ADAWIFI, Collaborative WiFi Sensing for Cross-Environment Adaptation

Naiyu Zheng, Yuanchun Li, *Member, IEEE*, Shiqi Jiang, Yuanzhe Li, Rongchun Yao, Chuchu Dong, Ting Chen, Yubo Yang, Zhimeng Yin, Yunxin Liu, *Member, IEEE,*

***Abstract*—Deep learning (DL) based Wi-Fi sensing has witnessed great development in recent years. Although decent results have been achieved in certain scenarios, Wi-Fi based activity recognition is still difficult to deploy in real smart homes due to the limited cross-environment adaptability, *i.e.* a well-trained Wi-Fi sensing neural network in one environment is hard to adapt to other environments. To address this challenge, we propose ADAWIFI, a DL-based Wi-Fi sensing framework that allows multiple Internet-of-Things (IoT) devices to collaborate and adapt to various environments effectively. The key innovation of ADAWIFI includes a collective sensing model architecture that utilizes complementary information between distinct devices and avoids the biased perception of individual sensors and an accompanying model adaptation technique that can transfer the sensing model to new environments with limited data. We evaluate our system on a public dataset and a custom dataset collected from three complex sensing environments. The results demonstrate that ADAWIFI is able to achieve significantly better sensing adaptation effectiveness (*e.g.* 30% higher accuracy with one-shot adaptation) as compared with state-of-the-art baselines.**

***Index Terms*—Wi-Fi sensing, Deep learning, Domain adaptation, IoT devices, Smart home.**

I. INTRODUCTION

RECENT years have witnessed the rapid development of wireless communication technology and large-scale adoption of Wi-Fi based Internet-of-Things (IoT) devices. The digital signals transmitted and received by these devices could be used to sense our behavior, which enables many useful wireless sensing applications such as gesture recognition [1]–[4], sleep monitoring [5]–[7], fall detection [8]–[10], indoor localization [11]–[14], etc. Compared with traditional sensing methods, *i.e.* camera and inertial measurement unit (IMU), Wi-Fi sensing is contactless and alleviates people's concerns about privacy leakage [15], [16]. Moreover, due to the low cost of Wi-Fi chips and the popularity of Wi-Fi devices [17], [18], it is a very economic solution to achieve ubiquitous perception. In smart home scenarios, appliance companies expect their products is able to follow the user's gestures and actions to respond accordingly, *i.e.* when users want to turn on the television, they just need to wave at it. Therefore, gesture recognition is regarded as an important issue in both academia and industry among various wireless sensing applications.

Recent advancements in artificial intelligence (AI) and deep learning (DL) have simplified the development of sensing applications through model training with wireless signals [19]–[22]. The deep learning-based method offers a crucial benefit of automatically identifying high-dimensional patterns from data. Prior to the widespread adoption of deep learning, traditional approaches focused on complex mathematical transformations to obtain clear features from raw signals. As sensing tasks become more diverse, deep learning-based sensing technology will increasingly play a vital role in smart home scenarios.

Wireless sensing technology shows great potential but faces challenges in practical deployment. Wi-Fi signals are impacted by reflection, diffraction, and scattering during transmission, conveying information about both the target object and the environment. Even slight disturbances, such as movement of surrounding objects, can significantly alter the received signal. Especially when the same activity is performed in different environments, the signal will be distorted severely due to the changes in signal propagation paths, which is called 'domain shift' problem [23]–[25]. The key challenge in Wi-Fi sensing deployment is achieving cross-environment adaptability, where a model developed in known environments can efficiently and effectively adapt to unseen environments.

Pioneering works have made efforts to mitigate the domain shift problem through various methods. Some propose utilizing the characteristics of the signal itself to obtain domain-independent features through mathematical calculations [26], [27]. However, extracting such features necessitates substantial expert knowledge and prior understanding of deployment details. Recent techniques focus on designing deep learning models, particularly using adversarial learning to eliminate environment and subject-specific information [28]–[30]. However, collecting a large amount of unlabeled data for this approach is time-consuming. Another method is meta-learning [23], [24], [31], [32], leveraging abundant samples and task experiences from the source domain to adapt to the target domain more rapidly. However, this approach relies on the availability of a sufficient number of meta-training tasks with adequate task variability, which is not always the case. Furthermore, meta-learning incurs computational costs

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 62272261 and Midea Group Co., Ltd under a collaborative research project (Corresponding author: Yuanchun Li).

Naiyu Zheng and Zhimeng Yin are with the Department of Computer Science, City University of Hong Kong, Hong Kong, China (email: naiyzheng2-c@my.cityu.edu.hk; zhimeyin@cityu.edu.hk).

Yuanchun Li, Yuanzhe Li, and Yunxin Liu are with Institute for AI Industry Research (AIR), Tsinghua University, Beijing, China (email: liyuanchun, liyuanzhe, liuyunxin@air.tsinghua.edu.cn).

Shiqi Jiang is with Microsoft Research Asia, Shanghai, China (email: shijiang@microsoft.com).

Rongchun Yao is with Tencent Technology Co.,Ltd, ShenZhen, China (email: rongchunyao@smail.nju.edu.cn).

Chuchu Dong, Ting Chen, and Yubo Yang are with Midea Group Co.,Ltd, Foshan, China (email: dongcc7, ting1.chen, yangyb33@midea.com).

during the pretraining phase compared to standard supervised training. Therefore, effectively adapting to limited labeled data and complex environments remains an ongoing challenge that requires further improvement.

