DASECount: Domain-Agnostic Sample-Efficient Wireless Indoor Crowd Counting via Few-shot Learning

Huawei Hou, Suzhi Bi, Lili Zheng, Xiaohui Lin, Yuan Wu, and Zhi Quan

Abstract—Accurate indoor crowd counting (ICC) is a key enabler to many smart home/office applications. Recent development of WiFi-based ICC technology relies on detecting the variation of wireless channel state information (CSI) caused by human motions and has gained increasing popularity due to its low hardware cost, reliability under all lighting conditions, and privacy preservation in sensing data processing. To attain high estimation accuracy, existing WiFi-based ICC methods often require a large amount of labeled CSI training data samples for each application domain, i.e., a particular WiFi transceiver or background deployment. This makes large-scale deployment of WiFi-based ICC technology across dissimilar domains extremely difficult and costly. In this paper, we propose a Domain-Agnostic and Sample-Efficient wireless indoor crowd Counting (DASECount) framework that suffices to attain robust cross-domain detection accuracy given very limited data samples in new domains. DASECount leverages the wisdom of few-shot learning (FSL) paradigm consisting of two major stages: source domain meta training and target domain meta testing. Specifically, in the meta-training stage, we design and train two separate convolutional neural network (CNN) modules on the source domain dataset to fully capture the implicit amplitude and phase features of CSI measurements related to human activities. A subsequent knowledge distillation procedure is designed to iteratively update the CNN parameters for better generalization performance. In the meta-testing stage, we use the partial CNN modules to extract low-dimension features out of the high-dimension input target domain CSI data. With the obtained low-dimension CSI features, we can even use very few shots of target domain data samples (e.g., 5-shot samples) to train a lightweight logistic regression (LR) classifier, and attain very high cross-domain ICC accuracy. Experiment results show that the proposed DASECount method achieves over 92.68%, and on average 96.37% detection accuracy in a 0-8 people counting task under various domain setups, which significantly outperforms the other representative benchmark methods considered.

Index Terms—WiFi sensing, indoor crowd counting, cross-domain detection, few shot learning.

I. INTRODUCTION

AUTOMATIC indoor crowd counting (ICC) has important applications in a number of areas, such as public health management, security monitoring, and home/office automation. For instance, in the recent global outbreak of COVID-19,

H. Hou, S. Bi, L. Zheng, X. Lin, and Z. Quan are with the College of Electronics and Information Engineering, Shenzhen University, Shenzhen, China 518060 (e-mail: 2070436152@email.szu.edu.cn, {bsz,zhengll,xhlin,zquan}@szu.edu.cn). S. Bi and Z. Quan are also with the Peng Cheng Laboratory, Shenzhen, China 518066. (Corresponding Author: Suzhi Bi)

Y. Wu is with The State Key Lab of Internet of Things for Smart City, and also with the Department of Computer and Information Science, The University of Macau, Taipa, Macao SAR, China (e-mail: yuanwu@um.edu.mo).

ICC helps maintain social distancing in the indoor environment for effective epidemic prevention. Besides, knowing the exact number of people enables to fine-tune the air-conditioner for energy conservation and improve comfort in the indoor office environment. Existing ICC methods are mainly based on surveillance cameras, wearable sensors and radar, etc [1]. Among them, using cameras raises concerns on privacy violations and is highly susceptible to weak light conditions. On the other hand, ICC based on wearable sensors causes additional hardware overhead, e.g., a target needs to wear a special bracelet, which is costly and inconvenient for public or large-scale application scenarios. Although ICC based on radar equipment enjoys high detection accuracy when the radars are fined-tuned and properly deployed, the installation and hardware costs are uneconomic for extensive deployment in budget-limited home/office applications.

In recent years, there has been a growing interest in exploiting WiFi signals for indoor wireless sensing applications. By capturing the impact of human activity on the channel state information (CSI) between the WiFi transmitter and receiver, many indoor wireless sensing tasks can be effectively performed, such as human presence detection, activity and gesture recognition, respiration monitoring, as well as the focus of this paper, indoor crowd counting [2]–[5]. Compared with the above-mentioned ICC methods, WiFi has minimum privacy violation issues and works under any lighting conditions. Besides, WiFi routers are prevalent in home/office spaces, thus the hardware infrastructure is already established in most indoor environments. In addition, WiFi-based ICC takes a cost-efficient device-free approach and does not require the targets to wear additional sensors. Due to the above-mentioned technical advantages, WiFi-based ICC is expected to be widely used in future wireless sensing applications.

The existing WiFi-based ICC methods can be mainly divided into two categories, depending on the need of manual feature extraction [6]–[13]. One relies on explicit manual features engineered from raw data, such as mean value, variance, median, and range, and then uses threshold-based or learning-based classifiers like support vector machine (SVM) to identify the crowd number. The other takes a fully data-driven approach and relies on deep learning models to extract the implicit features from the raw data measurements and performs crowd counting accordingly. The performance of the former method is critically related to the data feature selection. For example, to select the best-performing features, Zou et al. [6] proposed a "Transfer Kernel Learning (TKL)" method that selects data

features based on a mutual information criterion from a feature pool including several statistical, transformation-based, and shape-based features. The performance of the latter method highly depends on iteratively training with a large number of labeled samples. For instance, the number of training samples used by WiCount [7] is more than 20000, which is difficult to implement in realistic application scenarios.

Therefore, despite the respective contributions of the above studies, the proposed methods suffer from a common drawback in practice. That is, although the well-trained deep learning models may achieve highly accurate ICC in one specific domain (even close to 100% accuracy), once used in a new and dissimilar environment, e.g., different transceiver or background deployment, the cross-domain detection accuracy often plummets. For instance, our experiments show that the accuracy decreases sharply from 99% to 12% after applying a deep learning model trained in a rich-scattering office environment to a more spacious conference room. To achieve high ICC accuracy in a new domain, the above methods often require training their models from scratch. In practice, this is indeed infeasible because of the prohibitively high cost of collecting and labeling a large number of data samples for each new domain encountered. To facilitate large-scale deployment in the future, the WiFi-based ICC method must be able to achieve high classification accuracy across different domains even if only a very limited number of samples are available in cross-domain scenarios.

In this article, we leverage the wisdom of few-shot learning (FSL) [14] to address the problems of model robustness and insufficient sample size in cross-domain ICC applications. In particular, we consider a practical scenario that the source domain has sufficient labeled training samples collected offline while the target domain only contains very few labeled samples. In this case, we propose a **D**omain-**A**gnostic and **S**ample-**E**fficient wireless crowd **C**ounting (DASECount) framework that can achieve high ICC accuracy in both source and target domains. To the authors' best knowledge, this is the first work that leverages FSL to achieve robust ICC performance across different domains. The main contributions of this paper are summarized as follows:

- We propose a DASECount framework for performing robust cross-domain ICC tasks. The DASECount framework includes two major stages: source domain meta training and target domain meta testing. In the meta-training stage, a priori deep learning CNN model is trained on datasets collected in a local source domain to extract features from CSI amplitude and phase input data. In the meta-testing stage, the well-trained CNN extracts the features of target domain data as the input to a tailor-made classifier, which eventually reports the final ICC result. The DASECount framework is particularly useful as it requires as few as only 5 labeled data samples in a dissimilar target domain to reach over 99% ICC accuracy.

- In the source domain meta-training stage, DASECount devises two separate data pre-processing procedures for CSI amplitude and phase data, respectively. After pre-processing, it uses two CNN-based feature extractors to derive the low-dimension amplitude and phase fea-

tures contained in the high-dimension input data, which facilitates training of the target domain classifier with very limited data samples. DASECount also applies a knowledge distillation technique to iteratively update the parameters of the CNN-based feature extractor, which improves at least 5% ICC accuracy by experiments.

- For target domain meta-testing, we first use the source domain feature extractor model to process the target domain training data. The output low-dimension features are considered as the input to train a lightweight classifier, e.g., a logistic regression model. By doing so, we can achieve high-performance cross-domain ICC even with very limited target domain training data size.

- We have conducted extensive experiments to evaluate the performance of the proposed DASECount framework in cross-domain ICC tasks. Results show that, with only 5 labeled target domain training data samples per class, DASECount achieves accuracy of over 97% in a 9-class ICC task when the crowd is stationary, over 99% when the crowd moving randomly, and over 92% in a more complex scenario with a mixture of stationary and moving crowds. We have also discussed the impact of detailed module designs in the proposed DASECount framework, e.g., selection of feature dimensions and classifier structures, on the cross-domain ICC performance. Overall, the proposed DASECount method attains robust and high accuracy in various cross-domain application scenarios.

