuracy. We explain further the reasons for this phenomenon in the subsequent section. In summary, the t-SNE visualization and confusion matrix provide evidence of the efficacy of our approach, significantly enhancing performance even in the absence of labeled information.

Analysis for Confusion Between Gestures "4" and "9": (1) The continuous writing motion in air gestures causes the right portion of "4" to form a closed loop. Red box in Fig. 13(a) shows the connecting stroke, green box indicates the formed loop. This visual similarity to digit "9" which also contains a loop. So this easily contributes to confusion between "4" and "9". Additionally, users' inconsistent writing distances and the inherent irregularity of radar-tracked gestures often produce ambiguous samples. For instance, some users draw a straight line when writing the right portion of "4" (blue box in Fig. 13(a)), increasing similarity to "9". We calculate Structure Similarity Index Measure (SSIM) between sample trajectories of "4" and "9", with higher values indicating greater similarity. As shown in Fig. 13(b), SSIM consistently exceeds 0.6, confirming their visual resemblance. In contrast, non-cursive "4" and "9" from MNIST exhibit significantly lower SSIM. This demonstrates

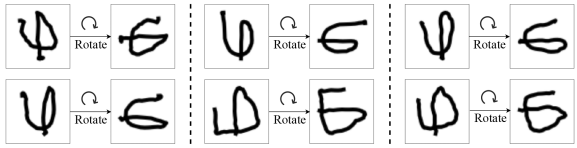


Fig. 14. Gesture "4" exhibits high similarity to gesture "6" after rotation.

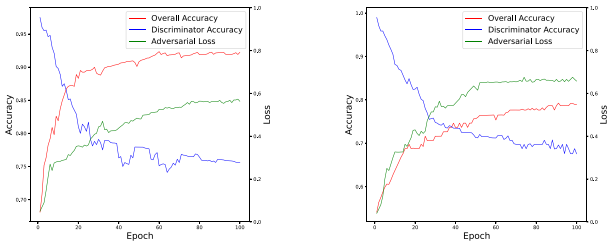
that continuous air writing affects gesture shapes. However, unlike pen-and-paper writing where strokes can be discrete, continuous motion is unavoidable in wireless sensing-based air gesture recognition. (2) As shown in Fig. 11, t-SNE visualization reveals overlapping distributions between "4" and "9", indicating their inherent difficulty for unsupervised models to distinguish. Since our source domain data (MNIST) lacks continuous writing samples, the unsupervised alignment fails to adequately address this pattern. When supervised training is applied to target domain data, gesture "4" recognition improves to 92.2%, as shown in Fig. 13(c). It can be proven that explicit exposure to such data distributions with ground truth labels can resolve this issue. Therefore, future work may incorporate annotated continuous writing samples in source domain training to enhance unsupervised generalization.

Analysis for Confusion Between Gestures "4" and "6": The digit "4" is often written cursively, with a circular stroke on its right side. When rotated by 90 degrees, this circular part appears at the bottom, as indicated in Fig. 14. Since the digit "6" also exhibits a circular stroke at the bottom, the two gestures appear similar. However, as observed in the t-SNE visualization, the distributions of "4" and "6" are not adjacent, which is also normal. This is because the t-SNE visualization does not fully represent the underlying data distribution. First, the geometric structure of neural network feature spaces with hundreds or even thousands of dimensions is highly complex. The proximity relationships in high-dimensional space become distorted when compressed into a two-dimensional plane. Second, the confusion matrix microscopically reflects the model's performance details at decision boundaries, while t-SNE macroscopically reveals the global category structure learned by the model. Consequently, even if two classes appear well-separated in t-SNE, misclassifications may still occur in the confusion matrix, which is a common phenomenon. In future work, we will further investigate methods to distinguish such easily confusable gestures to enhance the robustness of unsupervised recognition.

Trajectory Tracking Error: For gesture trajectory acquisition, we employ standard FMCW signal processing components integrated with filtering and noise removal techniques. Several established radar trajectory reconstruction methods exist, such as mmWrite [68] and mTrack [69]. However, significant differences exist in radar hardware configurations. For instance, mmWrite [68] utilizes a high-resolution 60 GHz radar platform based on modified Qualcomm 802.11ad chips, while mTrack [69] employs a customized 60 GHz system incorporating multiple receivers and mechanically steered antennas. In contrast, our implementation uses a single commercial mmWave radar device without custom components or additional hardware.

TABLE V
TRAJECTORY ERROR COMPARISON WITH MMWRITE [68]. WE REPRODUCE MMWRITE [68] ON OUR DATASET AND HARDWARE PLATFORM. WE INPUT THE RESULTING TRAJECTORIES OF MMWRITE [68] INTO OUR PROPOSED MODEL TO EVALUATE UNSUPERVISED RECOGNITION ACCURACY.