To overcome aforementioned practical drawbacks and limitations, there are three challenges to motivate us to design our system. The first challenge is **deployment heterogeneity**, which refers to the fact that the number, location, capabilities, and surroundings of IoT devices deployed in different environments may differ from each other significantly. For example, the deployment of furniture and the number of connected devices may be very personalized, making it difficult to design one model that is able to effectively adapt to all environments. The second challenge is **label scarcity**, which refers to the fact that it could be difficult or even impossible to obtain large amounts of labeled training data for deep sensing models. Collecting and labeling data for Wi-Fi signals is time-consuming and expensive, especially in the target environments (*e.g*. end-users' homes). Since deep sensing models are usually data-hungry, it can be difficult to train/fine-tune the sensing models in new environments with limited data. The third challenge is the **influence of low-quality signals**. Unlike most in-lab experiments where all transceivers have good sensing signal quality, real environments are very likely to contain devices with low-quality sensing data. This can be caused by a variety of factors, such as poor hardware conditions and the distance and obstacles between devices and sensing targets. Such low-quality wireless signals may have a negative impact on the deep sensing model during cross-environment adaptation if they are treated equally with other good devices.

To this end, we introduce ADAWIFI, a learning-based collaborative Wi-Fi sensing system that can be easily and effectively adapted to unknown environments. Collaborative sensing is defined as the utilization of multiple devices to produce integrated observational results and avoid biased perception of individual sensors for a given action. In this paper, we take gesture recognition as an example which serves as a fundamental facilitator for a diverse array of applications, including but not limited to smart home systems, security surveillance technologies, and virtual reality environments. Our framework is based on a deployment-independent neural network architecture that can aggregate high-level information from multiple sensors and an accompanying model adaptation method that can robustly adapt the sensing model to unknown environments with few labels.

Specifically, the sensing model includes encoder, aggregator, and classifier modules. Each encoder transforms sensor readings into a fixed-length vector embedding, which are merged by the aggregator and fed to the classifier. This architecture permits sensor addition or removal without retraining the model. We propose an adaptation method tailored to the model, using a virtual environment to augment limited labeled data. This method creates synthetic intermediate samples to progressively transfer from the source environment to the target environment. We further propose estimating signal quality by analyzing their contribution to collective predictions. A two-stage method is adopted to achieve this, first finding optimal sensor weights for highest accuracy, then tuning other parameters with augmented data while fixing sensor weights.

We evaluate our approach on both a custom dataset collected from home-like environments and a large public dataset, Widar3.0 [26]. The results show that our framework can achieve more than 80% adaptation accuracy in unknown environments with very few labeled samples, *e.g*. 1-10 shots, significantly outperforming the state-of-the-art baselines [28], [29], [31]–[33]. Our system is robust against low-quality sensors in the target domain, where the accuracy drop caused by low-quality sensors is less than 5%. We also demonstrate the high flexibility and low overhead of our system based on an implementation with Raspberry Pi.

In summary, we make the following technical contributions in this paper:

1) We introduce a generic neural network architecture for sensing user activities with multiple Wi-Fi devices, which can be applied to different homes with heterogeneous device deployments.
2) We design a cross-environment sensing system adaptation method based on the collective sensing model. The adaptation requires very few labeled samples in the target environment and can effectively resist low-quality sensors.
3) We demonstrate the good performance of our approach through experiments on both a custom dataset and an open dataset. The source code and the dataset will be released.

In the following of the paper, Section II introduces related work of Wi-Fi sensing. Section III demonstrates our motivational study and the detailed design of ADAWIFI is described in Section IV. The experimental results of the evaluation are presented in Section V. Discussion and conclusion are reported in Section VI and VII respectively.

II. RELATED WORK

In this section, We first introduce background of Wi-Fi sensing technology and then select representative domain adaptation methods previously used to address cross-environment problem to better understand the limitations of existing work.

A. Wi-Fi based Sensing

Using Wi-Fi signals for sensing has been an attractive and popular research direction in the mobile community due to various benefits including ubiquitous deployment, low cost, and privacy friendliness [20], [21], [34].

Received Signal Strength Indicator (RSSI) or Channel State Information (CSI) are commonly extracted from Wi-Fi devices. While RSSI had been initially preferred for its high accuracy in indoor localization [35] and simpler application in human activity recognition [36], CSI is more widely used due to its capacity for finer sensing granularity, greater deployment cost-effectiveness, and better resistance to multipath interference [19]. Furthermore, processed amplitude information derived from CSI has been applied to successfully extract human activities such as walking and running [37], [38], although the presence of noise in CSI signals can significantly affect the accuracy of the recognition results. To address this limitation, researchers have developed more comprehensive methods, such as CARM [39], which quantitatively correlates CSI value dynamics with human movement speeds. Other approaches, such as Fresnel zone model [40], have theoretically demonstrated the feasibility of decimeter-scale activity recognition. Moreover, the Doppler Frequency Shift (DFS) technique, transforms CSI data from time-domain to frequency-domain, providing access to a clearer analysis of frequency composition for comprehensive sensing of complex tasks [41].

B. Cross-environment Sensing Adaptation

Cross-environment sensing adaptation refers to the ability of a sensing system to perform effectively across different environments. The need for cross-environment sensing arises due to the inherent variations and differences of signals that exist between different environments. Prior approaches have attempted to address the challenges of cross-environment sensing adaptation from different perspectives [25], [42]–[44], mainly focusing on extracting environment-independent features from sensing data or adapting the pretrained sensing model to new environments with little effort [45]–[48].

For the viewpoint of extracting environment-independent features, it holds that signals in both the original and target domains have inherent characteristics. In other words, if the noise interference in different environments can be eliminated, environment-independent features can be extracted. Some works model the physical characteristics of signals and obtain a new feature through complex mathematical calculations and expert knowledge in the field of communication. Widar3.0 [26] proposes body-velocity profile (BVP), an advanced feature extracted from CSI that is domain-independent. However, computing BVP requires much prior knowledge, including the user's location, orientation, etc., which restricts its usage scenarios. Adversarial learning is utilized to learn an environment-independent feature encoder [28], [29] in recent years. However, getting a unified representation across diverse environments requires abundant data (*e.g.* hundreds of unlabeled or well-segmented samples) from the target environment, which is not practical in real-world scenarios.