II. RELATED WORKS

A. Learning-based WiFi ICC methods

In recent years, deep learning methods have been widely used in IoT applications, such as smart cities [15], Internet of Vehicles [16], etc. For deep learning-based WiFi ICC methods, Liu et al. proposed WiCount [7] model that implements a deep neural network (DNN) for CSI-based crowd counting and achieves 82.3% accuracy. The authors further improve the accuracy to 88.66% with a new DeepCount model [8] that combines conventional neural network (CNN) [17] and long short-term memory (LSTM) [18] structures. Xi et al. [9] exploited CSI phase information and built a Resnet-based [19] model, achieving on average 99% accuracy of human counting. Wand et al. [10] compared the performance of different deep learning networks, including CNN, LSTM, and gated recurrent unit [20], and showed that CNN achieves the best crowd counting accuracy.

Although human activity causes fluctuation of both amplitude and phase, many studies only use the amplitude information for ICC (like in [6], [11], [12]), mainly because the phase information often suffers from more severe hardware measurement noise such as carrier frequency offset and sampling time offset [21]–[23]. Instead of using raw phase measurements, Zong et al. [13] computed the phase difference between adjacent antennas as the input to an SVM-based ICC classifier. Liu et al. [24] utilized a CNN to extract the features of the CSI amplitude and phase information and use both features to detect human presence. It focuses on a binary
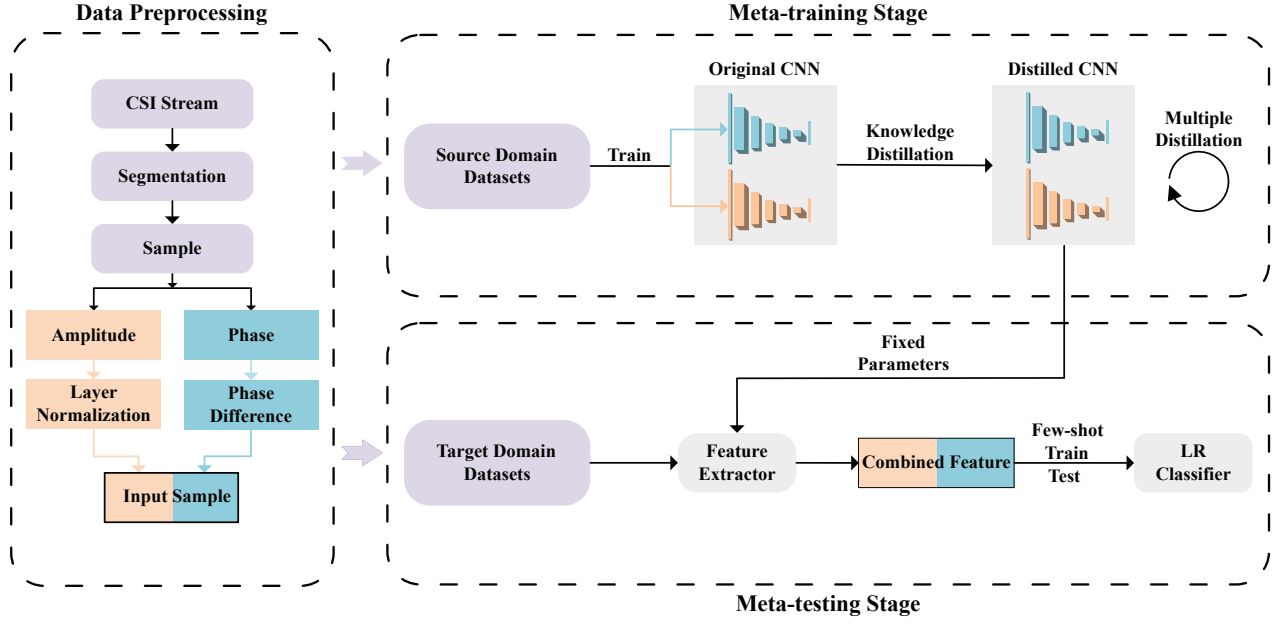
Fig. 1. Schematics of the proposed DASECount cross-domain ICC method. The amplitude and phase of CSI measurements in both source and target domains are first preprocessed. In the meta-training stage, a CSI feature extractor, consisting of two CNN submodels (for processing amplitude and phase input), is trained on the source domain dataset , followed by a distillation process to refine the CNN model parameters. The parameters of the CNN submodels are fixed after source-domain training. In the meta-testing stage, the few labeled data samples in the target domain are first processed by the feature extractor. The combined amplitude-phase feature output is then used as the input for training a lightweight logistic regression (LR) classifier in a supervised manner.

classification problem which is a simplified special case of the general crowd counting problem considered in this paper.

B. Few-shot Learning

FSL and cross-domain algorithms were originally developed and applied in the field of computer vision and have now been extended to multiple application fields [14], [25]–[30]. There are several popular models to perform FSL. For instance, matching networks [31] encodes the data of the source domain and target domain into a feature space by learning an embedding function. Then, it compares the similarity between the two through cosine similarity to determine which category of the target domain data belongs to. In recent years, researchers realized the importance of a priori models, that is, how a model that is fully trained and performs well on a task could be fine-tuned to handle a new task through learning with a limited number of samples. [25] proposed a meta-learning algorithm named "MAML", which is suitable for many popular learning models that apply gradient descent for parameter training. The model trained by MAML can be efficiently fine-tuned with target domain data samples and it shows high classification accuracy even under a very limited target domain training data set.

Different from fine-tuning global model parameters like MAML, [26] proposed a new method called "MTL", which first trains a deep neural network (DNN) in the source domain. In the target domain, it fixes the general mass of neurons and fine-tunes the other neurons by a few-shot training sample of the target domain classification task. In a more recent work [27], the authors took another FSL approach rather than fine-tuning the parameters of the well-trained model in the

source domain. Instead, it utilized a pre-trained model as a feature extractor to process the target domain few-shot training samples for training a lightweight machine learning model.

C. Applications of FSL to Wireless Sensing

FSL methods have been practiced for CSI-based wireless sensing applications such as human activity recognition and gesture recognition. Shi et al. [32] proposed a MaNet-eCSI architecture using a matching network for CSI-based human activity recognition which can achieve a cross-domain recognition accuracy of 92.3% with 5 training samples of the target domain. Zhang et al. [33] used MAML to train and fine-tune a 4-layer CNN for cross-domain human body activity recognition which reaches 89.6% accuracy with 5 samples of new activity datasets. [34] proposed a human activity recognition model named "CSI-GDAM", which uses a convolutional block attention module [35] layer to extract activity-related feature in CSI. CSI-GDAM reaches 99.74% accuracy in 5-shot cases for cross-domain activity recognition. For CSI-based gesture recognition, Yang et al. [36] proposed a novel deep Siamese neural networks [37] with multiple kernel variant of maximum mean discrepancies [38] for cross-domain gesture recognition. The method can achieve an accuracy of 89.5% with only 1 sample in the target domain.

It is worth noting that the above cross-domain wireless sensing methods mostly focus on fine-grained applications that classify human activities from a given set of patterns, such as a set of known gestures and body motions. In this case, the induced CSI variations are of a similar pattern and less sensitive to the background environment, thus high classification performance is likely achievable with a small

set of training samples in the new domain. In contrast to fine-grained applications, accurate cross-domain ICC tasks are more difficult because the CSI amplitude and phase variations caused by human free activities do not exhibit fixed patterns, instead are much more random and dependent on the domain environment. In this case, an ICC classifier trained with source domain data may face a severe over-fitting problem when applied to a new target domain. In this paper, we fully consider the unique challenge of cross-domain ICC tasks and propose a DASECount framework that provides robust cross-domain ICC performance.

III. CSI Signal Model and Preprocessing Method

In this section, we first introduce the WiFi sensing signal model and data format. Then we describe the CSI data pre-processing method to prepare input data for the cross-domain ICC tasks of the DASECount framework as shown in Fig. 1.

A. CSI Signal Model

In WiFi communication, channel state information (CSI) reflects the signal variations during transmission between the transmitter and receiver, including channel amplitude attenuation and phase shift [2]. The channel frequency response described by CSI is

$$\widetilde{H}(f;t) = \sum_{n=1}^{N} a_n(t) e^{-j2\pi f \tau_n(t)}, \tag{1}$$

where N represents the number of multipaths, $a_n(t)$ and $\tau_n(t)$ represent the amplitude attenuation and propagation delay in the nth path, and f denotes the carrier frequency. The receiving signal could be described as

$$Y(f;t) = \widetilde{H} \cdot X(f;t) + n(f;t), \tag{2}$$

where $X(f;t)$ is the transmitting signal in frequency f and at time t, Y is the conrresponding received signal, and n is the receiver noise.