Scheme	mmDigit		mmLetter	
	Tracking Error (↓)	Unsupervised Accuracy (↑)	Tracking Error (↓)	Unsupervised Accuracy (↑)
mmWrite [68]	2.61 cm	0.892	3.69 cm	0.769
Ours	1.63 cm	0.923	2.57 cm	0.792

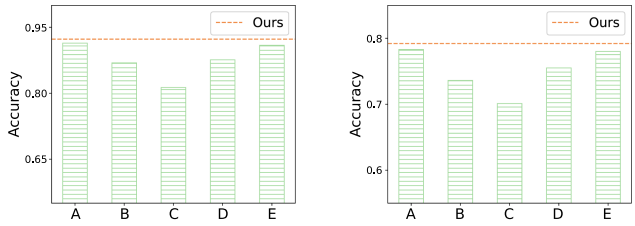


(a) mmDigit (b) mmLetter

Fig. 15. The curves of overall accuracy, domain discriminator accuracy, and domain discriminator loss. (a) mmDigit. (b) mmLetter.

TABLE VI
MODEL CLASS ACCURACY UNDER UNSEEN SCENARIOS. WE REMOVE ONE CLASS DURING TRAINING AND EVALUATE ITS ACCURACY AS AN UNSEEN CLASS DURING TESTING.

Unseen Gesture	0	1	2	3	4	5	6	7	8	9
Only Source	84.29	98.10	88.75	86.19	32.62	63.10	80.00	11.19	76.85	56.90
Ours (Unseen)	91.42	99.23	91.65	89.20	41.59	85.62	90.15	71.63	84.92	77.68
Ours (Seen)	95.95	99.52	95.71	91.67	48.33	97.62	95.71	99.05	91.56	87.14



(a) mmDigit (b) mmLetter

Fig. 16. Signal processing ablation results, deleting one of the processing steps. A: without SOR; B: without APOR; C: without MF; D: without KF; E: without S-G. (a) mmDigit. (b) mmLetter.

Furthermore, mmWrite [68] uses 8×8 antenna array to provide superior resolution compared to our 2×4 configuration. Consequently, compared to these customized high-resolution hardware platforms, our approach necessitates a more tightly coupled signal processing pipeline to enhance trajectory quality. For example, in the intra-frame processing, SOR and energy-weighted averaging are employed for point cloud denoising. In the inter-frame processing, APOR, MF, KF, and S-G smooth are employed for trajectory filtering. We conduct quantitative comparisons with existing mmWave tracking solutions. Due to the requirement of additional mechanical components in mTrack [69], we compare against the classical radar gesture trajectory reconstruction method mmWrite [68]. mmWrite [68] employs different hardware configurations, we apply their methodology solely to our dataset and feed the resulting trajectories into our proposed model to evaluate unsupervised recognition accuracy. We compare in-domain performance, with results presented in Table V. Our approach demonstrates a smaller average point error of trajectory reconstruction and higher unsupervised recognition accuracy, indicating that our intra-frame and inter-frame signal processing and denoising techniques enhance trajectory image robustness.

Domain Discriminator Loss and Accuracy: DAL confuses the source and target domains through a min-max game to learn domain-invariant features. To examine the process of domain confusion, we plot the curves of the discriminator accuracy and overall classification accuracy on mmDigit and mmLeter, as shown in Fig. 15. It can be observed that the discriminator accuracy decreases gradually, indicating that the model starts to have difficulty distinguishing between the source and target domains as their distributions become increasingly similar. As the model learns domain-invariant features, the overall unsupervised accuracy gradually improves. Additionally, it can be seen from Fig. 15 that the adversarial loss increases corresponding to the changes in discriminator accuracy. This result also indicates that our adoption of DAL is an effective approach.

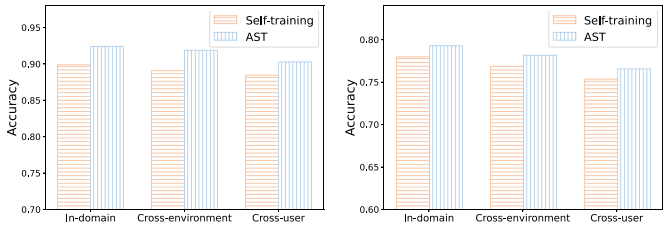
Unseen Class Performance: To verify that our model robustly transfers knowledge from images to radar gestures rather than simply performing inter-class distribution mapping, we remove one target domain class during training and evaluate its accuracy as an unseen class during testing. We assess this on digit gestures by sequentially removing classes 0∼9 to construct unseen class scenarios. As shown in Table VI, the model still achieves strong performance on unseen classes, significantly outperforming the only source baseline. For example, when gesture "0" is unseen, the model attains 91.42% accuracy, which achieves a 7.13% improvement over the only source. However, performance remains slightly lower compared to when all classes are available during training, such as 95.95% accuracy with gesture "0", demonstrating that more data can enhance model generalization. Table VI shows that our model can effectively transfer knowledge from images to radar gestures even under unseen scenarios. We believe there is still room to improve recognition of unseen classes in unsupervised domain adaptation, and we may explore this further in future work.

