

# Dynamic Associate Domain Adaptation for Human Activity Recognition Using WiFi Signals

Yuh-Shyan Chen, Chun-Yu Li, and Tong-Ying Juang

Department of Computer Science and Information Engineering, National Taipei University.

E-mail: yschen@mail.ntpu.edu.tw

**Abstract**—In this paper, a semi-supervised transfer learning with dynamic associate domain adaptation is proposed for human activity recognition by using the channel state information (CSI) of the WiFi signal. We propose a dynamic associate domain adaptation (DADA), by modifying the existing associate domain adaptation algorithm, while the target domain can dynamically provide a different ratio of labelled data set/unlabelled data set. The advantage of DADA is that it provides a dynamic strategy to eliminate different effects under the different environments. We designed an attention-based DenseNet model (AD) as our training network, so our proposed scheme is simplified as DADA-AD scheme. The experimental results illustrate that the accuracy of human activity recognition of the DADA-AD scheme is 97.4%. It also shows that DADA-AD has advantages over existing semi-supervised learning schemes.

**Index Terms**—Human activity recognition, channel state information, semi-supervised learning, domain adaptation, attention.

## I. INTRODUCTION

Environmental sensors are widely deployed everywhere in our daily environmental. With the environmental sensor data, it records our daily activities through human activity recognition (HAR). The research significance and practical value of HAR has attracted recently, so a large number of research results on HAR have been attention recently. The existing HAR systems usually use cameras, wearable devices, and sensors [1]. However, all of the aforementioned methods require a large amount of hardware equipment with the limitation of the power lifetime to limit its universality. Consequently, it is very important and valuable to investigate the device-free HAR system.

Therefore, WiFi-based HAR had made great progress, many efforts are dedicated to develop practical applications. For example, positioning had been carried out by using Received Signal Strength Indication (RSSI) and Channel State Information (CSI) [2], CSI illustrates the overall amplitude response, thereby more finely depicting the state of the channel. But there are still many challenges for using CSI. For instance, due to the different superposition of multipath, the received signal of the same activity and the influence on the wireless channel are significantly different at different locations. The contributions of this paper are as follows:

- We design a new semi-supervised transfer learning with dynamic associate domain adaptation (DADA) capability for HAR. Our proposed DADA scheme can dynamically

adjust ratio of labeled data set and unlabeled data set of the target domain, which is dynamically depended on the target environment status.

- Our actual experimental results show that if the data is unbalanced, the average accuracy of DADA is 4.17% higher than that of ADA, but if there is no data unbalance is only 1.08%.
- To increase the recognition accuracy, an attention-based densenet model (AD) is designed as our new training network. Our experimental results show that the accuracy of AD as our training network is increased by 4.13%, compared to existing HAR-MN-EF scheme [3].

The rest of the paper is organized as follows. Section II describes the related work. Section III describes the problem formulation. Section IV describes the proposed scheme. Section V discusses the performance results. Finally, Section VI concludes this paper.

## II. RELATED WORK

When the environment changes, the background noise of the environment will change the characteristics, resulting in poor recognition efficiency. Shi et al. [3] proposed an environment-robust channel state information (CSI) based HAR by leveraging the properties of a matching network (MatNet) and enhanced features. MatNet allows to learn and extract inherent and transferable functions, thereby transferring knowledge in different environments. Unfortunately, although the knowledge of CSI information after feature extraction can be transferred, but the required accuracy cannot be met only by directly transferring the features. Ding et al. [4] proposed a semi-supervised WiFi location-independent HAR, called WiLISensing. Han et al. [5] proposed DANGR. The key idea is to adopt the domain adaptation based on the multi-core maximum mean difference scheme.

## III. PROBLEM FORMULATION

In this work,  $D_s$  and  $D_t$  represent as the source domain and the target domain,  $\tilde{H}^s$  and  $\tilde{H}^t$  are further denoted as the CSI matrix from the source domain and the target domain, respectively. We consider a source domain,  $D_s = \{\tilde{H}_i^s, y_i^s\}_{(i=1, \dots, n_s)}$ , where  $\tilde{H}_i^s$  is the  $i$ -th collected CSI matrix  $\tilde{H}_i^s$  from the source environment, and  $y_i^s$  is the corresponding label of  $\tilde{H}_i^s$ . The target domain,  $D_t = \{\tilde{H}_i^t, y_i^t\}_{(i=1, \dots, n)} \cup \{\tilde{H}_i^t\}_{(i=n+1, \dots, n_t)}$ ,  $n^s$  and  $n^t$  represent

the total number of  $D_s$  and  $D_t$  data respectively, where  $\tilde{H}_i^t$  is the  $i$ -th collected CSI matrix  $\tilde{H}^t$  from the target environment. It is observed that, CSI matrix  $\tilde{H}_i^t$  of the target environment has target label  $y_i^t$ , where  $1 \leq i \leq n$ . But, there are no target labels for all  $\tilde{H}_i^t$ , where  $n+1 \leq i \leq n_t$ . That is, the target labels  $\{y_i^t\}_{(i=n+1, \dots, n_t)}$  are not available for training.