IEEE 802.11a/b/n WiFi protocol supports multiple-input multiple-output (MIMO) and orthogonal frequency-division multiplexing (OFDM) transmissions. CSI acquisition requires specialized software operating on particular WiFi card chips. Two popular WiFi CSI acquisition softwares are the Intel 5300 CSI Tool [39] and the Atheros CSI Tool [40]. While the former supports 20Mhz bandwidth 30 subcarriers, the latter supports two operating modes: 20Mhz bandwidth 56 subcarriers and 40Mhz bandwidth 114 subcarriers. In this paper, we use the Atheros CSI tool to collect the CSI between a pair of transceivers with 2 transmitting and 3 receiving antennas, operating at 40MHz with 114 subcarriers. In this case, the collected CSI data is expressed as a 4-dimensional complex tensor $\bar{H} \in \mathbb{C}^{T \times N_r \times N_t \times N_{sc}}$, where T, N_r, N_t, N_{sc} are the number of time frames, receiving antennas, transmitting antennas, and subcarriers, respectively.

B. Proposed CSI Preprocessing Method

To facilitate subsequent processing by machine learning models, we propose the following preprocessing procedures on the collected CSI tensor data. As shown in Fig. 2, we first use a slide window of duration T_s to split the raw data into equal segment of duration T_w. The resulting CSI data within a tagged segment is expressed as $\widetilde{H} \in \mathbb{C}^{T_w \times N_r \times N_t \times N_{sc}}$. Here, we set $T_s < T_w$ to produce an overlap $T_w - T_s$ between the two adjacent segments, which brings two benefits: increase the number of training samples after segmentation and traverse the CSI variations caused by human movements in different periods. Then, we extract the amplitude data $\widetilde{H}^{amp}$ and phase data $\widetilde{H}^{pha}$ from each complex CSI data segment $\widetilde{H}$.
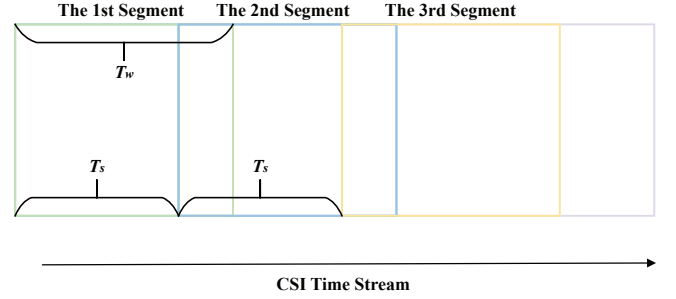

Fig. 2. Illustration of the data segmentation process. The shaded parts in the figure represent the overlap between two segments.

It is a common practice to impose noise reduction methods, such as Hampel [23] and low-pass filters [41], to amplitude data for fine-grained application scenarios like gesture recognition and respiratory monitoring. However, for a coarse-grained application like ICC, our empirical results show that amplitude noise reduction may lead to severe performance degradation as it may falsely remove the random high-frequency signal variations caused by simultaneous movements of multiple people. Therefore, we use the raw CSI amplitude data without applying the noise reduction technique.

Here, we first rearrange the amplitude data to a dimension of $N_{rt} \times T_w \times N_{sc}$, where $N_{rt} = N_r \cdot N_t$ denotes the number of parallel CSI between the transmit and receive antennas. Then, we process each $\widetilde{H}^{amp}$ with Layer Normalization [42]. Specifically, we denote $\hat{a}_{l,i,j}$ as the amplitude measurement corresponds to the lth Tx-Rx antenna pair, the ith time slot and the jth sub-carrier. Then, we compute the mean μ_l and the standard deviation σ_l of the $T_w \times N_{sc}$ amplitude measurements taken from the lth antenna pair. The normalization method for each amplitude measurement is expressed as

$$a_{l,i,j} = \frac{\hat{a}_{l,i,j} - \mu_l}{\sigma_l}, \forall l, i, j. \tag{3}$$

After layer normalization, we denote the amplitude data as H^{amp}.

For phase data processing, due to hardware impairment of WiFi chips, such as carrier frequency offset, and sampling time offset [24], the CSI phase data often change abruptly in adjacent time slots. Here, we first use the "unwrap" function to correct phase jump, and then compute the phase difference
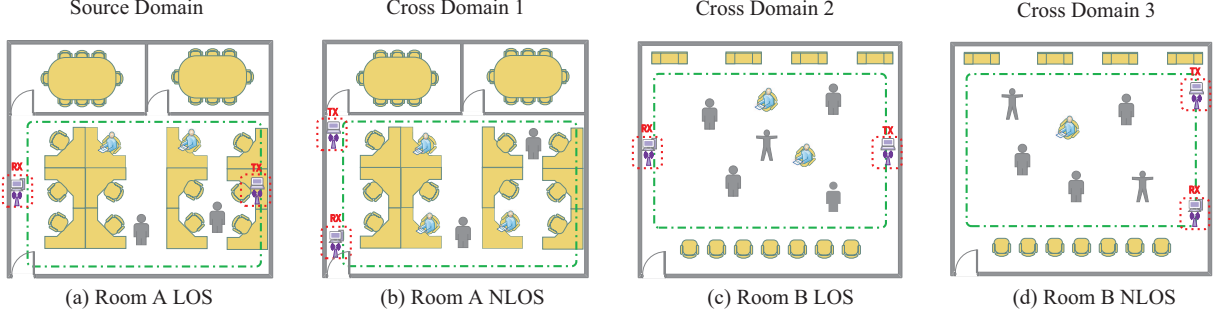
Fig. 3. Data collection scenarios. Data collection is conducted in 2 rooms, each containing LOS and NLOS scenarios.

TABLE I
NOMENCLATURE

Symbol	Terminology	Description
$\mathcal{S}$	Source domain dataset	Contains multiple ICC tasks of the source domain
$\mathcal{T}$	Target domain dataset	Contains multiple ICC tasks of the target domain
$\mathcal{D}^{train}$	Training set of $\mathcal{S}$	Training samples of an ICC task in the source domain
$\mathcal{D}^{val}$	Validating set of $\mathcal{S}$	Validating samples of an ICC task in the source domain
$\mathcal{D}^{sup}$	Support set of $\mathcal{T}$	Few-shot samples of an ICC task of the target domain, used to train a classifier
$\mathcal{D}^{que}$	Query set of $\mathcal{T}$	Used for evaluating the performance of the target domain classifier
x_*	Input sample	Includes the CSI amplitude part H_*^{amp} and the CSI phase difference part H_*^{phd}
y_*	Label	The ground-truth number of people in the scene
ϕ	CNN parameters	Contains amplitude and phase difference submodel ϕ^{amp}, ϕ^{phd}
ψ	Feature extractor	Generated by using partial parameters of ϕ
θ	Classifier	For specific ICC tasks in the target domain

between two adjacent receiving antennas to eliminate random phase noise [13], and denote the phase data after processing as H^{phd}.

With a bit abuse of notation, we denote the data samples in the ith segment as denoted as $x_i = (H_i^{amp}, H_i^{phd})$, where H_i^{amp} is the CSI amplitude part and H_i^{phd} is the CSI phase difference part. For each measurement x_i, we append a label $y_i \in \{0, 1, \cdots, M\}$ denoting the number of people in the test, where M denotes the maximum number of people considered.

IV. THE PROPOSED DASECOUNT FRAMEWORK

In this section, we introduce the DASECount framework for cross-domain ICC tasks. We divide the CSI data into source domain datasets and target domain datasets, respectively. The source domain dataset contains a large number of labeled sample sets collected from a local pre-set scene, while the target domain data set contains limited labeled data samples collected from the scene to be detected. For ICC tasks, CSI is often sensitive to the deployment of the WiFi transceivers and the surrounding environment. Without loss of generality, we consider a particular equipment deployment and room environment as the source domain scenario, and any significant change of equipment deployment or environment from the target domain leads to a new target domain scenario.

An example source-target domain setup is illustrated in Fig. 3. We consider two rooms where Room A is a rich scattering office room and Room B is a spacious conference venue. Besides, we also consider both line-of-sight (LOS) and non-line-of-sight (NLOS) WiFi equipment placements, where the human targets are in the LOS and NLOS channels of the WiFi transceivers, respectively. In total, there are four different scenarios, we consider without loss of generality that Room A LOS case as the source domain, and the rest three as target domains.