D. Ablation Study

Signal Processing Ablation: Within trajectory generation, a series of denoising and smoothing techniques such as SOR, APOR, MF, KF, and S-G methods are applied to signals both intra-frame and inter-frame. These methodologies aid in producing cleaner gesture trajectory images, directly impacting subsequent unsupervised recognition accuracy. By removing a single signal processing module, we observe how the results change. The ablation results on mmDigit and mmLetter for in-domain scenario are depicted in Fig. 16. It is evident that each denoising and smoothing technique contributes to enhancing model performance, notably APOR, MF, and KF, which

TABLE VII
ACDTM MODULE ABLATION RESULTS

Module	mmDigit			mmLetter		
	In-domain	Cross-environment	Cross-user	In-domain	Cross-environment	Cross-user
RDMap	0.105	0.106	0.099	0.036	0.038	0.029
+Trajectory	0.670	0.669	0.678	0.502	0.511	0.507
+DAL	0.867	0.860	0.855	0.748	0.732	0.723
+SMCL	0.889	0.878	0.871	0.757	0.744	0.733
+AST	0.923	0.918	0.902	0.792	0.781	0.765



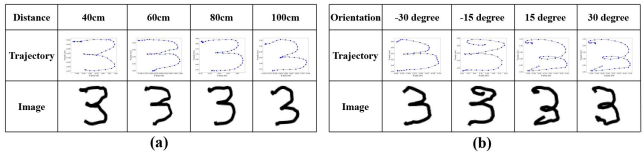
(a) mmDigit (b) mmLetter

Fig. 17. Performance comparisons of AST and self-training. (a) mmDigit. (b) mmLetter.

respectively increased by about 5%, 10%, and 4%. This indicates that pristine data input is fundamental, and our denoising and smoothing procedures significantly enhance data robustness and model generalization.

ACDTM Module Ablation: We utilize trajectory images as an intermediate bridge to mitigate the heterogeneity between radar signals and images. Here, we directly transfer the pre-trained model from source domain to radar signals (e.g., RDMap), and results are shown in Table VII. It can be seen that the accuracy is only about 10% and 3% on mmDigit and mmLetter, respectively, which is close to random guessing. This indicates that the heterogeneity between images and radar has a significant negative impact on the transfer. Therefore, we choose to convert radar signals into trajectory images, achieving an accuracy of over 65% and 50% on mmDigit and mmLetter, respectively. Due to the characteristics of radar point clouds, there is still a significant disparity between the distribution of trajectory and open images. To address this, we employ the proposed ACDTM to align features, including DAL, SMCL, and AST modules. We added these three modules one by one, and the results are shown in Table VII. It can be seen that each module significantly improves recognition accuracy. For example, after adding DAL on trajectory images, the accuracy is improved by over 15%. Continuing to add SMCL, the accuracy can increase by approximately 1∼2%. After adding AST, the accuracy can increase by around 4%. The ablative experiments of ACDTM demonstrate the effectiveness of our approach.

AST vs. Self-Training: In ACDTM, AST is utilized to explore more information from unlabeled target domains instead of directly employing self-training. The self-training process may involve incorrect pseudo-labels, whereas adversarially training the model when data contains some erroneous pseudo-labels results in a tighter generalization error bound. We compared the effectiveness of AST and direct self-training, as shown in Fig. 17. It can be observed that AST achieves better performance.



(a) (b)

Fig. 18. Trajectories and corresponding generated images with different distances and orientations. (a): Different distance. (b): Different orientations.

TABLE VIII
MODEL PERFORMANCES ACROSS DIFFERENT DISTANCES AND ORIENTATIONS

Datasets	mmDigit		mmLetter	
	Different distances	Different orientations	Different distances	Different orientations
Accuracy	0.919	0.905	0.786	0.774

TABLE IX
MODEL PERFORMANCE WITH VARYING DATASET SIZES

Dataset Size	mmDigit			mmLetter		
	In-domain	Cross-environment	Cross-user	In-domain	Cross-environment	Cross-user
30% Ratio	0.849	0.838	0.819	0.706	0.687	0.664
50% Ratio	0.892	0.887	0.870	0.752	0.748	0.726
80% Ratio	0.911	0.904	0.889	0.780	0.769	0.751
100% Ratio	0.923	0.918	0.902	0.792	0.781	0.765

Trajectory with Different Distances and Orientations: As shown in Fig. 18, we illustrate trajectory images generated at various distances (e.g., 40 cm, 60 cm, 80 cm, 100 cm) and angles (e.g., -30 degrees, -15 degrees, 15 degrees, 30 degrees), demonstrating relatively complete trajectories that highlight our method's robust adaptability. To evaluate the model's accuracy performance across varying distances and directions, we collected approximately 300 gesture samples from a user at different distances and orientations. The unsupervised model tested is operated in the trained in-domain mode. The results are presented in Table VIII. It can be seen that the accuracy of our unsupervised model under different distances and orientations remains close to that achieved in the in-domain setting, such as digit gesture recognition accuracy consistently exceeding 90%. This demonstrates that by converting radar signals into trajectory images, our unsupervised model achieves robustness to gestures performed at varying distances and orientations.