We formalized the objective function  $\mathcal{L}_f$  as follows:

$$\begin{aligned} & \arg \min^* \mathcal{L}_f \\ & \mathcal{L}_f = (1 - \lambda)\mathcal{L}_c + \lambda \cdot \mathcal{L}_{sim} \quad (1) \\ & \text{subject to } 0 \leq \lambda \leq 1 \end{aligned}$$

where  $\mathcal{L}_f$  is the combined objective function of both considering the  $\mathcal{L}_c$  and  $\mathcal{L}_{sim}$ , where  $\mathcal{L}_c$  is the objective function of classification,  $\mathcal{L}_{sim}$  is the objective function of similarity problem,  $\mathcal{L}_f$  is the weigh-sum of  $\mathcal{L}_c$  and  $\mathcal{L}_{sim}$ , as the total objective function, where  $\lambda$  is the hyper-parameter of the hybrid objective function. The objective function  $\mathcal{L}_{sim}$  is used to measure the difference between two different distributions, which is expressed as wasserstein distance.

$$\mathcal{L}_{sim} = \sqrt{\min_{P_{st}} E_{P_{st}} \|\phi(D_s) - \phi(D_t)\|_2^2}. \quad (2)$$

The distribution  $P_s$  represents the distribution of  $D_s$ , the distribution  $P_t$  represents the distribution of  $D_t$ , and  $P_{st}$  is the joint distribution of  $P_t$  and  $P_s$ , and  $P_s \neq P_t$ ,  $\phi$  represents as a mapping function, which maps data of different distributions to the same space.  $\mathcal{L}_{sim}$  represents is the joint distribution  $P_{st}$ , find out the minimum expected value  $E_{P_{st}}$  by mapping  $D_s$  and  $D_t$  in the same space through  $\phi$ .

$\mathcal{L}_c$  is the objective function used for classification problems, is given as follows.

$$\mathcal{L}_c = \max \left[ \frac{1}{N_s} \sum_{i=1}^{N_s} H(y_i^s, p_i^s), \frac{1}{N_t} \sum_{i=1}^{N_t} H(y_i^t, p_i^t) \right], \quad (3)$$

$$H(y_i^s, p_i^s) = -\frac{1}{m} \sum_{j=1}^m y_{i,j}^s \cdot \log(p_{i,j}^s), \quad (4)$$

$$H(y_i^t, p_i^t) = -\frac{1}{m} \sum_{j=1}^m y_{i,j}^t \cdot \log(p_{i,j}^t), \quad (5)$$

$$\text{subject to } \begin{cases} y_{i,j}^s, y_{i,j}^t, & \forall i, j \\ 0 \leq p_{i,j}^s, p_{i,j}^t \leq 1, & \forall i, j \end{cases}$$

where  $Q(y_i^s, p_i^s)$  and  $Q(y_i^t, p_i^t)$  are cross-entropy function of source domain and target domain for classification, where  $Q(y_i^s, p_i^s)$  is the cross-entropy used for classification problem in source domain,  $y_i^s$  is denoted as the  $i$ -th data belongs to the real category in source domain, and  $p_i^s$  is denoted as the predicted probability in source domain,  $Q(y_i^t, p_i^t)$  is the cross-entropy used for classification problem in target domain,  $y_i^t$  is denoted as the  $i$ -th data belongs to the real category in target domain, and  $p_i^t$  is denoted as the predicted probability in target domain, and  $n_s$  and  $n_t$  represent the number of training data of the source and target domains.  $m$  is defined as the number of classification categories,  $y_{i,j}^s$  and  $y_{i,j}^t$  are respectively denoted as the  $i$ -th data in  $D_s$  and  $D_t$ , the data belongs to the real

category of the  $j$ -th category.  $p_{i,j}^s$  and  $p_{i,j}^t$  are expressed as the predicted probabilities that belongs to the predict category of the  $j$ -th category of  $D_s$  and  $D_t$  respectively. The purpose of  $\mathcal{L}_c$  is to minimize the classification error.

#### IV. A SEMI-SUPERVISED TRANSFER LEARNING WITH DYNAMIC ASSOCIATE DOMAIN ADAPTATION FOR HAR

In this section, we propose a HAR algorithm based on semi-supervised dynamic associate domain adaptation learning in WiFi networks to predict unlabeled activity recognition with the cross-domain data. The algorithm is divided into four phases. The system structure of DADA-AD scheme is shown in Fig. 1.