As shown in Fig. 1, after data collection and preprocessing, the proposed DASECount framework contains two main stages: the meta-training stage and the meta-testing stage. We will describe each stage below. The symbols involved are shown in Table I.

A. Source Domain Meta-training Stage

We denote the source domain dataset as $\mathcal{S} = (\mathcal{D}^{train}, \mathcal{D}^{val})$, where $\mathcal{D}^{train} = \{\mathcal{D}_s^{train}\}_{s=1}^{S} \triangleq (x_i, y_i)_{i=1}^{I}$ is the training set, $\mathcal{D}^{val} = \{\mathcal{D}_s^{val}\}_{s=1}^{S} \triangleq (\hat{x}_j, \hat{y}_j)_{j=1}^{J}$ is the validating set. Here, s represents the sth type of ICC task in the source domain. For example, in the simulation section, we consider $S = 3$ types of ICC tasks, where $s = \{1, 2, 3\}$ corresponds to ICC tasks when the targets are under static, dynamic, and a mixed static and dynamic motions, respectively. Besides, I and J denote the total number of data samples used for
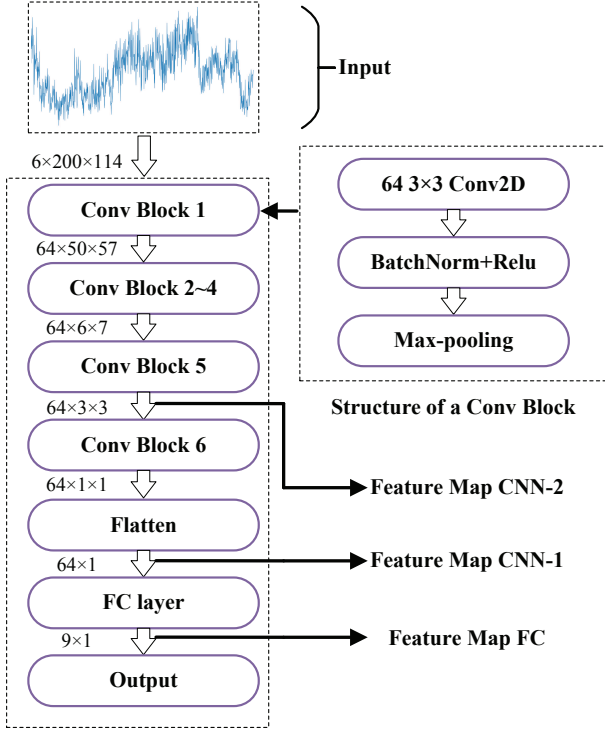
Fig. 4. The structure of the CNN stacks. The amplitude and phase difference submodels have the same CNN structure in the figure.

We present the training procedures of the amplitude and phase submodels in Algorithm 1.

Algorithm 1 Training Procedures of the Feature Extractor

Input:
   Merged training set $\mathcal{D}^{train}$;
Output:
   Feature extractor $\phi = (\phi^{amp}, \phi^{phd})$;
1: Initialize model parameters ϕ^{amp}, ϕ^{phd}, learning rate η.
2: **for all** H_i^{amp} of $x_i \in \mathcal{D}^{train}$ **do**
3:     Calculate the output $\hat{y}_i^{amp} = f(H_i^{amp}; \phi^{amp})$
4:     Calculate $\mathcal{L}^{ce}(\hat{y}_i^{amp}, y_i)$
5:     Update $\phi^{amp} = \phi^{amp} - \eta\nabla_{\phi^{amp}}\mathcal{L}^{ce}(\hat{y}_i^{amp}, y_i)$
6: **end for**
7: **for all** H_i^{phd} part of $x_i \in \mathcal{D}^{train}$ **do**
8:     Calculate the output $\hat{y}_i^{phd} = f(H_i^{phd}; \phi^{phd})$
9:     Calculate $\mathcal{L}^{ce}(\hat{y}_i^{phd}, y_i)$
10:     Update $\phi^{phd} = \phi^{phd} - \eta\nabla_{\phi^{phd}}\mathcal{L}^{ce}(\hat{y}_i^{phd}, y_i)$
11: **end for**
12: Output ϕ^{amp}, ϕ^{phd}

To improve the generalization capability of the feature extractor, we apply the knowledge distillation technique [43], which has shown effective performance improvement of FSL problem for image classification [27]. We treat the original trained CNNs in Algorithm 1 as the initial teacher model ϕ_0, and iteratively distill knowledge from the teacher model to a student model using the same source domain training dataset.
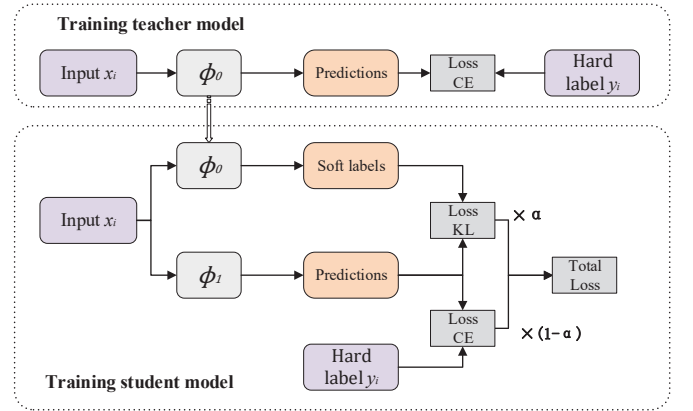

Fig. 5. The model knowledge distillation procedure. The figure shows the process of training the 1st generation distilled model. The total loss is used to update the model parameters of ϕ_1.

training and validation, respectively. The detailed descriptions are presented in Section V-A.

In the meta-training stage, we first train a CSI feature extractor using the source domain dataset $\mathcal{S}$ by supervised learning. We denote the model parameters of the feature extractor as ϕ, which contains two sub-models, one processes CSI amplitude information denoted as ϕ^{amp} and the other processes CSI phase difference information denoted as ϕ^{phd}. Different from [24], where amplitude and phase modules are concatenated by a fully connected layer, the amplitude and phase difference submodels of DASECount are independent and described below.

To fully exploit both the chronological and subcarriers correlations, we apply a 2D CNN consisting of 6 convolutional blocks and a fully connected layer, as shown in Fig. 4. Each convolutional block contains a convolutional layer (64 3×3 convolution 2D kernels), a batch normalization layer, a Relu activation function, and a max-pooling layer. The number of neurons in the fully connected layer is the same as the number of sample classes. It is worth mentioning that the first pooling layer has a kernel of 4×2 and all the others have a kernel of 2×2.

The process of training ϕ is expressed as follows:

$$\phi = \arg\min_{\phi} \mathcal{L}^{ce}(\mathcal{D}^{train}; \phi), \tag{4}$$

where $\mathcal{L}^{ce}$ represents the cross-entropy loss between source domain data and corresponding labels. We train the ϕ on the training set $\mathcal{D}^{train}$ and evaluate it on the validating set $\mathcal{D}^{val}$.

In the proposed DASECount framework, the teacher and student models have the same structure, and we use a fixed training set $\mathcal{D}^{train}$ as the input to all the models. As shown in Fig. 5, suppose that the input for training models is x_i and the corresponding label is y_i. Let $f(x_i; \phi_0)$ denote the output of the initial teacher network given an input x_i. To obtain the first distilled model ϕ_1, we use y_i as the hard label and $f(x_i; \phi_0)$ as the soft label of the input x_i. Similarly, to generate the kth distilled model ϕ_k, we solicit both the hard label y_i and the soft label $f(x_i; \phi_{k-1})$ to the input x_i and minimize the weighted loss caused by both the hard and soft labels [44].

Therefore, the distillation of the kth model can be written as follows:

$$\phi_k = \arg\min_{\phi}(\alpha \mathcal{L}^{ce}(\mathcal{D}^{train};\phi)+$$
$$(1-\alpha)KL(f(\mathcal{D}^{train};\phi), f(\mathcal{D}^{train};\phi_{k-1}))), \quad (5)$$

where $\alpha \in [0,1]$ is the weight of cross-entropy (CE) loss, KL represents the Kullback-Leibler divergence between two distributions. Both the amplitude and phase CNN models are distilled several times. The parameter update of model distillation is shown in Algorithm 2. After distillation, we have obtained a series of distilled model $\{\phi_k\}_{k=0}^{K}$. We choose one model $\phi_m = (\phi_m^{amp}, \phi_m^{phd})$ to generate the final CSI feature extractor, denoted as g_ψ, where ψ represents the model parameters. We will demonstrate the selection method of the distilled models and the advantage of distillation to the cross-domain ICC performance in Section V-C6.