Different Dataset Sizes: To evaluate the impact of dataset size on model performance, we tested our unsupervised recognition approach using 30%, 50%, and 80% of the dataset. The results in Table IX demonstrate that: (1) Increasing the dataset size improves generalization performance, indicating that data volume affects model robustness. (2) With 50% of the data, the model already achieves strong unsupervised performance. For instance, on the mmDigit dataset, it attains an in-domain accuracy of 0.892, approaching the performance of the full dataset. (3) Further increasing the dataset to 80% and 100% yields marginal gains. For example, expanding from 50% to 80% only improves accuracy by approximately 2%, and from 80% to 100% just improves accuracy by around 1%. This suggests that once trained on a sufficient data volume, our model achieves robust generalization without heavily relying on additional data.

TABLE X
COMPARISON RESULTS OF MAJORITY-VOTING-BASED AND HIGHEST-SIMILARITY-BASED METHODS IN SMCL

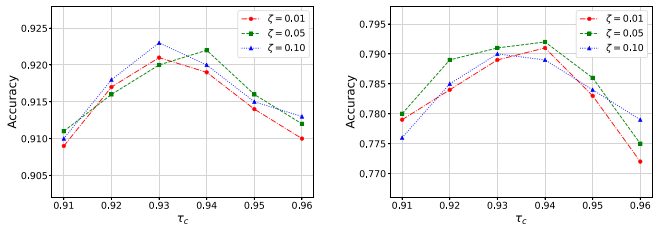
	mmDigit			mmLetter		
	In-domain	Cross-environment	Cross-user	In-domain	Cross-environment	Cross-user
Highest Similarity	0.917	0.914	0.889	0.788	0.775	0.761
Majority Voting	0.923	0.918	0.902	0.792	0.781	0.765

TABLE XI
HYPER-PARAMETERS α AND β EXPERIMENTS ON MMDIGIT (IN-DOMAIN / CROSS-ENVIRONMENT / CROSS-USER)

α \ β	0.1	0.2	0.3	0.4
0.1	0.918 / 0.909 / 0.898	0.922 / 0.916 / 0.899	0.920 / **0.918** / 0.901	0.915 / 0.907 / 0.896
0.2	0.920 / 0.912 / 0.895	**0.923** / 0.917 / 0.900	0.918 / 0.912 / **0.902**	0.918 / 0.908 / 0.895
0.3	0.915 / 0.911 / 0.891	0.919 / 0.910 / 0.897	0.919 / 0.913 / 0.896	0.911 / 0.903 / 0.890

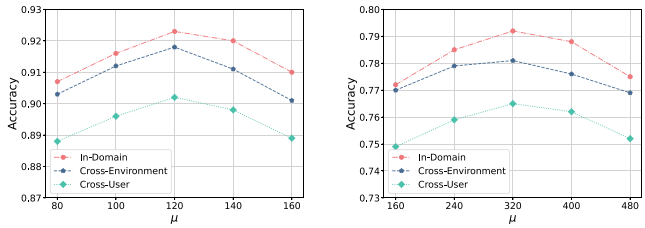
TABLE XII
HYPER-PARAMETERS α AND β EXPERIMENTS ON MMLETTER (IN-DOMAIN / CROSS-ENVIRONMENT / CROSS-USER)

α \ β	0.1	0.2	0.3	0.4
0.1	0.789 / 0.772 / 0.756	0.788 / 0.779 / **0.765**	**0.792** / 0.772 / 0.763	0.783 / 0.769 / 0.762
0.2	0.785 / 0.776 / 0.755	0.790 / **0.781** / 0.761	0.789 / 0.773 / 0.760	0.786 / 0.772 / 0.758
0.3	0.780 / 0.771 / 0.754	0.783 / 0.774 / 0.757	0.782 / 0.775 / 0.754	0.781 / 0.769 / 0.755



(a) mmDigit (b) mmLetter

Fig. 19. Sensitivity of hyper-parameters τ_c and ζ. (a) mmDigit. (b) mmLetter.



(a) mmDigit (b) mmLetter

Fig. 20. Sensitivity of hyper-parameter μ. (a) mmDigit. (b) mmLetter.