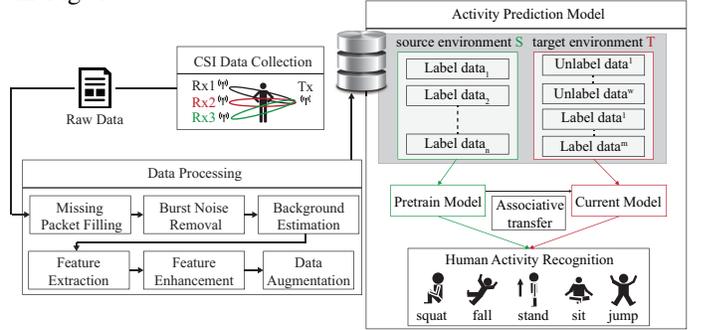


Fig. 1. System structure of DADA-AD scheme.

##### A. Data collection and processing phase

CSI matrix  $H$  is performed six steps, including missing packet filling, burst noise removal, background estimation, feature extraction, feature enhancement, and data augmentation operations. In this work, we use one transmitting antenna ( $N_r = 1$ ) and three receiving antennas ( $N_s = 3$ ) for five kinds of activity recognition; standing, sitting, squatting, jumping, and falling. We provide a 2D diagram of the each data matrix in Fig. 2.

**Missing packet filling:** The linear interpolation is used to repair the lost packets. The lost packet  $h_{j,k}(i)$  can be repaired by a simple linear interpolation function as:

$$h_{j,k}(i) = (i - p) \frac{h_{j,k}(n) - h_{j,k}(p)}{n - p} + h_{j,k}(p) \quad (6)$$

where  $h_{j,k}(p)$  and  $h_{j,k}(n)$  are represented as the previous packet and the next packet of  $h_{j,k}(i)$ , respectively. The output matrix  $H^{pf} = [h^{pf}(1), \dots, h^{pf}(i), \dots, h^{pf}(K)]$ , where  $1 \leq i \leq K$ , is obtained, where  $H^{pf} = \text{linear\_interpolation}(H)$ .

**Burst noise removal:** We adopt the Wavelet transform denoising algorithm to  $H^{pf}$  matrix to obtain  $H^{nr}$  matrix as follow.

$$\begin{aligned} H^{nr} &= DWT(o, p, H^{pf}) \\ &= \int_{-\infty}^{\infty} 2^{-\frac{\sigma}{2}} \psi(2^{-\sigma}i - p) h^{pf}(i) di \end{aligned} \quad (7)$$

A 6-level discrete wavelet transform is used to decompose, and Symlet is used as the wavelet base, and the denoised CSI packet sequence will be reconstructed through inverse transform.

**Background estimation:** We let  $h^{nr}(i)$  be represented as  $h^{nr}(i) = h^{be}(i) + h^{fe}(i)$ , for  $1 \leq i \leq K$ . The main work is to estimate the dynamic CSI vector  $h^{fe}(i)$ , be generated by the human activities, so  $h^{be}(i)$  is initially obtained, for

$1 \leq i \leq K$ , by adopting the exponentially weighted moving average (EWMA) algorithm [6], as follows.

$$h^{be}(i) = \lambda h^{nr} + (\lambda - 1)h^{be}(i - 1) \quad (8)$$

where  $1 \leq i \leq K$ ,  $\lambda$  is the forgetting factor, where  $0 \leq \lambda \leq 1$ . Each new estimated point is recursively calculated from the previous observations and attenuated by a forgetting factor. If static CSI matrix  $H^{be}$ , is finding, so the dynamic CSI matrix  $H^{fe}$  is obtained by  $H^{fe} = H^{nr} - H^{be}$ .

**Feature enhancement:** We adopt the similar feature enhancement algorithm [3] to obtain the correlation matrix  $\tilde{H}$ , where  $\tilde{H} = H^{fe} \times H^{feT}$ . The correlation matrix between the signals on all subcarriers  $M$  eliminates the time dimension information, leaving the characteristics of the correlation between the subcarriers.

**Data augmentation:** To enhance the robustness of model training, the data augmentation technique is used to enlarge the training data set to generate more training data. In this work, the correlation matrix  $\tilde{H}$  will be augmented by adopting the spin, mask, and zoom methods.

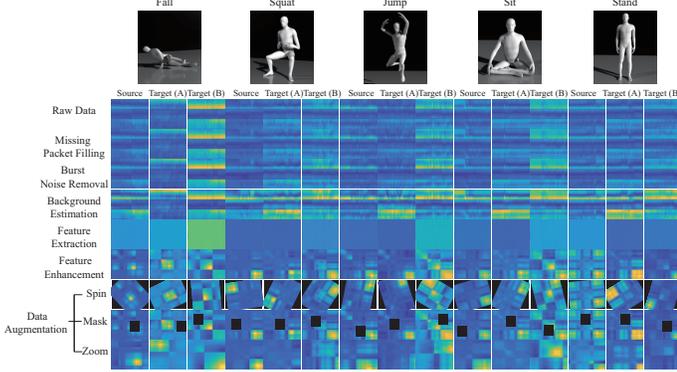


Fig. 2. The 2-D diagrams of CSI data pre-processing.