For the perspective of adapting the pretrained sensing model to new environments with little effort, it proposes the design of improved algorithms to achieve adaptation with limited data based on the pretrained model in the source domain. Among them, few-shot learning mainly follows the paradigm of meta-learning [49] which teaches DL models how to learn from hundreds of sampled tasks from source domain and generalize to new environments. For instance, RFNet [31] and ProtoNet [32] introduce a metric-based meta-learning framework that learns a metric function from source environments which can be used to assemble the suitable sensing model in the target domain. MetaSense [33] applies meta-learning based on the MAML algorithm [50] to update the deep sensing model for new conditions. The authors in [23], [24] make improvements through data augmentation, designing better loss functions and model architectures, based on classic meta-learning models. However, meta-learning usually requires a number of domains of training data to be effective. Meanwhile, meta-learning algorithms are very unstable during training [51]. Current studies in the ML area also find that meta-learning may be ineffective in realistic scenarios [52], [53] when solving cross-domain tasks, and even be inferior to simple fine-tuning, which limits the practicality of such approaches in a real deployment.

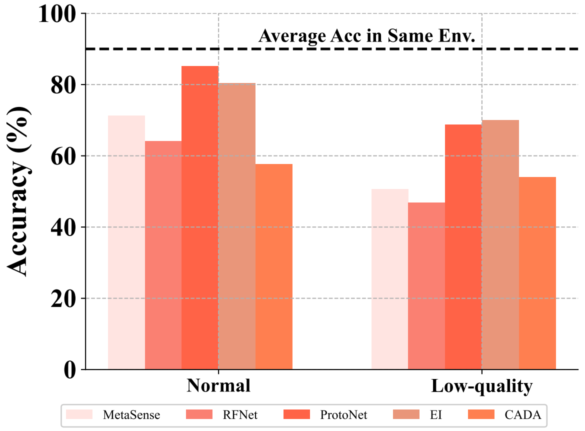


Fig. 1: Performance degradation caused by cross-environment and low-quality signals for prior sensing adaptation Work.

In general, due to the fundamental limitations arising from domain dependencies of signals and the inherent drawbacks of the model itself, the utilization of the aforementioned methods remains limited for achieving generalization across different domains.

III. Motivation

To motivate the design of AdaWiFi, we implement the motivational study in this section. We conduct a series of pilot experiments to analyze the difficulties of applying learning-based Wi-Fi sensing across environments. The experiments are conducted on a well-known public Wi-Fi sensing dataset, Widar3.0 [26]. It contains Wi-Fi DFS profiles corresponding to around 250K gesture samples, collected from 3 indoor environments with 6 Wi-Fi links in a 2m×2m sensing area.

Cross-Environment Performance Degradation. We examine how the sensing model trained in one environment would perform in another environment. Specifically, we select two environments and 6 types of gestures.

We first implement models of mentioned prior work in 1-shot setting (one sample for each gesture to adapt) for target environment, as shown in Fig 1. Their average test accuracy in the same environment can achieve 90%. However, whether in 'normal' or 'low-quality' settings, there is a obvious decrease in accuracy across environments. This indicates that even if advanced adaptive methods are used, they still face performance degradation issues when deploying across environments. Note that the data in Widar3.0 are collected in relatively clean areas, and the environments we select have similar device deployments. It suggests that the accuracy gap between the source and target environments might be even larger in real complicated deployments.

Opportunity via Multiple Devices. We train a vanilla CNN (four convolutional layers) model with different numbers of Wi-Fi links using the DFS profile samples of Widar from the first environment to classify gestures and test the trained model on both environments. The results are shown in Table I. We observe the benefit of adding more Wi-Fi links for sensing in Table I. Specifically, by using three pairs of devices, the gesture classification accuracy in the same-environment setting is increased to 91.67%, much higher than using only one device pair with an accuracy of 70.00%. Although the classification accuracy drops by about 30% on average in cross-environment settings using different numbers of links, the performance is still better when using more devices. Such a benefit is worth exploiting because today's smart homes usually have many IoT devices connected with Wi-Fi, which naturally creates the opportunity of using multiple device pairs for sensing.

Impacts of Low-quality Signals. We further investigate the impact of low-quality signals on previous adaptive methods in Fig 1. To establish a basis of comparison with normal situations, we set up two scenarios: 'Normal', which represents the original Widar 3.0 data in the target environment, and 'Low-quality', which represents the introduction of an artificial noise sensor to simulate a low-quality signal. The value of this sensor is sampled from a standard Gaussian distribution, as discussed in detail in Section V-D. Notably, the performance of all these methods experienced varying degrees of degradation under the 'Low-quality' setting, indicating that the existence of low-quality signals is overlooked in previous work.

These methods demonstrate promising results in their respective scenarios but fail to adequately address deployment heterogeneity, low-quality signals and label scarcity. Specifically, these methods require consistent data input formats in both the source and target domains with the number of sensors being fixed, and they have relatively static sensor placement. Furthermore, they assume ideal deployment environments free of interference. However, in practical scenarios, the data obtained may not reflect these assumptions, even with Widar3.0 multi-sensor deployment solution. Although meta-learning, representing small sample learning, partially mitigates the challenge of label scarcity, it struggles to address significant differences in sensing data between domains and lacks smooth transitions in different environments. These results and discussions motivate us to design a better technique for cross-environment adaptation.