Majority-Voting-Based Approach in SMCL: When constructing the similarity matrix in SMCL, we define the class of a target-domain sample as the most frequent class among its k-nearest neighbors. This majority-voting-based method statistically mitigates uncertainty and enhances model generalization. We compare this approach with the highest-similarity-score method, with experimental results presented in Table X. The majority-voting strategy achieves slightly higher performance than the highest-similarity method, demonstrating its superior robustness.

E. Parameters Study

Hyper-Parameters α and β: In our loss function, there are two hyper-parameters: the weights α and β. We select different loss weights to observe the impact of these parameters on recognition performance. The experimental results of mmDigit and mmLetter are shown in Table XI and Table XII, respectively. It can be observed that the optimal hyper-parameters vary for different dataset settings. For instance, when $\alpha = 0.3$ and $\beta = 0.1$, mmDigit performs better in the cross-environment setting, while in the cross-user setup, the optimal parameters are $\alpha = 0.3$ and $\beta = 0.2$. However, the performance is not significantly influenced by varying α and β values, demonstrating the robustness of our model to these parameters.

Hyper-Parameters τ_c and ζ: In AST, there are two hyper-parameters, the threshold τ_c and the perturbation weight ζ. Under in-domain settings, we demonstrate the impact of different τ_c and ζ values on performance for mmDigit and mmLetter, as shown in Fig. 19. It can be seen that setting τ_c too small or too large can slightly decrease performance. A too small τ_c reduces the threshold, potentially leading to more noisy pseudo-labels. Conversely, a too large threshold may result in many samples without pseudo-labels, losing some data information. The perturbation weight $\zeta = 0.1$ is more suitable for mmDigit, while $\zeta = 0.05$ is better for mmLetter, and different values of ζ have a minor impact on performance.

Hyper-Parameters μ: In SMCL, we select the top μ samples used for constructing the similarity matrix in a batch. Namely, the maximum value of μ is equal to batch size. In Fig. 20, we explore different μ values impacts on recognition performance, and results show that the performance is optimal when μ is set to 120 and 320 on mmDigit and mmLetter, respectively. The sample parameter μ is used to filter the predict confidence (see Eq. (29)), so a high μ value (e.g., 160 on mmDigit) can introduce excessive noise, while a small μ value (e.g., 80 on mmDigit) may result in the loss of useful samples.

VI. DISCUSSION

There is still room for improvement in our work, and we give some directions to consider.

Multi-Source Domains: Current mmWave-based gesture recognition research typically focuses on a few classes or a specific type of gesture. Similarly, in our work, we test numeric and letter gestures separately. However, in practical applications, when a wider range of gestures is needed, such as a fusion of numeric and letter gestures, it becomes necessary to incorporate multiple publicly available images into the source domain. These publicly available letter and numerical gesture images inherently possess significant distribution biases, thus introducing challenges when training with multi-source domains [70]. In future work, we will address this practical factor and explore corresponding methods to tackle it.

Other RF Device: The wireless gesture signal collection of our approach is under the mmWave radar. Our approach can be

applied to mmWave radar-based gesture recognition. We propose an unsupervised radar gesture recognition model that can be readily integrated into other gesture recognition systems. For example, when a radar gesture recognition system lacks labeled data, our method can be employed. For gesture recognition using other RF devices such as WiFi, in theory, our method can also be applied for unsupervised heterogeneous transfer learning. As long as the WiFi device can describe the gesture trajectory image, our deep domain adaptation model can be applied. Fortunately, the reconstruction of gesture trajectories based on WiFi signals is theoretically feasible [71]. Thus, our work not only constitutes a standalone gesture recognition system but can also be directly incorporated into other gesture recognition frameworks to address data annotation challenges and enable unsupervised gesture recognition. In the future, we will conduct experiments using multiple RF devices (WiFi, RFID, etc.) and design a unified RF unsupervised gesture recognition approach, allowing the model to be trained once and applied across different RF devices.

More Gestures and 3D Trajectory-based Alignment: We employ classic digit and letter gestures as examples to demonstrate our method's effectiveness, as these gesture categories are commonly used in RF-based gesture recognition [8], [71], [72], [73], [74]. Our approach remains equally applicable to other gesture types (e.g., circles, triangles, push-pull motions) since they can theoretically be converted into trajectory maps and matched with publicly available image datasets for unsupervised recognition. Compared to these simpler gestures, digits and letters exhibit finer granularity, making them more representative test cases. Additionally, our selection of 2D trajectory rather than 3D is motivated by: (1) Natural alignment with open-source image datasets. Mainstream public gesture datasets primarily contain 2D images. Our approach achieves unsupervised radar gesture recognition by aligning open image gesture datasets, requiring radar data conversion into structurally compatible inputs for effective transfer. (2) Hardware constraints. Our radar configuration, 2 transmitter and 4 receiver antennas, only provides azimuthal information without elevation data, limiting us to 2D trajectory generation. Consequently, 3D trajectory alignment presents substantial practical challenges. In the future, we will explore 3D trajectory-based alignment and develop corresponding methods.