### B. Pre-training phase

The basic training network of this work is adopted a deep DenseNet model [7]. The structure of AD shows in Fig. 3. In denseblock, the output of all previous layers is connected as input,  $z_o^D$ , for the next layer, be expressed as:

$$z_o^D = \sigma_D(D_0, \dots, D_{L-1}) \quad (9)$$

where,  $\sigma_D$  represents a non-linear transformation function, and  $D_{L-1}$  represents the output of the  $L - 1$  layer in the denseblock. Assuming that each layer in the denseblock uses  $k$  convolution kernels. Let the channel number of the feature map in the input layer be  $c_0$ , and the last output channel number is

$$C^D = c_0 + k^D(L - 1) \quad (10)$$

where, denseblock utilizes the bottleneck architecture to reduce the calculation cost. We expect to adding the channel attention mechanism to strengthen the correlation between feature channels to improve training accuracy. Suppose there are  $C^D$  input channels, the number of output channels,  $C^T$ , of a denseblock is expressed as:

$$C^T = \theta C^D \quad (11)$$

where  $0 < \theta \leq 1$ ,  $\theta$  is the compression factor. The output feature,  $z_o^T$ , by the connection layer is expressed as:

$$z_o^T = \sigma_T(z_o^D) \quad (12)$$

where  $\sigma_T$  represents the non-linear transformation, which is repeatedly used in the transition layer [7], by adding the ECA network as a substructure, it is embedded in the connection layer to learn feature weights to achieve better training results.

The relationship can be expressed as  $C^T = \psi(cs)$ , where  $\psi$  is the approximate exponential mapping function, is expressed as  $\psi(cs) = 2^{(\gamma * cs - \omega)}$ , given the channel dimension  $C^T$ , the adaptation channel size  $cs$  [8], i.e. the number of neighbouring channels, is expressed as:

$$cs = \left\lfloor \frac{\log_2(C^T) + \omega}{\gamma} \right\rfloor_{odd} \quad (13)$$

where  $\omega$  and  $\gamma$  is set as 1 and 2. With ECAT, the channel and feature size is adjusted and the channel dimension is also reduced, and the important channel weight can be increased through channel attention mechanism. The weight is expressed as:

$$w_o = \sigma_{ECA}^{cs}(z_o^T) \quad (14)$$

$\sigma_{ECA}^{cs}$  is an adaptation non-linear transformation which is composed of global average pooling and  $1 \times 1 \times cs$  convolution. Consequently, the output weighted feature  $z_o^{ECAT}$  is,

$$z_o^{ECAT} = z_o^T * w_o \quad (15)$$

by multiplying weight  $w_o$  and output  $z_o^T$  of the connection layer. After repeating the denseblock structures with ECAT mechanism twice, the maxpooling operation  $MP$  is applied to the feature to extract the maximum value.

$$m_o = MP(z_o^p) \quad (16)$$

where  $z_o^p$  is the final output before reaching to the flattening layer, where  $m_o$  is the flattened feature. Finally,  $m_o$  pass through a  $f$ -layer fully connected layer to obtain the final feature, denoted as  $d$  and used for the final activity prediction through a fully connected layer by an activation function of softmax.

$$p_i^s = W_o \times d + b_o \quad (17)$$

where  $p_i^s$  represents the predicted value of  $\tilde{H}_i^s$ , where  $W_o$  and  $b_o$  are trainable parameters. After obtaining the final activity prediction, the loss function is calculated through cross entropy, that we have defined in formula 4.

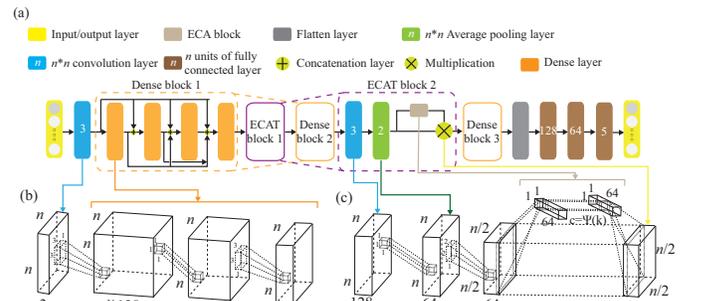


Fig. 3. (a) The attention-based DenseNet (AD) model. (b) Illustration of Denseblock with a  $1 \times 1$  convolution layer and a  $3 \times 3$  convolution layer. (c) Illustration of transition block with a  $1 \times 1$  convolution layer, a  $2 \times 2$  pooling layer, and an ECA structure.

### C. Dynamic associate domain adaptation phase

The source domain and target domain are mapped into the same feature space. Let  $S_i = \phi\{\tilde{H}_i^s\}_{i=1,\dots,n_s}$ ,  $U_j = \phi\{\tilde{H}_j^t\}_{i=1,\dots,n}$ ,  $L_k = \phi\{\tilde{H}_k^t\}_{i=n+1,\dots,n_t}$ , then dot product is used to calculate the similarity of the source domain and the target domain. The similarity of domain features is calculated by a similarity matrix between the source domain and the target domain,  $F_{ij} = S_i \cdot U_j$  and  $G_{ik} = S_i \cdot L_k$ , where  $F_{ij}$  is the similarity matrix of the unlabeled data of the source domain and the target domain, and  $G_{ik}$  is the similarity matrix of the labeled data of the source domain and the target domain.