TABLE I: The classification accuracy of the sensing model trained with different numbers of Wi-Fi links.

Test On	One Link	Two Links	Three Links
Same Env.	70.00%	86.65%	91.67%
Cross Env.	45.95%	50.92%	54.30%

Required Labeling Efforts to Recover Accuracy. We next analyze the amount of labeled training data necessary for adapting a pre-trained sensing model to a new environment using conventional transfer learning. Different numbers of labeled samples were used to fine-tune the pre-trained model in the source environment, and the accuracy was tested in the target environment, as shown in Fig II. Although the transfer learning technique effectively improved performance, achieving comparable accuracy in the source environment requires a large number of labeled samples. For instance, using three links, over 600 samples are needed to achieve 90% accuracy. Given that labeling a considerable number of samples is time-consuming, taking a few hours (roughly 10 seconds per sample), it is inappropriate to require the end-user to undertake such a labelling process.

TABLE II: Required labeling efforts to recover accuracy in cross-environment sensing adaptation.

Test On	0	200	400	600
One Link	46.33%	63.95%	69.88%	77.13%
Two Links	49.46%	66.71%	79.22%	83.33%
Three Links	54.00%	70.58%	86.56%	90.13%

IV. Our Approach: AdaWiFi

Based on the discussions presented in Section III, we hereby propose our solution, named AdaWiFi, which utilizes the DFS information extracted from multiple CSI streams as input and automatically learns how these streams can collaborate with each other to adapt to diverse environments. AdaWiFi comprises three novel designs: deployment-independent collective sensing architecture, progressive tuning with virtual domains, and contribution-aware sensor reweighting. We first describe the overview of AdaWiFi and then introduce our three novel designs in sequence.

A. Overview

The workflow of our approach is shown in Fig 2. We assume the sensing model is originally trained in one or more source environments (*e.g.* the developers' laboratories), where the developers have strong motivation and sufficient time to collect a large dataset of labeled samples. The trained model is then distributed to different target environments (*e.g.* the end-users' homes) to adapt. Unlike the developers in the source environments, the end-users are usually less patient to carefully collect and label the sensing samples. Therefore, we can only assume there are few labeled samples (*e.g.* 1-10 samples per class) in the target environments, which can probably be obtained by asking the users to do a short demonstration.

At training stage in source environment, we introduce a new sensing model technique that is tailored for adaptability, named collective sensing network architecture, to encode and aggregate Wi-Fi links flexibly. Afterwards, the pre-trained model and training data in the source domain are sent to the target environment for adaptation. At this stage, we implement progressive model tuning technique through constructing virtual domain to mitigate the huge data distribution gap between environments. Meanwhile, contribution-aware sensor reweighting technique is responsible for analysing sensor quality to mitigate the negative influence of low-quality sensors.

As for preprocessing of wireless signals, we follow the standard procedures in prior literature to extract the DFS

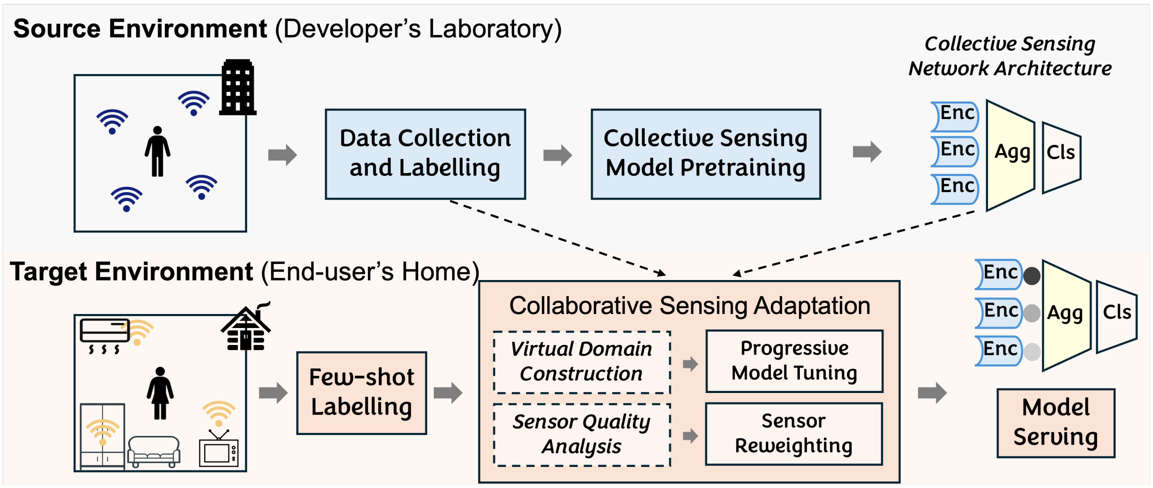

Source Environment (Developer's Laboratory)
Data Collection and Labelling
Collective Sensing Model Pretraining
Collective Sensing Network Architecture
Enc
Enc
Enc
Agg
Cls
Target Environment (End-user's Home)
Few-shot Labelling
Collaborative Sensing Adaptation
Virtual Domain Construction
Progressive Model Tuning
Sensor Quality Analysis
Sensor Reweighting
Enc
Enc
Enc
Agg
Cls
Model Serving

Fig. 2: Workflow of our AdaWiFi.

[37], [41], [54] information from raw Wi-Fi signals. We will introduce the main components in more detail in the following subsections.