Video-based Alignment: The gesture actions have a temporal component, yet to align images, we transform temporal gestures into two-dimensional trajectories, potentially sacrificing their temporal nature. Gesture movements captured by cameras typically involve video data, retaining the temporal sequence of gestures. Therefore, in future endeavors, we will investigate aligning radar gestures with video data to facilitate unsupervised domain transfer.

Synthetic Radar Data: Although our approach achieves effective radar gesture recognition without requiring any labels, it still relies on the collection of real training data. Inspired by video-based radar data generation methods [27], [28], [29], [30], we aim to mitigate the data collection burden through synthetic data in future work. Data generation represents a significant research direction, and current radar data synthesis techniques have demonstrated effectiveness in specific application scenarios. Moreover, video-based data generation methods generally offer greater flexibility and can support finer-grained sensing tasks such as pose estimation. Although existing data generation methods still depend on real training data, data generation and unsupervised learning are not mutually exclusive but rather represent distinct yet complementary research directions. For instance, a generative model trained on real data can be combined with synthesized data to augment unsupervised learning, thereby expanding the dataset while reducing annotation costs. In future work, we will explore fine-grained synthesis of raw radar data to enhance unsupervised training in scenarios with limited real data.

VII. Conclusion

This paper proposes a novel unsupervised mmWave-based gesture recognition method by aligning open images based on heterogeneous transfer learning. We first employ mmWave gesture trajectory to alleviate heterogeneity gap, and then design an ACDTM to achieve fine-grained alignment. The experiment results demonstrate the effectiveness of our proposed approach and underscore the potential of our heterogeneous transfer paradigm as a promising approach in the field of HGR.

References

[1] H. Liu et al., "Real-time arm gesture recognition in smart home scenarios via millimeter wave sensing," in *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, 2020, vol. 4, no. 4, pp. 1–28.
[2] S. An and Ü. Y. Ogras, "MARS: mmWave-based assistive rehabilitation system for smart healthcare," *ACM Trans. Embedded Comput. Syst.*, vol. 20, no. 5, pp. 1–22, 2021.
[3] A. Amir et al., "A low power, fully event-based gesture recognition system," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 7388–7397.
[4] B. K. Chakraborty, D. Sarma, M. K. Bhuyan, and K. F. MacDorman, "Review of constraints on vision-based gesture recognition for human-computer interaction," *IET Comput. Vis.*, vol. 12, no. 1, pp. 3–15, 2018.
[5] B.-B. Zhang, D. Zhang, Y. Li, Y. Hu, and Y. Chen, "Unsupervised domain adaptation for RF-based gesture recognition," *IEEE Internet Things J.*, vol. 10, no. 23, pp. 21026–21038, Dec. 2023.
[6] J. Zhang et al., "A survey of mmWave-based human sensing: Technology, platforms and applications," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 4, pp. 2052–2087, Fourthquarter 2023.
[7] P. S. Santhalingam, A. A. Hosain, D. Zhang, P. Pathak, H. Rangwala, and R. S. Kushalnagar, "mmASL: Environment-independent ASL gesture recognition using 60 Ghz millimeter-wave signals," in *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, 2020, vol. 4, no. 1, pp. 1–30.
[8] Y. Zhang et al., "Widar3.0: Zero-effort cross-domain gesture recognition with Wi-Fi," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8671–8688, Nov. 2022.
[9] Q. Feng et al., "Imbalanced semi-supervised learning for wifi gesture recognition via dynamic threshold-based spatio-temporal attention networks," *IEEE Trans. Mobile Comput.*, early access, Jul. 25, 2025, doi: 10.1109/TMC.2025.3592965.
[10] C. Li, Z. Cao, and Y. Liu, "Deep AI enabled ubiquitous wireless sensing: A survey," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–35, 2022.
[11] S. Palipana, D. Salami, L. A. Leiva, and S. Sigg, "Pantomime: Mid-air gesture recognition with sparse millimeter-wave radar point clouds," in *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, 2021, vol. 5, no. 1, pp. 1–27.
[12] Y. Li et al., "Towards domain-independent and real-time gesture recognition using mmWave signal," *IEEE Trans. Mobile Comput.*, vol. 22, no. 12, pp. 7355–7369, Dec. 2023.
[13] A. Khamis, B. Kusy, C. T. Chou, M.-L. McLaws, and W. Hu, "RfWash: A weakly supervised tracking of hand hygiene technique," in *Proc. 18th Conf. Embedded Networked Sensor Syst.*, 2020, pp. 572–584.

[14] H. Liu et al., "mTransSee: Enabling environment-independent mmWave sensing based gesture recognition via transfer learning," in *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, 2022, vol. 6, no. 1, pp. 1–28.