Algorithm 2 Distillation Procedure of Feature Extractor

Input:
 Merged training set $\mathcal{D}^{train}$;
 Initial feature extractor $\phi_0 = (\phi_0^{amp}, \phi_0^{phd})$;
Output:
 $(\phi_k^{amp}, \phi_k^{phd})_{k=0}^{K}$;
1: **for** k=1:K **do**
2:   Initialize the kth model $\phi_k = (\phi_k^{amp}, \phi_k^{phd})$
3:   **for all** H_i^{amp} of $x_i \in \mathcal{D}^{train}$ **do**
4:     Calculate the output $f(H_i^{amp};\phi_k^{amp})$
5:     Obtain soft label $f(H_i^{amp};\phi_{k-1}^{amp})$
6:     Update ϕ_k^{amp} by the Equation (5)
7:   **end for**
8:   **for all** H_i^{phd} of $x_i \in \mathcal{D}^{train}$ **do**
9:     Calculate the output $f(H_i^{phd};\phi_k^{phd})$
10:     Obtain soft label $f(H_i^{phd};\phi_{k-1}^{phd})$
11:     Update ϕ_k^{phd} by the Equation (5)
12:   **end for**
13: **end for**

B. Target Domain Meta-testing Stage

After obtaining the distilled CSI feature extractor in the source domain, we select the output of a particular layer as the feature to train the target domain classifier. For instance, as illustrated in Fig. 4, the choice can be the feature map of FC, CNN-1, CNN-2, etc, and we leave the discussion of feature map selection in Section V-C4. For the target domain, we denote the dataset as $\mathcal{T} = (\mathcal{D}^{sup}, \mathcal{D}^{que})_{t=1}^{T}$, where $\mathcal{D}^{sup} = \{\mathcal{D}_t^{sup}\}_{t=1}^{T} \triangleq (x_t^p, y_t^p)_{p=1}^{P}$ is the support set, $\mathcal{D}^{que} = \{\mathcal{D}_t^{que}\}_{t=1}^{T} \triangleq (x_t^q, y_t^q)_{q=1}^{Q}$ is the query set and t represents the tth ICC task in the target domain. Because the support set contains very limited labeled training samples, we use a shallow classifier parameterized by θ. Some example classifiers include logistic regression (LR) and support vector machine (SVM), etc. In particular, we minimize the cross-entropy loss with the FSL training samples of the support set:

$$\theta = \arg\min_{\theta} \mathcal{L}^{ce}(\mathcal{D}^{sup};\theta). \quad (6)$$

Using LR as the classifier, the meta-testing procedure in the target domain is shown in Algorithm 3. Finally, we evaluate the performance of the classifier in the query set $\mathcal{D}^{que}$.

Algorithm 3 Training procedure of target domain classifier

Input:
 Support set $\mathcal{D}^{sup}$ of an ICC task in the target domain;
 Feature extractor g_ψ;
Output:
 An LR classifier θ for the ICC task;
1: **for all** few-shot samples $x_t^p \in \mathcal{D}_t^{sup}$ **do**
2:   Compute feature map $\Phi_p = g_\psi(x_t^p)$
3:   Reshape feature map to 1D vector and augment 5 times

4:   Training the LR classifier:
5:   $\theta = \theta - \eta(-\frac{1}{P}\sum_{p=1}^{P}[(y_t^p - \frac{1}{1+e^{-W^T\Phi_p}})\Phi_p])$
6: **end for**

V. EXPERIMENT RESULTS

In this section, we perform experiments to evaluate the performance of the proposed DASECount framework. We first describe the experiment setups and parameter settings in Section V-A. Then, we show the performance of the CNN-based feature extractor in handling source domain ICC tasks Section V-B. In Section V-C, we present the results of cross-domain ICC tasks, where we compare DASECount with other benchmark methods and discuss design factors that influence the performance.

A. Experiment Setups

We use a laptop as the transmitter and a desktop as the receiver, where both devices communicate with Atheros 802.11n WiFi card (AR9580/AR9382). Meanwhile, both devices run Ubuntu 14.04 system and the receiver uses the Atheros CSI tool to collect the CSI. The transmission works in the 2.4GHz spectrum, occupying 40MHz bandwidth with 114 subcarriers, using 2 transmitting antennas and 3 receiving antennas. The transmit rate is set to 100 pkts/s. All the collected data packets are parsed by the Atheros CSI tools and further processed by MATLAB. All the data processing and computations are performed on a Dell PowerEdge T640 server with 256GB of RAM and a Tesla P100 GPU.

For cross-domain ICC task setups, we conduct experiments in 2 rooms where each room contains a line-of-sight (LOS) and a none-line-of-sight (NLOS) experimental scenario, resulting 4 different scenarios in total. As shown in Fig. 3, room A is an office space while room B is a lecture hall. The considered cross-domain scenario setup is similar to that in [45].

To measure the effects of different types of human activity on performance, we consider the following three motion types and collect data for each type when 0 to 8 (i.e., 9 classes) volunteers are in the room.

- Static: volunteers are required to remain seated but can act freely, such as eating, typing, or sleeping;
- Dynamic: volunteers walk randomly walk around the venue;

- Mixed: there is no restriction to the volunteers' activities, and they can move freely in the venue including but not limited to walking, sitting, eating, and sleeping.

CSI data of each category (4 scenarios × 3 motion types × 9 classes = 54 categories in total) are collected for 5 minutes to obtain a total of about 30000pkt for each category. In the CSI pre-processing stage, we set segmentation window T_w as 200 (i.e., 2 seconds as a unit) and sliding window T_s as 50 (i.e., 0.5 seconds), so we obtain 600 segments for each category.

After preprocessing, we have obtained CSI data of 3 motion types in each scenario, where each scenario-motion pair has 5400 samples (600 samples × 9 classes). We treat these different motion types as different ICC tasks in each scenario (i.e., subscript s for source domain and t for target domain in IV-A). Without loss of generality, data samples collected under Room A LOS scenario are used as source domain dataset $\mathcal{S}$ and samples in other 3 scenarios are used as target domain datasets $\mathcal{T}$.
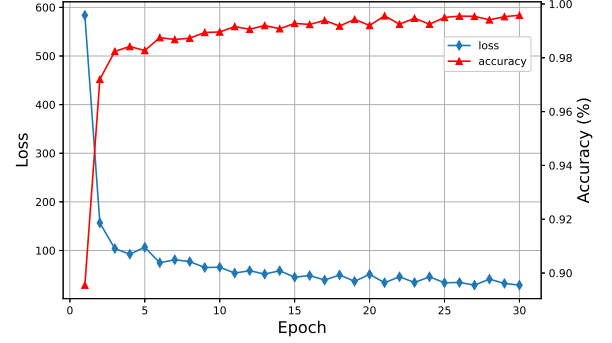
B. Feature Extractor Configuration

We generate the feature extractor in the source domain dataset $\mathcal{S}$, which contains data of all the three motion types. We train a unified feature extractor rather than one for each of the 3 motion types. Therefore, we combine the samples of all motion types to build a united dataset, and then divide it into training set (i.e., $\mathcal{D}^{train}$ in IV-A)) and validating set (i.e., $\mathcal{D}^{val}$ in IV-A) in a ratio of 9 to 1. Hence, the training set $\mathcal{D}^{train}$ contains $3 \times 9 \times 540 = 14580$ samples and the validating set $\mathcal{D}^{val}$ contains $3 \times 9 \times 60 = 1620$ samples. The training of the feature extractor is carried out on $\mathcal{D}^{train}$. Some main training parameters involved are shown in Table II. Both the amplitude and phase difference submodels are trained with the same training parameters.

TABLE II
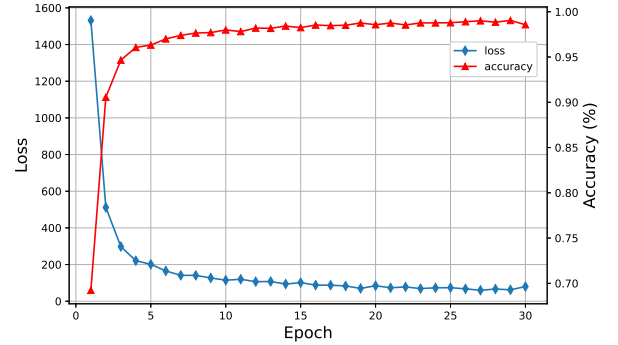TRAINING PARAMETERS OF FEATURE EXTRACTOR

Batch size	Epochs	Learning rate	Optimizer
8	30	10^{-3}	Adam

After training, we evaluate the ICC performance of the amplitude and phase difference submodels on the source domain validation set $\mathcal{D}^{val}$. In Fig. 6, we show the loss and accuracy variations in 30 training epochs. As can be seen from Fig. 6a, the training loss of the amplitude submodel drops rapidly in the first 10 epochs and gradually converges after 30 epochs. The accuracy of the model reaches 98% on the validation set. The loss and accuracy of the phase difference submodel vary similarly to the amplitude submodel during the training process.