After obtaining the similarity matrix,  $F_{ij}$ , a conversion probability  $P_{ij}^{SU}$  followed by [9] is

$$\begin{aligned} P_{ij}^{SU} &= P(U_j|S_i) \\ &= SM_{columns}(F)_{ij} = \frac{\exp(F_{ij})}{\sum_{j'} \exp(F_{ij'})} \end{aligned} \quad (18)$$

where  $P_{ij}^{SU}$  is the conversion probability of the similarity matrix  $F_{ij}$  by applying softmax function for the column of  $F_{ij}$ . To more consideration of DADA, we further calculate  $P_{ik}^{SL}$  as follows. Similarly, the conversion probability from [9] of the target domain of  $F_{ij}$  is,

$$\begin{aligned} P_{ij}^{US} &= P(S_i|U_j) \\ &= SM_{rows}(F)_{ij} = \frac{\exp(F_{ij})}{\sum_{i'} \exp(F_{i'j})} \end{aligned} \quad (19)$$

where  $P_{ij}^{US}$  is the conversion probability of the similarity matrix  $F_{ij}$  by applied softmax function to the row of  $F_{ij}$ . To more consideration of DADA, we also further calculate  $P_{ik}^{LS}$  as follows. Following [9], the subsequent calculation of the associated similarity for unlabeled data in the target domain can be expressed as  $P^{SUS} = (P^{SU} P^{US})_{ij} = \sum_n P_{in}^{SU} P_{ni}^{US}$ , where  $P_{ij}^{SUS}$  [9] is the round-trip probability of similarity matrix  $F_{ij}$ , starting from  $S_i$  and ending at  $S_j$ . Assuming that the label mapped back to  $S_j$  is unchanged relative to  $S_i$ , the label distribution of  $S_i$  [9] is expressed as  $Y_{ij} = \begin{cases} 1/S_i & \text{class}(S_i) = \text{class}(S_j) \\ 0 & \text{else} \end{cases}$ .

The cross-entropy with the round-trip probability can be expressed as  $\mathcal{L}_{SUS} = Q(Y_{ij}, P^{SUS})$ , but this round-trip mapping cannot directly reflect the difference degree, so we further modify the ADA by dynamically utilizing a different ratio of labeled data of the target domain to map back to the source domain to obtain the difference degree. Since both parties have labels, the new defined cross-entropy calculation,  $\mathcal{L}_{LS}$ , is performed through the conversion probability of  $P_{ik}^{LS}$  and the distribution probability  $J_{ij}$  of the label data of the target domain mapped to the source domain,

$$\mathcal{L}_{LS} = Q(J_{ij}, P_{ik}^{LS}) \quad (20)$$

Assuming that the label mapped to  $S_i$  relative to  $L_k$  is unchanged, the label distribution of  $Y_{ij}$  can be expressed as  $J_{ik} = \begin{cases} \frac{1}{L_k} & \text{class}(L_k) = \text{class}(S_i) \\ 0 & \text{else} \end{cases}$ . The divergence between the two domains is,

$$\mathcal{L}_{div} = \max[Q(Y_{ij}, P^{SUS}), Q(J_{ij}, P^{LS})] \quad (21)$$

where  $\mathcal{L}_{div}$  is the loss of the divergence of the two domains. Two different mapping functions is referenced to illustrate the distance degrees.

It is unreasonable that the calculation of  $\mathcal{L}_{vis}$  under the data distribution must be balanced [9]. This is because that the number of unlabeled data is unknown before training. To provide the data imbalance capability and release the limitation of  $\mathcal{L}_{vis}$ , our DADA scheme is replaced traditional  $\mathcal{L}_{vis}$  [9] with a new loss function of synchronization, denoted  $\mathcal{L}_{syn}$ , as follows.

$$\begin{aligned} \mathcal{L}_{syn} &= Q(P_j^{US}, P_j^{SU}) \\ \text{subject to } &\begin{cases} P_j^{SU} = \sum_{j'} P_{ij'}^{SU} \\ P_j^{US} = \sum_{j'} SM_{columns}(P_{ij'}^{SU T}) \end{cases} \end{aligned} \quad (22)$$

where  $P_j^{SU}$  adds up the columns in line units and  $P_j^{US T}$  adds up the columns in line units to make sure that both  $P_j^{SU}$  and  $P_j^{US T}$  are still kept in same distribution. This work is measured the correlation which can avoid only correlating the simple and easily correlated data in the unlabeled target domain, under the data distribution of  $U_j$  is imbalance. Finally,  $\mathcal{L}_{sim}(D_s, D_t)$  is obtained by

$$\mathcal{L}_{sim}(D_s, D_t) = \beta \mathcal{L}_{dis} + (\beta - 1) \mathcal{L}_{syn} \quad (23)$$

where  $\beta$  is the hyperparameter of the combined targets.  $\mathcal{L}_{sim}$  is represented as the combined loss,  $\mathcal{L}_{div}$  and new constructed  $\mathcal{L}_{syn}$ .