[15] S. Zhao et al., "A review of single-source deep unsupervised visual domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 473–493, Feb. 2022.

[16] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, vol. 37, pp. 97–105.

[17] Y. Ganin et al., "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, pp. 1–35, 2016.

[18] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1647–1657.

[19] A. Sharma, T. Kalluri, and M. Chandraker, "Instance level affinity-based transfer for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 5361–5371.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[21] "A-z handwritten alphabets," (n.d.). [Online]. Available: https://www.kaggle.com/datasets/sachinpatel21/az-handwritten-alphabets-in-csv-format

[22] G. Cohen, S. Afshar, J. Tapson, and A. V. Schaik, "EMNIST: An extension of MNIST to handwritten letters," in *Proc. Int. Joint Conf. Neural Netw.*, Anchorage, AK, USA, 2017, pp. 2921–2926.

[23] C. Chen et al., "HoMM: Higher-order moment matching for unsupervised domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 3422–3429.

[24] C. Chen et al., "Progressive feature alignment for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recogni.*, Long Beach, CA, USA, 2019, pp. 627–636.

[25] Q. Wang and T. P. Breckon, "Unsupervised domain adaptation via structured prediction based selective Pseudo-labeling," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 6243–6250.

[26] Y. Ge, D. Chen, and H. Li, "Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification," in *Proc. Int. Conf. Learn. Representations*, 2020.

[27] K. Ahuja, Y. Jiang, M. Goel, and C. Harrison, "Vid2Doppler: Synthesizing doppler radar data from videos for training privacy-preserving activity recognition," in *Proc. Conf. Hum. Factors Comput. Syst.*, 2021, pp. 1–10.

[28] K. Deng et al., "Midas: Generating mmwave radar data from videos for training pervasive and privacy-preserving human sensing tasks," in *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, 2023, vol. 7, no. 1, pp. 1–26.

[29] K. Deng, D. Zhao, Z. Zhang, S. Wang, W. Zheng, and H. Ma, "Midas++: Generating training data of mmWave radars from videos for privacy-preserving human sensing with mobility," *IEEE Trans. Mobile Comput.*, vol. 23, no. 6, pp. 6650–6666, Jun. 2024.

[30] J. Li et al., "SBRF: A fine-grained radar signal generator for human sensing," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 13114–13130, Dec. 2024.

[31] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervention*, 2015, vol. 9351, pp. 234–241.

[32] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[33] M. A. Alam, M. M. Rahman, and J. Q. Widberg, "PALMAR: Towards adaptive multi-inhabitant activity recognition in point-cloud technology," in *Proc. INFOCOM-IEEE Conf. Comput. Commun.*, Vancouver, BC, Canada, 2021, pp. 1–10.

[34] P. Zhao et al., "mID: Tracking and identifying people with millimeter wave radar," in *Proc. Int. Conf. Distrib. Comput. Sensor Syst.*, Santorini, Greece, 2019, pp. 33–40.

[35] S. Zhang, T. Zheng, Z. Chen, and J. Luo, "Can we obtain fine-grained heartbeat waveform via contact-free RF-sensing?," in *Proc. INFOCOM-IEEE Conf. Comput. Commun.*, London, U.K., 2022, pp. 1759–1768.

[36] Z. Shi, T. Gu, Y. Zhang, and X. Zhang, "mmBP: Contact-free millimetre-wave radar based approach to blood pressure measurement," in *Proc. 20th ACM Conf. Embedded Networked Sensor Syst.*, 2022, pp. 667–681.

[37] U. Ha, S. Assana, and F. Adib, "Contactless seismocardiography via deep learning radars," in *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw.*, 2020, pp. 1–14.

[38] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[39] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet : Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.

[40] "Texas instruments," (n.d.). [Online]. Available: https://www.ti.com/

[41] A. Farina and F. A. Studer, "A review of CFAR detection techniques in radar systems," *Microw. J.*, vol. 29, 1986, Art. no. 115.

[42] J. Choi, K. Kang, and K. Kim, "Remote respiration monitoring of moving person using radio signals," in *Proc. Eur. Conf. Comput. Vis.*, 2022, vol. 13697, pp. 253–270.

[43] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Automat.*, Shanghai, China, 2011, pp. 1–4.

[44] G. M. Rao and C. Satyanarayana, "Object tracking system using approximate median filter, Kalman filter and dynamic template matching," *Int. J. Intell. Syst. Appl.*, vol. 6, no. 5, 2014, Art. no. 83.

[45] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Anal. Chem.*, vol. 36, no. 8, pp. 1627–1639, 1964.

[46] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[47] S. J. Delany, P. Cunningham, D. Doyle, and A. Zamolotskikh, "Generating estimates of classification confidence for a case-based spam filter," in *Proc. Int. Conf. Case-Based Reasoning*, 2005, vol. 3620, pp. 177–190.