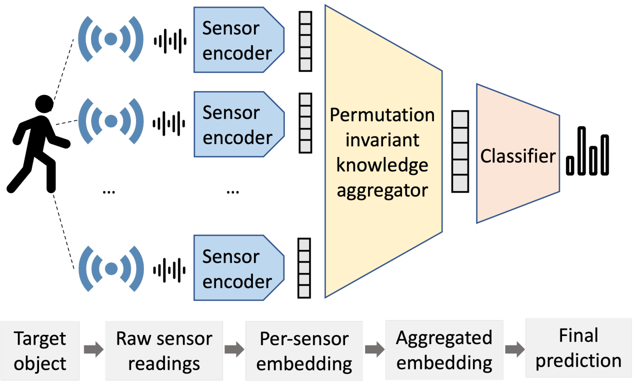


Fig. 3: The neural network architecture for collective sensing.

B. Deployment-independent Collective Sensing Architecture

Most deep learning model architectures used in Wi-Fi sensing are not designed for cross-environment scenarios due to the fixed input format and order. For example, the input of a typical CNN (Convolutional Neural Network) model should adhere to a fixed-size shape, where the number of devices and their order are predefined. A RNN (Recurrent Neural Network) model processes inputs one token at a time in a sequential manner, and they can struggle with permutation since they rely heavily on the specific order in which tokens are presented. However, in cross-environment settings, we expect the model to flexibly accept a variable number of signal streams with arbitrary order.

Therefore, we design a generic neural network architecture for collective sensing with multiple sensors, as shown in Fig 3. In the network architecture, the DFS information of each transmitter-receiver link is first processed by a sensor encoder module to generate a per-sensor embedding. The sensor encoder module uses the recurrent neural network [55], [56] (specifically, the long short-term memory, LSTM) to convert the time-series sensor data to a fixed-length vector at each time step. The per-sensor embedding is expected to represent the activity of the sensing target depicted by each sensor.

Then the per-sensor embeddings are fed into a knowledge aggregation module to generate a global embedding vector, which represents the activity of the target object depicted by all sensors collectively. We use the Transformer model [57] as the knowledge aggregator since the self-attention mechanism of Transformer naturally has the ability to exchange knowledge between different input tokens, while in our case, each token of the Transformer is a per-sensor embedding. Compared to the previously mentioned CNNs and RNNs, the self-attention mechanism of the Transformer allows the model to process all sensor embeddings in parallel, rather than sequentially, and understand the context of each embedding in relation to all other embeddings, making it robust to permutations. However, the conventional Transformer architecture is designed for sequence data (*e.g.* natural language text), in which the order of tokens is important. Specifically, it uses positional encodings to inject information about the position of each token in the sequence. Instead, the sensors in an environment are unordered, and the sensing model should be permutation-invariant, *i.e.* changing the order of sensors should not change the model prediction. To achieve this, we exclude the positional encodings in the Transformer model, so that the input tokens of the knowledge aggregator only contain the sensing signal information. Such a permutation-invariant architecture allows flexibly adding or removing sensors in the collective sensing system. Such nice properties make it natural to use the model across different sensing environments.

Finally, the aggregated embedding is fed into a Linear classifier to produce the final prediction.

More formally, suppose our model is denoted as f, which produces a prediction y based on the input sensor data x. The input sensor data $x = \{x^1, x^2, ..., x^m\}$ is a combination of the DFS information of multiple transmitter-receiver links. The shape of x can be described as $m \times s \times t$, which represent the number of Wi-Fi links, DFS feature size, and the duration of activity respectively. Our model can be represented as $f = f_{cls} \circ f_{agg} \circ f_{enc}$, where f_{enc}, f_{agg}, and f_{cls} are the per-sensor encoder module, the aggregator module, and the classifier respectively. $\theta = (\theta_{enc}, \theta_{agg}, \theta_{cls})$ is the set of model parameters for each module. We elaborate on the alterations in input structure within the system. Following the per-sensor encoding stage, the data shape undergoes transformation into an $m \times h$ matrix, where h represents the number of hidden features within the LSTM. Subsequently, the embedded data is forwarded to the aggregator, where it is reshaped into a one-dimensional matrix with a dimension of d, denoting the dimensionality of the feedforward Transformer model. Finally, this embedding is forwarded to the classifier for the purpose of classification. For a given sample $x_i = \{x_i^1, x_i^2, ..., x_i^m\}$, $enc_i^k = f_{enc}(x_i^k, \theta_{enc})$ is the encoding of the k-th sensor. $agg_i = f_{agg}((enc_i^1, enc_i^2, ..., enc_i^m), \theta_{agg})$ is the aggregated embedding. $\hat{y_i} = f_{cls}(agg_i, \theta_{cls})$ is the predicted probablity of target classes.

Training the sensing model in an environment is the same as training normal deep learning models, *i.e.* using gradient descent to find the parameters $\hat{\theta}$ that can minimize the following classification loss:

$$J_{pred}(\theta) = \frac{1}{n} \sum_{i=1}^{n} L_{cross-entropy}(y_i, f(\theta, x_i)) \qquad (1)$$

where $L_{cross-entropy}$ is the cross entropy loss and y_i is the label for the input x_i.

Flexible Distributed Computing for Collective Sensing. Our proposed architecture for collective sensing allows a flexible distribution of the computational workload among different IoT devices. The system consists of multiple client devices that capture the wireless signals and a master device (which can be either one of the clients or a separated device) that performs the sensing prediction. The sensing process comprises three stages: data preprocessing, which transforms the raw sensor signals into DFS; encoding, which generates the embedding for each sensor; and aggregation, which combines the embeddings to produce a prediction. The first two stages do not require any coordination among different devices and can be executed either on each client device or on the master device. Depending on the computational capabilities of the devices and the network conditions between them, we can flexibly determine how to allocate the computational workload to achieve optimal efficiency. For instance, if there is a powerful master device and a fast network connection between the master and clients, we can run all three stages on the master device; whereas if the client devices are powerful enough, they can perform most of the processing and encoding tasks locally to reduce the burden on the master.