Besides, we perform knowledge self-distillation 6 times to the original feature extractor, obtaining 7 generation models in total. Some main distillation parameters involved are shown in the table III. Cross entropy loss weight α in Equation (5) is set to 0.5. Among these models, we select a model with the best ICC accuracy as the target domain feature extractor, i.e., the 4th generation model.

(a) Amplitude Submodel

(b) Phase difference Submodel

Fig. 6. Training loss and ICC accuracy of the amplitude and phase difference submodels in the meta-training stage.

TABLE III
DISTILLATION PARAMETERS OF FEATURE EXTRACTOR

Batch size	Epochs	Learning rate	Optimizer	Weight decay
100	100	10^{-3}	SGD	$5 \cdot 10^{-4}$

C. Performance of DASECount

In the meta-testing stage, k-shot samples in the target domain are input to the obtained extractor to compute the corresponding feature maps, where $k \in \{1, 5\}$ correspond to 1-shot and 5-shot cases, respectively. Here, we choose the feature maps from the penultimate convolutional block to train and evaluate the target domain LR classifier with 1-shot learning and 5-shot learning (the feature map CNN-2 in Fig.4). As shown in Fig. 4, the $64 \times 3 \times 3$ feature map is flattened to a 1×576 vector. After concatenating features from the amplitude and the phase difference submodels, we have obtained a 1×1152 feature vector. Then, each k-shot sample vector is duplicated five times. For 1-shot learning, only 1 training sample is available for each class of the 9 classes. After duplication, the dimension of training data is 45×1152. For 5 shot learning, 5 samples are available for each class so that the dimension of training data is 225×1152, accordingly.

1) ICC accuracy with LR classifier: Table IV shows 1-shot and 5-shot results of the LR classifier with the 4th generation distillation feature extractor. All the accuracy result is an average obtained by repeating the experiment 10 times.
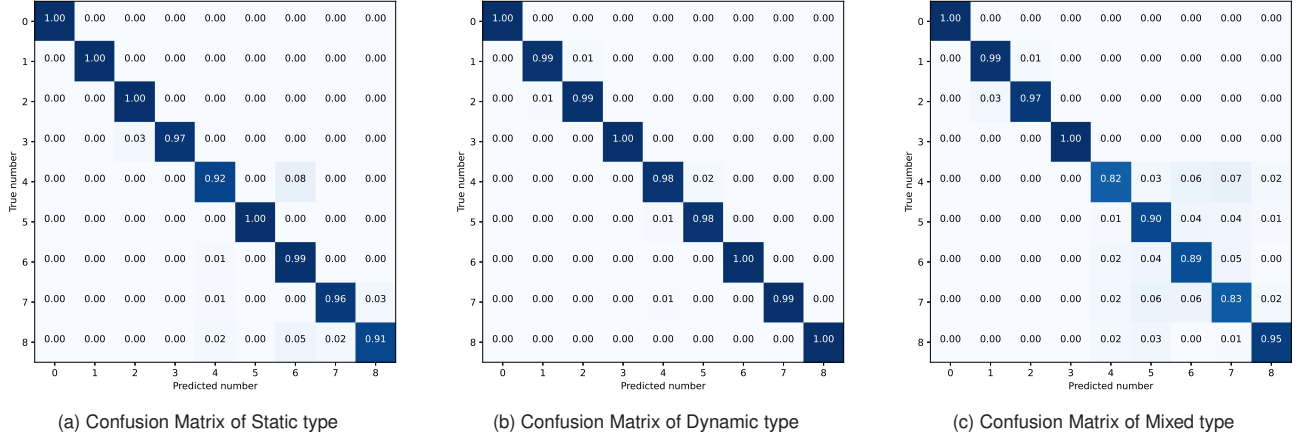
(a) Confusion Matrix of Static type   (b) Confusion Matrix of Dynamic type   (c) Confusion Matrix of Mixed type

Fig. 7. Confusion matrix of classification results in the Room B NLOS scenario.

TABLE IV
FEW-SHOT RESULTS FOR META-TESTING SCENARIOS

		Room A NLOS	Room B LOS	Room B NLOS
1 shot	Static	90.05%	88.63%	85.64%
	Dynamic	94.63%	93.41%	93.29%
	Mixed	80.65%	77.17%	76.26%
5 shot	Static	98.29%	97.83%	97.26%
	Dynamic	98.33%	98.94%	99.17%
	Mixed	96.85%	94.61%	92.68%

Fig. 8. Detection accuracy comparisons of different cross-domain ICC methods. *CNN_AMP* and *CNN_PHD* represent the CNN amplitude and phase difference feature extractor submodels, respectively. *LR* is a logistic regression classifier directly trained with 5 samples. *DASECount* is the proposed method.

We see that, compared with 1-shot learning, the accuracy improvement of 5-shot learning ranges from the lowest 4% (Dynamic type of Room A NLOS) to the highest 16% (Mixed type of Room B NLOS), which matches the intuition that increasing the number of shots improves detection performance. For the Room A NLOS scenario, the LR classifier achieves an average accuracy of 97.82% with 5-shot learning. The high accuracy is because only the transceiver deployment is changed compared to the source domain. As for the Room B NLOS scenario, the average detection accuracy drops slightly (about 1.5%) because of the change of the entire surrounding environment. From the perspective of motion types, the accuracy is higher (average 98.81% with 5-shot learning) with the Dynamic type and lower (average 94.71% with 5-shot learning) with the Mixed type, because of the higher degree of randomness in the CSI measurements under the Mixed motion type of the crowd.

Fig. 7 shows the confusion matrixes of ICC results in the Room B NLOS scenario. The (i, j)th element denotes the probability that the ground-truth i people is identified as j people. We see that the detection accuracy decreases slightly when more people participate in the experiment. Overall, DASECount achieves very high accuracy (i.e.,over 99%) where the error margins are mostly within 1-2 people. If we consider a presence detection problem, i.e., detecting whether there are any people in the room, with 5-shot learning, the proposed DASECount method achieves 100% accuracy.
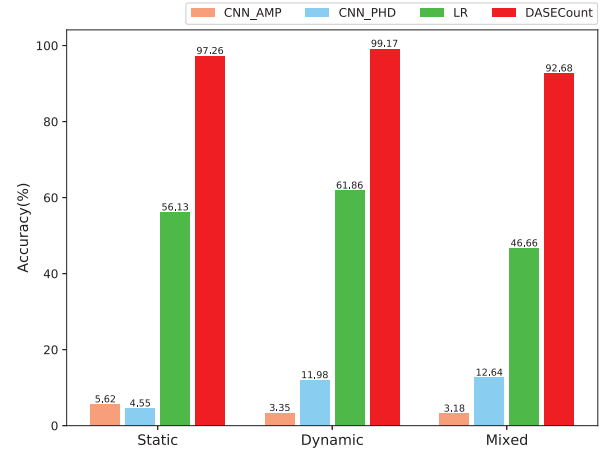
2) Advantage of cross-domain learning: To evaluate the performance of the proposed DASECount method, we present the classification accuracy when the following three benchmark classifiers are used in the Room B NLOS scenario:

(1) Source-domain CNN amplitude feature extractor;
(2) Source-domain CNN phase difference feature extractor;
(3) Directly train the LR classifier based on the 5 samples;

Fig. 8 shows the comparison result. If we directly apply the CNN feature extractor to the target domain ICC task without FSL, the accuracy is very poor, this verifies our claim that environment difference has a large impact on the ICC performance. The LR classifier is trained with raw CSI data (flattening CSI data to a vector without feature extractor processing). It achieves an average accuracy of 54.88% in the scenario. This is because the few target-domain data samples are not sufficient to train a classifier with high-dimension of raw input data. Overall, the proposed DASECount framework outperforms the benchmark methods by at least 30% in all the motion types considered.