### D. Associate knowledge fine-tuning phase

In the last phase, the learned features are transferred through the HAR of the image with domain-invariant characteristics, and the shallow weights of the source domain learned through pre-training phase are frozen as common features, and knowledge transfer is performed on the deep layer of the model.

The maximum pooling operation  $MP$  is applied to the feature to extract the maximum near-row flattening. This operation is expressed as formula 16, where  $m_o$  is the set of data features from source domain  $D_s$  and domain domain  $D_t$ , and is used as flattened feature, and  $m_o$  passes through the  $k$ -layer fully connected layer to calculate the similarity of the  $k$ -layer feature  $\mathcal{L}_{sim}$ , as given.

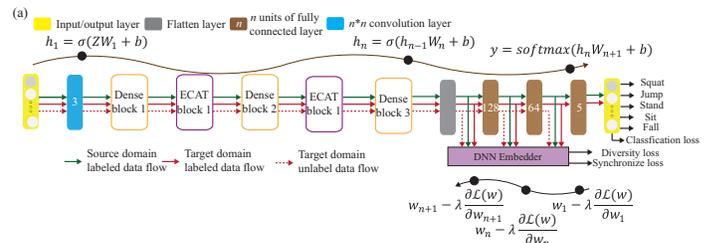


Fig. 4. Example of associate knowledge fine-tuning phase.

$$\sum_{f_k}^{l=f_1} \mathcal{L}_{sim}(D_s^l, D_t^l) \quad (24)$$

where  $l$  is the current number of layers. The similarity values of the  $k$ -layer features are accumulated as part of the loss.

The final feature is obtained by  $d = d^s \cup d^t$ , which is used for the final activity prediction through the fully connected layer with the activation function of softmax, where  $p_i^s = W_o \times d^s + b_o$ ,  $p_i^t = W_o \times d^t + b_o$ , where  $W_o$  and  $b_o$  are trainable parameters, and  $p_i^s$  is the predicted value of  $\tilde{H}_i^s$ , and  $p_i^t$  is the predicted value of  $\tilde{H}_i^t$ . Finally, the total loss can be represented as

$$\arg \min^* \mathcal{L}_f$$

$$\mathcal{L}_f = \lambda \mathcal{L}_c + (1 - \lambda) \sum_{l=f_1}^{f_k} \mathcal{L}_{sim}(D_s^l, D_t^l) \quad (25)$$

Let  $\sum_{l=f_1}^{f_k} \mathcal{L}_{sim}(D_s^l, D_t^l)$  be similarity between two domains of the fully connected layer. The final goal is to minimize  $\mathcal{L}_f$ , which  $\mathcal{L}_f$  is the combined function of  $\mathcal{L}_{sim}$  and  $\mathcal{L}_c$ , where  $\lambda$  is the hyper-parameter of the hybrid objective function, which  $0 \leq \lambda \leq 1$ .

As shown in Fig. 4, the weighs before flatten layer are frozen and the similarity loss with all data of both domains are calculated. The backpropagation operation is done to update the weights of the fully connected layer. For instance as shown in Fig. 4,  $\sum_{l=f_1}^{f_3} \mathcal{L}_{sim}(D_s^l, D_t^l)$  is similarity loss under  $\max \left[ \frac{1}{n_s} \sum_{i=1}^{n_s} Q(y_i^s, p_i^s), \frac{1}{n_t} \sum_{i=1}^{n_t} Q(y_i^t, p_i^t) \right]$ , where  $f_3$  denotes an example of 3 fully connected layer in our work.

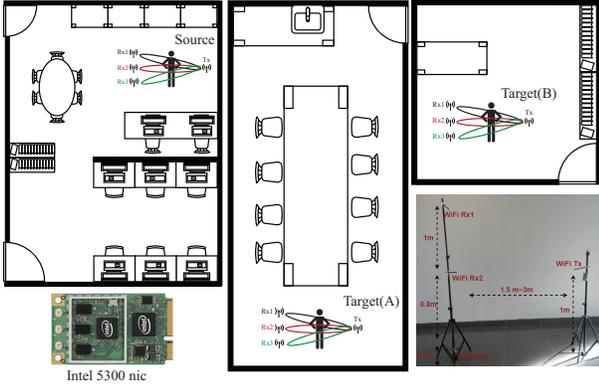


Fig. 5. Layout of three experimental areas (a) source domain with the size of is  $5m \times 4m$ , (b) target domain A with sizes of  $10m \times 4m$ , (c) target domain B with sizes of  $3m \times 4m$ .