C. Progressive Tuning with Virtual Domains

Although the collective sensing architecture solves the problem of deployment heterogeneity between different environments, directly transferring the model trained in the source environment to the target environment is still difficult due to huge data distribution differences.

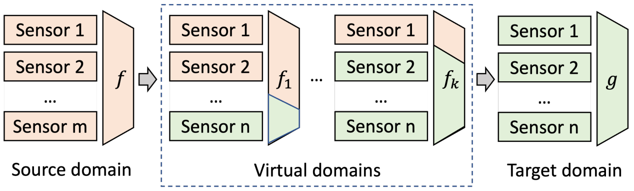


Fig. 4: Progressive adaptation by constructing virtual intermediate domains.

To fix the large gap between the source environment and the target environment, we introduce a progressive adaptation scheme. The method is two-fold, including a data augmentation technique to construct a set of intermediate virtual domains between the source and target environments, as illustrated in Fig 4, and training techniques to effectively adapt the sensing model based on the constructed virtual domains.

Virtual Domain Construction. Since our sensing system is based on multiple sensors (*i.e.* multiple transmitter-receiver links), it creates the unique opportunity to produce various intermediate domains by mixing the sensors from different environments. Specifically, a new virtual domain can be constructed by including k sensors in the source environment and l sensors in the target environment, and each sample $\{x_v, y_v\}$ in the virtual domain can be generated by combining a sample from the source environment ($x_s = \{x_s^1, x_s^2, ..., x_s^k\}$) and a sample from the target environment ($x_t = \{x_t^1, x_t^2, ..., x_t^l\}$) by concatenating the sensor readings to $x_v = \{x_s^{i_1}, x_s^{i_2}, ..., x_s^{i_m}, x_t^{j_1}, x_t^{j_2}, ..., x_t^{j_n}\}$, where $1 \leq m \leq k$, $1 \leq n \leq l$, and the x_s and x_t have the same label (*i.e.* $y_s = y_t = y_v$). Therefore, the difficult task of directly adapting the model between distributionally-different source and target environments can be converted to a series of simpler tasks of adapting the model between the distributionally-similar intermediate domains, which are much easier to achieve with limited data. Although the generated samples are not realistic in the physical world, they create intermediate steps between the source and target domains and enable smooth transferability between them.

We also incorporate sample-level signal processing techniques to further augment the data in each domain. Based on the insight that the activity recognition should remain consistent in a small duration of time or with slightly-different movement speed, we propose to generate meaningful sensing signals from existing signals with time-domain transformations. In general, we change the positions of the sampling points of each activity segment to obtain more samples. Specifically, an activity sample is collected at 1000Hz for about 3 seconds, and we can get a sequence of sensor signals $\{x_t, x_{t-d}, ..., x_{t-Nd}\}$ through fixed interval sampling, where t is the end time of the activity and d is the interval between successive sensor readings. Since the starting position and interval of sampling do not directly affect the overall features of the activity, We can further obtain more samples for the same activity as $\{x_{t'}, x_{t'-d'}, ..., x_{t'-Nd'}\}$, where $t' = t + \Delta t$ and $d' = \sigma d$ are slightly and randomly shifted end time and scaled sampling duration. In this way, the same activity segment can be used to

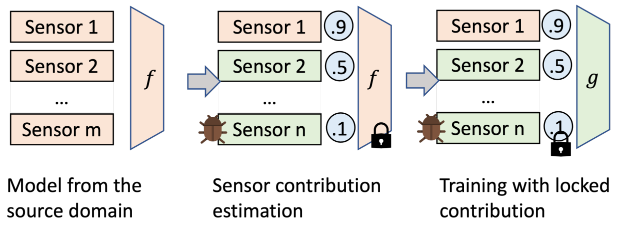


Fig. 5: Mitigating the negative influence of low-quality sensors through sensor contribution estimation.

generate multiple samples which greatly increases the diversity of data.

Cross-Domain Embedding Alignment. To fully utilize the virtual domains constructed above in sensing adaptation, we add an additional loss except for the normal loss described in Equation 1. Since a key objective of our cross-environment sensing system is to make the representation of the same activity consistent across different environments, we further define a cross-environment embedding alignment loss:

$$J_{consist}(\theta) = \frac{1}{n} \sum_{\substack{i=1 \\ x,x' \in E_1, E_2}}^{n} L_{mse}(f_{agg}(f_{enc}(\theta, x)), f_{agg}(f_{enc}(\theta, x'))) \tag{2}$$

where L_{mse} is the mean squared error, and x and x' are two samples belong to two different environments E_1 and E_2.

By minimizing the $J_{consist}$ loss together with the classification loss, we can train the model to generate consistent representations of the target objects across different environments. Therefore, the knowledge in the source environment can be transferred to the target environment. The concept of generating consistent representation between environments can be found in domain adaptation and domain generalization approaches, which are typically achieved by finding a domain-independent representation with unlabeled target-domain data or without target-domain data (*e.g.* using adversarial training or meta-learning techniques). While in our scenario, since we assume a few labeled target-domain samples are available, we can directly take a training approach with the domain consistency loss which can be more sample-efficient.

D. Contribution-aware Sensor Reweighting

Adapting a deep sensing model to new environments also faces the problem of low-quality sensing signals. Since the devices and networks may not be perfectly deployed and configured in the end-users' homes, the sensing signals of some devices may be noisy, which increases the difficulty of model adaptation. To mitigate the negative influence of low-quality sensors, we adopt a two-stage adaptation scheme, which reweights the contribution of each sensor before using the sensors for training in the target domain, as shown in Fig 5.