[48] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *IEEE/CVF Conf. Comput. Vis. Pattern Recogni.*, Seattle, WA, USA, 2020, pp. 9726–9735.

[49] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, vol. 119, pp. 1597–1607.

[50] G. Wilson and D. J. Cook, "A survey of unsupervised deep domain adaptation," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 5, pp. 1–46, 2020.

[51] D.-H. Lee et al., "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Workshop Challenges Representation Learn.*, 2013, vol. 3, pp. 896–901.

[52] L. Shi and W. Liu, "Adversarial self-training improves robustness and generalization for gradual domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 1–13.

[53] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Representations*, 2015.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[55] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, vol. 37, pp. 448–456.

[56] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[57] S. Narayan, "The generalized sigmoid activation function: Competetive supervised learning," *Inf. Sci.*, vol. 99, no. 1/2, pp. 69–82, 1997.

[58] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.

[59] T. Wolf et al., "Transformers: State-of-the-art natural language processing," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*, 2020, pp. 38–45.

[60] Q. Feng, Z. Lu, C. Li, F. Huang, J. Weng, and P. S. Yu, "End-to-end privacy-preserving image retrieval in cloud computing via anti-perturbation attentive token-aware vision transformer," *Inf. Fusion*, vol. 121, 2025, Art. no. 103153.

[61] Q. Feng et al., "Evit: Privacy-preserving image retrieval via encrypted vision transformer in cloud computing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 8, pp. 7467–7483, Aug. 2024.

[62] Q. Feng et al., "Privacy-preserving image retrieval in cloud computing via adaptive secret keys and self-supervised block-augmented pretraining," *IEEE Trans. Serv. Comput.*, vol. 18, no. 4, pp. 2310–2325, Jul./Aug. 2025.

[63] L. Chen et al., "SALIENCE: An unsupervised user adaptation model for multiple wearable sensors based human activity recognition," *IEEE Trans. Mobile Comput.*, vol. 22, no. 9, pp. 5492–5503, Sep. 2023.

[64] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 3723–3732.

[65] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, vol. 70, pp. 2208–2217.
[66] N. Xiao and L. Zhang, "Dynamic weighted learning for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Nashville, TN, USA, 2021, pp. 15237–15246.
[67] B. Xie, S. Li, F. Lv, C. H. Liu, G. Wang, and D. Wu, "A collaborative alignment framework of transferable knowledge extraction for unsupervised domain adaptation," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 7, pp. 6518–6533, Jul. 2023.
[68] S. D. Regani, C. Wu, B. Wang, M. Wu, and K. J. R. Liu, "mmWrite: Passive handwriting tracking using a single millimeter-wave radio," *IEEE Internet Things J.*, vol. 8, no. 17, pp. 13291–13305, Sep. 2021.
[69] T. Wei and X. Zhang, "mTrack: High-precision passive tracking using millimeter wave radios," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 117–129.
[70] S. Zhao, B. Li, C. Reed, P. Xu, and K. Keutzer, "Multi-source domain adaptation in the deep learning era: A systematic survey," *CoRR*, vol. abs/2002.12169, 2020.
[71] D. Wu et al., "Fingerdraw: Sub-wavelength level finger motion tracking with wifi signals," in *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, 2020, vol. 4, no. 1, pp. 1–27.
[72] J. Zhang, Y. Li, H. Xiong, D. Dou, C. Miao, and D. Zhang, "Handgest: Hierarchical sensing for robust-in-the-air handwriting recognition with commodity wifi devices," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 19529–19544, Oct. 2022.
[73] S. K. Leem, F. Khan, and S. H. Cho, "Detecting mid-air gestures for digit writing with radio sensors and a CNN," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1066–1081, Apr. 2020.
[74] L. Zhao, R. Xiao, J. Liu, and J. Han, "One is enough: Enabling one-shot device-free gesture recognition with COTS wifi," in *Proc. INFOCOM-IEEE Conf. Comput. Commun.*, Vancouver, BC, Canada, 2024, pp. 1231–1240.

Qihua Feng received the MS degree in computer technology from Jinan University, Guangzhou, China, in 2022. He is currently working toward the PhD degree with the Beijing Institute of Technology, Beijing, China. His research interests include privacy preserving, deep learning, LLM, and Internet of Things.

Kunpeng Cheng received the BS degree from the Beijing Institute of Technology, Beijing, where he is currently working toward the MS degree. His research focuses on Internet of Things.

Chunhui Duan (Member, IEEE) received the BS and PhD degrees from the School of Software, Tsinghua University, Beijing, China, in 2013 and 2018, respectively. She was a postdoctoral research fellow with Tsinghua University. She is currently an associate professor with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing. Her research interests include RFID, Internet of Things, wireless sensing, and mobile computing.