## V. EXPERIMENTAL RESULT

The environment setup is described, mainly including the model parameter settings for our experimental results. In our experimental, the AI GPU utilizes NVIDIA RTX 3080, and the AI framework adopts pytorch version 1.4.0. The programming environment is the Python 3.8 version under Windows 10. The WiFi CSI data acquisition framework uses the Intel IWL 5300 NIC tool [10], two computers installed Ubuntu 14.04.4 equipped with the Intel 5300 NIC are used as the interface programming environment. The recognition activity patterns in our experimental are: jumping, squatting, sitting, standing and falling. The experimental parameters are given in in Table I. In the source domain as shown in Fig. 5. In the experimental, we use source, target(A), and target(B) to represent the source

domain, target domain A, and target domain B. The data augmented data set is 1.5 times than that of the original training set by adopting the rotate, map, and mask techniques. To illustrate the effect of environment-independent human activity recognition, the source domain is used to train the pre-trained model, and the pre-trained model is used to transfer trained knowledge to target(A) and target(B).

TABLE I  
EXPERIMENT PARAMETER.

Environment	Source	Target(A)	Target(B)
Sampling frequency	1000 Hz		
Transmit antenna	1 antenna		
Receiving antenna	3 antenna		
Sampling time	3 seconds		
Subcarriers per link	30 subcarriers		
Source data set	4000	1500	
Target data set	1500	500	
Data expansion factor	1.5		
Weight adjustment $\lambda$	0.5		
Compression factor $\theta$	0.5		
Weight adjustment $\beta$	0.5		
Learning rate	0.001		
Drop out	0.5		

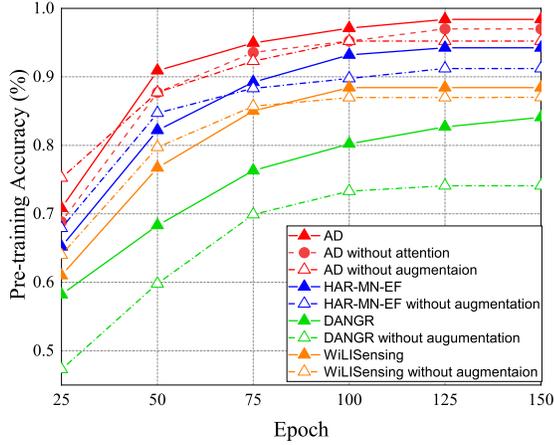
1) *Pre-training accuracy (PTA)*: The experimental results of pre-training accuracy (PTA) vs. epochs are shown in Fig. 6. The PTA is the ratio of the number of correct classification prediction to the total number of predictions. Fig. 6(a) illustrates that the PTA of AD is better than that of other schemes due to the advantage of the feature attention and reuse strategy. DANGR scheme does not use the feature enhancement, which leads to the poor learning efficiency. All other schemes adopt the feature enhancement, but leads to the huge differences in PTA. In general, feature enhancement is helpful for the pre-training. Experimental result also shows that PTA of pre-training schemes with data augmentation and attention mechanisms is higher than that of pre-training schemes without data augmentation and attention mechanism. The PTA of pre-training with the different number of antennas vs. epoch is illustrated in Fig. 6(b). The number of antennas affects PTA. With the same number of antennas, the PTA of our AD scheme is better that that of other schemes.

2) *Recognition accuracy (RA)*: Table II provides shows that RA vs. various ratios of labeled data for target(A) and target(B) under target data ratio = (1:1:1:1:1). We also observed that the average RA of our AD-DADA scheme > that of

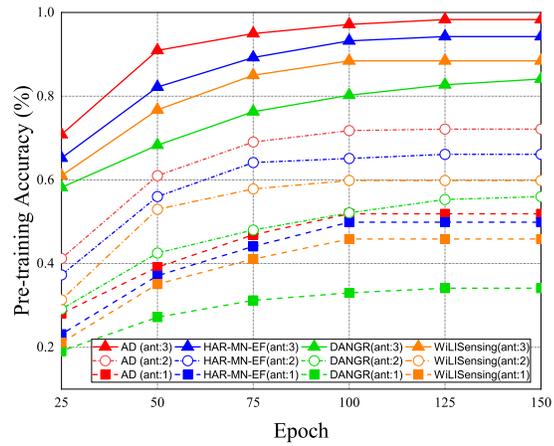
TABLE II

RECOGNITION ACCURACY (RA) OF ALL CASES FOR ENCOUNTERING FOR TARGET(A) AND TARGET(B).