The use of multiple sensors makes it possible to analyze the quality of participating sensors. The quality of each sensor can be estimated based on its contribution to global prediction. Specifically, we assign a learnable weight for each sensor in our collective sensing architecture, which is applied to the per-sensor embedding enc_i^k before feeding the embeddings to the knowledge aggregator f_{agg}. We train the weights by analyzing how the sensing encoding of each transmitter-receiver link aligns with other links with the same class label. A lower weight will reduce the impact of the corresponding sensor on the aggregated representation agg_i and the final prediction y_i. To learn the weights, we fix the other parameters in the sensing model and minimize the prediction losses J_{pred} and $J_{consist}$ through backpropagation, which will automatically find an optimal weight assignment. Each weight would represent the contribution of each sensor in the collective system, and low weights would be assigned to the low-quality sensors. The weights are used during sensing encoding aggregation to ensure that the low-quality signals have little influence on the generated global embedding.

After learning the contribution of each sensor, we fix these weights and train the other parameters for further adaptation with our progressive tuning technique. It is worth noting that the fixed sensor weights can also be periodically updated in the target environment to handle dynamic changing signal qualities. To solve the cross-environment problem, AdaWiFi do not only adapt the model weights, but also the sensor weight. Therefore, with the negative influence of low-quality sensors significantly reduced, AdaWiFi is able to more effectively adapt to new environments with the limited labeled samples.

E. Overall Training Process

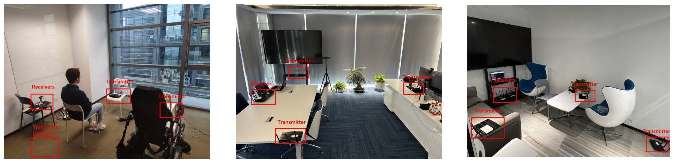
The training process is divided into two primary stages. In the initial stage, a base model of the AdaWiFi collective sensing network architecture is trained using data from the source environment to facilitate subsequent adaptation. During the second phase, the base model is adapted to the target environment using AdaWiFi adaptation technology. In particular, a learnable weight is assigned to each sensor prior to adaptation. Subsequently, all learnable parameters, except for the weight parameters, are frozen for several epochs. This step is conducted to analyze the contribution of each sensor and ensure that the weights of low-quality signals are reduced. In the remaining epochs, the learnable weights of the sensors are frozen, while other parameters (*e.g.* the LSTM-based sensor encoder, Transformer-based knowledge aggregator, and linear classifier) are enabled. During the whole process in the second stage no matter which parameters are frozen, the progressive tuning with virtual domains is enabled to achieve smooth adaptation.

V. Evaluation

In this section, we evaluate AdaWiFi to understand its sensing adaptation effectiveness, benefits of multi-sensor collaboration, robustness against low-quality sensors, system overhead, and the contribution of each component.

A. Experiment Setup

Hardware & Software. We used the TP-LINK WDR4310 routers equipped with Atheros AR9344 SoC as the receivers and transmitters in our experiments. To extract CSI, we use Atheros CSI tool [58]. The devices communicate at 5GHz. We



(a) Room 1. (b) Room 2. (c) Room 3.

Fig. 6: The three environments used in our experiments to collect the sensing data.

built the neural network and learning algorithm of AdaWiFi with PyTorch and used a desktop with one NVIDIA 3090 GPU to train the model. We used Raspberry Pi to evaluate the system overhead.

Model Settings. We described the implementation details of AdaWiFi. The LSTM-based sensor encoder incorporates a hidden state with 128 features and consists of a single recurrent layer. In the Transformer-based knowledge aggregator, we configure 4 attention heads, 1 encoder layer, and a feedforward network model with a dimensionality of 128. Training and adaptation procedures are executed over 600 and 200 epochs, respectively. Notably, parameters other than the learnable weights of sensors are frozen for a total of 60 epochs. Model parameter updates are performed using the Adam optimizer with a learning rate of 0.001.

Datasets. We evaluated our system on two datasets, including Widar3.0 [26], a public Wi-Fi sensing dataset with 258,575 samples that were collected in several clean and strictly controlled environments, and a self-collected dataset that can better imitate in-the-wild sensing environments. Specifically, we selected three typical rooms with common furniture and appliances such as TVs, desktops, sofas, and air conditioners, as shown in Fig 6. The three pairs of Wi-Fi devices in each room are placed with the electrical appliances, and the distances and obstacles between the devices are naturally and irregularly distributed. Such settings introduce more signal noise and dynamics due to the multi-path effect, etc., in turn raise challenges for the Wi-Fi sensing system.

In each room, we asked five volunteers to perform six types of gestures, including swipe, clap, slide, push & pull, draw zero, and draw zigzag, similar to the gestures used in the Widar3.0 dataset. Each volunteer is asked to perform more than 200 gestures in random order in each room. The gestures are labeled with a mobile app immediately after each gesture is completed. In total, we collected 10 hours of data with more than 9,000 samples.

Baselines. We selected several state-of-the-art model adaptation approaches for deep sensing, including EI [28], CADA [29], MetaSense [33], RFNet [31], and ProtoNet [32]. EI and CADA adopt adversarial learning to learn a powerful discriminator and feature extractor. Through constantly confusing the target domain and the source domain, the feature extractor can obtain domain-independent features. MetaSense, RFNet, and ProtoNet use meta-learning, a few-shot learning technique that learns how to adapt to new tasks with limited data based on sufficient data of multiple known tasks. MetaSense uses an optimization-based meta-learning method based on the insight that optimizing the algorithm's parameters through customized optimization procedures can be more effective in addressing the small sample classification problem. In contrast, RFNet and ProtoNet are metric-based meta-learning methods that utilize a distance metric to measure the similarity between the samples in the support and query sets. For those baseline approaches that are not open-sourced (*e.g.* EI) or not designed for wireless sensing (*e.g.* MetaSense), we reimplemented the core algorithms of them for Wi-Fi sensing according to their papers. Although these methods are not designed for multiple devices to collaborate sensing, we add an aggregation layer to optimize the observation results of multiple sensors to achieve fair comparison with AdaWiFi.