TABLE V
INFLUENCE OF COMBINED FEATURES

		AMP	PHD	DASECount
1 shot	Static	81.15%	74.79%	**85.64**%
	Dynamic	91.45%	76.28%	**93.29**%
	Mixed	67.56%	59.04%	**76.26**%
5 shot	Static	90.89%	91.82%	**97.26**%
	Dynamic	95.05%	91.50%	**99.17**%
	Mixed	88.20%	84.91%	**92.68**%

3) Advantage of combined amplitude and phase features:
DASECount combines features in CSI amplitude and phase
difference information from the target domain to train a
lightweight LR classifier. Since the feature extractor contains
independent amplitude and phase difference submodels, the
target domain LR classifier can be trained using features
extracted from one of the submodels alone. Specifically, the
vector length of the joint feature (1×1152) is twice as long
as that of the single amplitude or phase feature (1×576).
Table V shows the comparison results of LR classifiers in
the Room B NLOS scenario, where the *AMP* represents the
LR classifier is trained only with the amplitude feature and
the *PHD* represents the LR classifier is trained only with the
phase difference feature. Compared with amplitude or phase
difference feature alone, the accuracy improvement of the
LR classifier trained on joint features ranges from the lowest
1.84% (Dynamic type) to the highest 17.22% (Mixed type) for
1-shot learning. For 5-shot learning, the accuracy improvement
ranges from the lowest 4.12% (Dynamic type) to the highest
7.77% (Mixed type).

4) Selection of feature map: The CNN feature extractor of
DASECount has 6 convolutional blocks and a fully connected
layer, and the extracted features of each layer have different
dimensions and contain different information. We have tested
the features from the last two convolutional blocks and the
fully connected layer to train the LR classifier. Table VI shows
the performance of LR classifiers trained with 3 different kinds
of features in the Room B NLOS scenario. Compared with
features from the fully connected layer (*FC* in the table) and
the final convolutional block (*CNN-1* in the table), features
from the penultimate convolutional block result in the best
ICC accuracy in all cases.

TABLE VI
INFLUENCE OF FEATURE MAP FROM CNN

		FC(1×18)	CNN-1(1×128)	CNN-2(1×1152)
1 shot	Static	61.01%	66.18%	**85.64**%
	Dynamic	71.01%	84.41%	**93.29**%
	Mixed	39.03%	60.27%	**76.26**%
5 shot	Static	83.92%	90.14%	**97.26**%
	Dynamic	87.54%	93.39%	**99.17**%
	Mixed	79.14%	83.79%	**92.68**%

5) Selection of classifier model: We compare different
target domain classifiers: logistic regression (LR), support
vector machine, and K-nearest neiber (NN) in the Room B
NLOS scenario. Experiment results in Table VII show the

three machine learning classifiers have similar performance,
while the LR classifier is slightly better than the other two by
about 2% detection accuracy.

TABLE VII
PERFORMANCE OF DIFFERENT MACHINE LEARNING CLASSIFIERS

		LR	SVM	NN
1 shot	Static	**85.64**%	79.93%	79.60%
	Dynamic	**93.29**%	91.59%	87.01%
	Mixed	**76.26**%	73.06%	72.41%
5 shot	Static	**97.26**%	96.07%	95.12%
	Dynamic	**99.17**%	98.78%	98.18%
	Mixed	**92.68**%	91.78%	88.59%

6) Effect of distillation: In the meta-training stage, we
have distilled the amplitude and phase difference submodels
6 times and obtained 7 generations of models in total. We
conduct experiments under the Mixed type ICC in the Room
B NLOS scenario and evaluate the performance of the target
domain LR classifier using different generation models as
the feature extractor. The influence of different distillation
generation models is shown in Fig. 9. As can be seen from the
figure, compared with the model of generation 0, the accuracy
of the target domain classifier can be improved by 3-10%
after several rounds of distillations, but the performance of
the proposed DASECount drops after 4 rounds of distillation.
Therefore, we need to track the ICC accuracy of each round
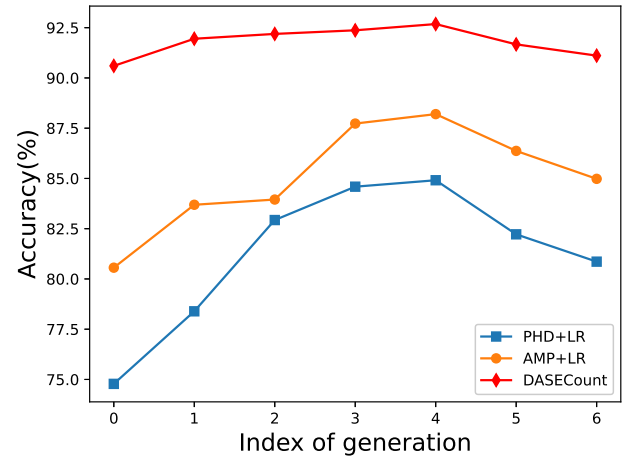of distillation and select the best one to produce the feature
extractor.

Fig. 9. Influence of the generation of knowledge distillation on the perfor-
mance of target domain classifier. *PHD+LR* represents the phase difference
submodel as the feature extractor. *AMP+LR* represents the amplitude sub-
model as the feature extractor. *DASECount* represents the proposed method,
which combines both of amplitude and phase difference submodels as the
feature extractor.

7) Compared with MAML FSL method: Fig. 10 shows the
comparison in scenario Room B NLOS with 5-shot learning.
In particular, we compare the performance of the proposed
FSL-based DASECount method with the well-known MAML
method. It shows that the accuracy of the proposed DASEC-
ount method is on average 27.86% higher than MAML when

using only amplitude as the input measurement, and 13.6% higher when using only amplitude as the input measurement. In both cases, the proposed method significantly outperforms the benchmark MAML. Besides, we can further improve the accuracy by combing the phase and amplitude features under the proposed DASECount framework.
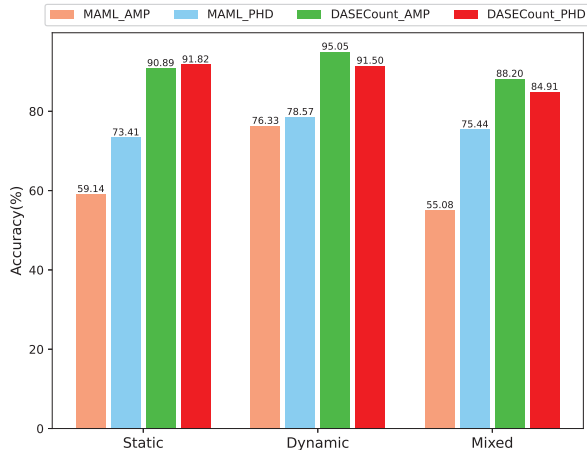

Fig. 10. Detection accuracy comparisons of proposed DASECount and MAML in Room B NLOS scenario with 5-shot learning.

VI. CONCLUSIONS AND DISCUSSIONS

In this paper, we have proposed a DASECount framework based on FSL to achieve highly accurate and robust cross-domain ICC performance. DASECount contains a meta-training stage and a meta-testing stage. In the meta-training stage, DASECount trains a CSI feature extractor consisting of the amplitude and phase difference CNN submodels with supervised learning. We also applied a knowledge distillation procedure to iteratively update the parameters of the CNN submodels for better generalization performance. In the meta-testing stage, thanks to the feature extractor that generates low-dimension features of the target domain data, DASECount attains very high cross-domain ICC accuracy with a simple lightweight LR classifier given very limited target domain data samples. Experimental results show that the proposed DASECount achieves over 92.68%, and on average 96.37% detection accuracy, in a 0-8 people counting task under various domain setups, which significantly outperforms the other representative benchmark methods considered. Overall, the proposed DASECount framework significantly enhances the robustness of cross-domain ICC tasks and reduces the operating cost in large-scale deployment of future WiFi-based indoor sensing applications.

Notice that the training of feature extractor is performed offline just once on the source domain dataset. We can therefore perform the training on a powerful server using the sufficiently large source domain dataset, and fix the parameters of the feature extractor after the training converges. In our simulations, the complete CNN feature extractor contains 189513 training parameters and the training process converges in less than 5 minutes. In the target domain, we reuse the obtained feature extractor and only need to train a lightweight LR classifier consisting of only 1153 training parameters. Besides, the training is performed on a very small target domain data set following the FSL paradigm. Therefore, the classifier in the target domain can be quickly trained with the few-shot samples, and it takes less than 1 second in our simulations to complete the training. Overall, the proposed DASECount framework can be quickly extended to perform ICC in new target domains with very low computational complexity.