Model	Target data ratio		1:1:1:1:1					1:1:0.5:1:0.5					1:1:1:1:0.1				
	Domain	Scheme	0%	25%	50%	75%	100%	0%	25%	50%	75%	100%	0%	25%	50%	75%	100%
AD	Target(A)	DADA	94.4%	96.7%	96.8%	96.9%	97.1%	93.9%	96.0%	96.1%	96.2%	96.4%	89.5%	91.2%	91.7%	92.8%	93.1%
		ADA	93.5%	94.9%	95.5%	95.4%	95.2%	88.1%	89.7%	91.7%	93.3%	94.2%	87.2%	89.7%	89.2%	88.5%	88.3%
		MK-MMD	91.9%	92.3%	93.8%	95.9%	96.9%	91.1%	91.7%	93.8%	95.1%	96.3%	87.0%	88.9%	90.5%	91.8%	93.1%
		MMD	83.2%	83.9%	85.1%	89.3%	97.0%	83.7%	85.1%	87.7%	91.3%	96.2%	84.9%	85.7%	88.2%	90.7%	93.0%
	Target(B)	DADA	95.1%	97.4%	97.6%	97.8%	98.1%	94.7%	96.8%	97.6%	97.8%	97.9%	92.1%	93.7%	94.0%	94.7%	95.4%
		ADA	94.9%	95.6%	96.3%	95.9%	95.5%	89.3%	90.9%	93.3%	95.1%	96.1%	89.3%	90.5%	93.7%	93.4%	92.9%
		MK-MMD	93.3%	93.9%	94.4%	96.1%	97.5%	92.8%	93.1%	94.4%	96.1%	97.5%	89.4%	91.5%	92.1%	93.6%	95.4%
		MMD	87.6%	88.7%	91.0%	94.8%	97.4%	86.6%	87.6%	91.0%	94.8%	97.1%	85.1%	86.6%	89.1%	92.2%	95.4%



(a)



(b)

Fig. 6. PA vs. epoch for (a) data argumentation in different methods. (b) different number of antennas in different methods.

AD-ADA scheme > that of AD-MK-MMD scheme > that of AD-MMD scheme > that of DANDR scheme > that of HAR-MN-EF scheme from the perspective of ratios of labeled data. The improvement of RA for target(B) is better than that of target(A). This is because that target(B) is a small area. The experimental result of RA vs. various ratios of labeled data for target(A) and (b) target(B) under target data ratio = (1:1:0.5:1:0.5). We observed that if ratio of labeled data = 25%, the RA can be improved, especially for the data imbalance problem is occurred. In general, our proposed AD-DADA can provide a general adjustment scheme to dynamically increase the ratio of labeled data if a user encountering a new poor target environment.

## VI. CONCLUSIONS

In this paper, a semi-supervised transfer learning with dynamic associate domain adaptation is proposed for human activity recognition. DADA can provide a dynamic ratio of labeled data set/unlabeled data set. Finally, the experimental results shows that if the data is unbalanced, the average accuracy of DADA is 4.17% higher than that of ADA, but if there is no data unbalance, it is only 1.08%. In future work, we may extend this research result to mmWave sensor network for human activity recognition with fine motion.

## ACKNOWLEDGEMENTS

This research was supported by the MOST 109-2221-E-305-004-MY3 and MOST 110-2221-E-305-001.

## REFERENCES

- [1] A. Gupta, K. Gupta, K. Gupta, and K. Gupta. "A Survey on Human Activity Recognition and Classification". In *Proceedings of International Conference on Communication and Signal Processing (ICCSPP)*, pages 0915–0919, Chennai, India, July 2020.
- [2] T. F. Sanam and H. Godrich. "FuseLoc: A CCA Based Information Fusion for Indoor Localization Using CSI Phase and Amplitude of Wifi Signals". In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7565–7569, Brighton, UK, May 2019.
- [3] Z. Shi, J. A. Zhang, R. Xu, Q. Cheng, and A. Pearce. "Towards Environment-Independent Human Activity Recognition using Deep Learning and Enhanced CSI". In *Proceedings of IEEE Global Communications Conference*, pages 1–6, Taipei, Taiwan, Dec. 2020.
- [4] X. Ding, T. Jiang, Y. Li, W. Xue, and Y. Zhong. "Device-Free Location-Independent Human Activity Recognition using Transfer Learning Based on CNN". In *Proceedings of IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6, Dublin, Ireland, Jun. 2020.
- [5] Z. Han, L. Guo, Z. Lu, X. Wen, and W. Zheng. "Deep Adaptation Networks Based Gesture Recognition using Commodity WiFi". In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–7, Seoul, Korea, May 2020.
- [6] J. Stuart Hunter. "The Exponentially Weighted Moving Average". *Journal of Quality Technology*, pages 203–210, 1986.
- [7] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. "Densely Connected Convolutional Networks". In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.
- [8] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu. "ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks". In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11531–11539, Seattle, WA, USA, Jun. 2020.
- [9] P. Haeusser, T. Frerix, A. Mordvintsev, and D. Cremers. "Associative Domain Adaptation". In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 2784–2792, Venice, Italy, Oct. 2017.
- [10] X. B. Peng, A. Kanazawa, S. Toyer, P. Abbeel, and L. Sergey. "Tool Release: Gathering 802.11n Traces with Channel State Information". *Association for Computing Machinery*, page 53, Jan. 2011.