It is an important working direction for us to further improve the robustness of DASECount in some application cases. For example, if the location of the WiFi transceiver or the background environment of devices changes significantly, it may cause sample data distribution shift and affect the training performance of the classifier. In this case, it requires recollecting labeled data samples and retraining the classifier, which however is very costly if the change happens frequently. A more feasible yet much more challenging solution is for DASECount to adapt to new domains with an unsupervised learning method. A promising method is to design a zero-shot framework based on a generative adversarial network (GAN), which requires no labeled data at all. This is considered as an important future work.

REFERENCES

[1] T. Teixeira, G. Dublon, and A. Savvides, "A survey of human-sensing: Methods for detecting presence, count, location, track, and identity," *ACM Computing Surveys*, vol. 5, no. 1, pp. 59–69, Jan. 2010.

[2] Y. Ma, G. Zhou, and S. Wang, "Wifi sensing with channel state information: A survey," *ACM Computing Surveys*, vol. 52, no. 3, pp. 1–36, Jun. 2019.

[3] J. Liu, H. Liu, Y. Chen, Y. Wang, and C. Wang, "Wireless sensing for human activity: A survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1629–1645, Aug. 2019.

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

[5] A. Khalili, A.-H. Soliman, M. Asaduzzaman, and A. Griffiths, "Wi-fi sensing: applications and challenges," *The Journal of Engineering*, vol. 2020, no. 3, pp. 87–97, Feb. 2020.

[6] H. Zou, Y. Zhou, J. Yang, W. Gu, L. Xie, and C. Spanos, "Freecount: Device-free crowd counting with commodity wifi," in *Proc. IEEE GLOBECOM*, Dec. 2017, pp. 1–6.

[7] S. Liu, Y. Zhao, and B. Chen, "Wicount: A deep learning approach for crowd counting using wifi signals," in *Proc. IEEE ISPA/IUCC*, Dec. 2017, pp. 967–974.

[8] S. Liu, Y. Zhao, F. Xue, B. Chen, and X. Chen, "Deepcount: Crowd counting with wifi via deep learning," 2019, *arXiv:1903.05316*. [Online]. Available: http://arxiv.org/abs/1903.05316

[9] J. Xi, Z. Xu, L. Chen, and J. Li, "Human counting and action recognition with wifi via deep learning," in *Proc. IEEE CSRSWTC*, Mar. 2020, pp. 1–3.

[10] Z. Wang, J. Fan, X. Song, N. Zhou, F. Chen, Y. Guo, and D. Chen, "Crowd counting based on csi and convolutional neural network," in *Proc. IEEE CCDC*, May. 2021, pp. 1249–1254.

[11] S. Di Domenico, M. De Sanctis, E. Cianca, and G. Bianchi, "A trained-once crowd counting method using differential wifi channel state information," in *Proc. ACM WPA*, Jun. 2016, pp. 37–42.

[12] Y.-K. Cheng and R. Y. Chang, "Device-free indoor people counting using wi-fi channel state information for internet of things," in *Proc. IEEE GLOBECOM*, Dec. 2017, pp. 1–6.

[13] J. Zong, B. Huang, L. He, B. Yang, and X. Cheng, "Device-free crowd counting based on the phase difference of channel state information," in *Proc. IEEE ICIBA*, Nov. 2020, pp. 1343–1347.

[14] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM CSUR*, vol. 53, no. 3, pp. 1–34, Mar. 2020.

[15] Q. Chen, W. Wang, F. Wu, S. De, R. Wang, B. Zhang, and X. Huang, "A survey on an emerging area: Deep learning for smart city data," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 3, no. 5, pp. 392–410, May. 2019.

[16] J. Chen, K. Li, and P. S. Yu, "Privacy-preserving deep learning model for decentralized vanets using fully homomorphic encryption and blockchain," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 11 633–11 642, Aug. 2022.

[17] 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.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, Jun. 2016, pp. 770–778.

[20] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: http://arxiv.org/abs/1406.1078

[21] L. Gong, W. Yang, Z. Zhou, D. Man, H. Cai, X. Zhou, and Z. Yang, "An adaptive wireless passive human detection via fine-grained physical layer information," *Ad Hoc Networks*, vol. 38, pp. 38–50, Mar. 2016.

[22] S. Palipana, P. Agrawal, and D. Pesch, "Channel state information based human presence detection using non-linear techniques," in *Proc. ACM BuildSys*, Nov. 2016, pp. 177–186.

[23] H. Zhu, F. Xiao, L. Sun, R. Wang, and P. Yang, "R-ttwd: Robust device-free through-the-wall detection of moving human with wifi," *IEEE JSAC.*, vol. 35, no. 5, pp. 1090–1103, Mar. 2017.

[24] Y. Liu, T. Wang, Y. Jiang, and B. Chen, "Harvesting ambient rf for presence detection through deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 4, pp. 1571–1583, Dec. 2020.

[25] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. PMLR ICML*, Aug. 2017, pp. 1126–1135.

[26] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *Proc. IEEE CVPR*, Jun. 2019, pp. 403–412.

[27] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola, "Rethinking few-shot image classification: a good embedding is all you need?" in *Proc. Springer ECCV*, Aug. 2020, pp. 266–282.

[28] C. Chen, K. Li, W. Wei, J. T. Zhou, and Z. Zeng, "Hierarchical graph neural networks for few-shot learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 1, pp. 240–252, Jan. 2022.

[29] H. Zheng, R. Wang, Y. Yang, J. Yin, Y. Li, Y. Li, and M. Xu, "Cross-domain fault diagnosis using knowledge transfer strategy: A review," *IEEE Access*, vol. 7, pp. 129 260–129 290, Sep. 2019.

[30] J. Chen and P. S. Yu, "A domain adaptive density clustering algorithm for data with varying density distribution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 6, pp. 2310–2321, Nov. 2021.

[31] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," in *Proc. Curran Associates, Inc. NIPS*, vol. 29, Dec. 2016, pp. 3630–3638.

[32] Z. Shi, J. A. Zhang, Y. D. R. Xu, and Q. Cheng, "Environment-robust device-free human activity recognition with channel-state-information enhancement and one-shot learning," *IEEE Trans. Mobile Comput.*, vol. 21, no. 2, pp. 540–554, Jul. 2022.

[33] Y. Zhang, X. Wang, Y. Wang, and H. Chen, "Human activity recognition across scenes and categories based on csi," *IEEE Trans. Mobile Comput.*, vol. 21, no. 7, pp. 2411–2420, Dec. 2020.

[34] Y. Zhang, Y. Chen, Y. Wang, Q. Liu, and A. Cheng, "Csi-based human activity recognition with graph few-shot learning," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4139–4151, Aug. 2022.

[35] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proc. Springer ECCV*, Sep. 2018, pp. 3–19.

[36] J. Yang, H. Zou, Y. Zhou, and L. Xie, "Learning gestures from wifi: A siamese recurrent convolutional architecture," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10 763–10 772, Sep. 2019.

[37] G. Koch, R. Zemel, R. Salakhutdinov *et al.*, "Siamese neural networks for one-shot image recognition," in *Proc. Lille ICML deep learning workshop*, vol. 2, Jul. 2015, pp. 1–8.

[38] A. Gretton, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, K. Fukumizu, and B. K. Sriperumbudur, "Optimal kernel choice for large-scale two-sample tests," in *Proc. Citeseer NIPS*, Dec. 2012, pp. 1205–1213.

[39] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *ACM SIGCOMM CCR*, vol. 41, no. 1, pp. 53–53, Jan. 2011.

[40] Y. Xie, Z. Li, and M. Li, "Precise power delay profiling with commodity wifi," in *Proc. 21st ACM AICMCNe*, ser. MobiCom '15, Jul. 2015, p. 53–64.

[41] O. Oshiga, H. U. Suleiman, S. Thomas, P. Nzerem, L. Farouk, and S. Adeshina, "Human detection for crowd count estimation using csi of wifi signals," in *Proc. IEEE ICECCO*, Dec. 2019, pp. 1–6.

[42] J. Lei Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*. [Online]. Available: http://arxiv.org/abs/1607.06450

[43] G. Hinton, O. Vinyals, J. Dean *et al.*, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*. [Online]. Available: https://arxiv.org/abs/1503.02531

[44] T. Furlanello, Z. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born again neural networks," in *Proc. PMLR ICML*, Jul. 2018, pp. 1607–1616.

[45] Z. Gao, J. Xue, J. Zhang, and W. Xiao, "Ml-wigr: a meta-learning-based approach for cross-domain device-free gesture recognition," *Soft Computing*, vol. 26, no. 13, pp. 6145–6155, May. 2022.