B. Adaptation Effectiveness

Self-collected Dataset. We first evaluate the performance of our method on the self-collected dataset, which closely aligns with real scenarios. The results are shown in Table III. Overall, AdaWiFi outperforms the baselines with 20% higher accuracy on average. We test the performance of all models under 1, 10, and 20-shot cases in three rooms, where '1 -shot' means one labeled sample for each gesture to be collected in the target environment. Note that this concept differs from few-shot learning in meta-learning. AdaWiFi can quickly adapt to new environments even in 1-shot setting with above 70% accuracy, while most baselines have less than 40% accuracy. With increasing numbers of samples in the target domain, the performance of our model is further improved. For example, the gesture classification accuracy can even achieve as high as 88.50% in Room 2 after adapting with 20-shot samples.

Baseline approaches have lower performance with only around 40% accuracy in 1-shot setting. Although performance improves under 20-shot setting, the maximum accuracy is limited to around 70%. Adversarial-learning-based approaches, *i.e.* EI and CADA, show the highest improvement with increased adaptation samples, which can help to learn a domain-independent feature extractor by confusing the domain discriminator. But limited labeled samples in the target environment greatly affect their performance. MetaSense, originally for IMU-based sensing, has the lowest accuracy due to the instability of the MAML algorithm used in training on both datasets. Metric-based meta-learning approaches like RFNet and ProtoNet have slightly better performance, but require many domains of training data to be effective. The significant differences between environments and unbalanced signal quality worsen the adaptation ability of baseline methods developed with simple sensing environments.

The reason why AdaWiFi outperforms the baseline methods is twofold. First, it is hard for most baselines to extract enough domain-independent features in the target domain with limited data. Although they have powerful neural network backbones and novel adaptation techniques, the data distribution between the source and target domains is too large, and a bridge is needed to connect them. Due to the design of AdaWiFi, we can obtain numerous intermediate samples by generating virtual domains even there are only few shot samples in target domain. Through these intermediate

TABLE III: Sensing adaptation accuracy of different approaches on our self-collected datasets. Each cell represents the gesture classification accuracy achieved by the corresponding method in each room. The models are sufficiently trained in other rooms and adapted in the target room with N shots of samples (N=1,10,20).

Setting		Method					
		MetaSense	RFNet	ProtoNet	EI	CADA	AdaWiFi
Room1	1 shot	35.33%	40.63%	33.01%	31.07%	33.01%	**72.82%**
	10 shot	40.00%	44.38%	44.82%	50.49%	39.81%	**81.55%**
	20 shot	47.53%	49.38%	58.73%	60.19%	50.49%	**84.47%**
Room2	1 shot	34.01%	46.88%	37.12%	35.13%	32.74%	**72.57%**
	10 shot	40.00%	50.00%	55.10%	45.13%	50.44%	**84.96%**
	20 shot	42.68%	50.00%	60.73%	55.56%	57.52%	**88.50%**
Room3	1 shot	38.33%	50.00%	31.70%	43.22%	32.20%	**73.73%**
	10 shot	51.66%	53.13%	55.45%	61.02%	64.41%	**80.51%**
	20 shot	53.35%	62.50%	55.73%	73.73%	72.88%	**78.81%**
Average	1 shot	35.89%	45.84%	33.94%	36.47%	32.65%	**73.04%**
	10 shot	43.89%	49.17%	51.79%	52.21%	51.55%	**82.34%**
	20 shot	47.91%	53.96%	58.40%	63.16%	60.30%	**83.93%**

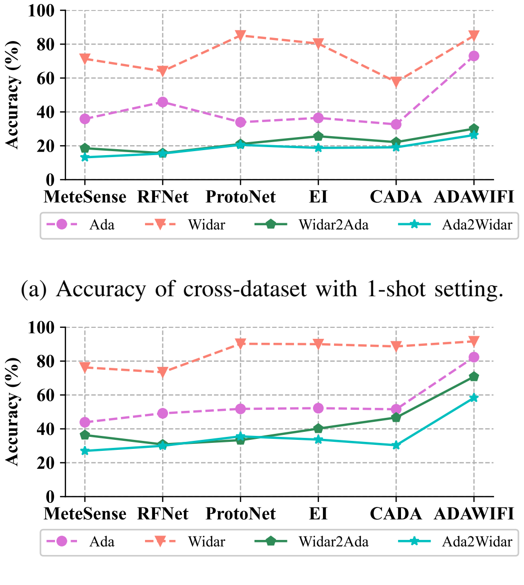
TABLE IV: Sensing adaptation accuracy of different methods on Widar3.0 datasets. The sensing models are sufficiently trained in one environment and adapted with few-shot samples in another environment.

Method	Setting	
	1 shot	10 shot
MetaSense	71.33%	76.26%
RFNet	64.06%	73.44%
ProtoNet	85.12%	90.22%
EI	80.33%	90.00%
CADA	57.67%	88.67%
AdaWiFi	85.00%	91.67%

samples, the model is able to adapt to new environment step by step, rather than one-step adaptation in baseline models. We figure step-by-step adaptation will gradually approach the data distribution of the target domain in the mixed domain data during the training process, and ultimately transform to the target domain. Effectiveness of AdaWiFi is more obviously when target samples are extremely scarce, like 1-shot setting. Second, the real sensing environment is more challenging as the sensing devices might be placed in corners or obstructed by huge obstacles, which seriously change the propagation path of signals and introduce noise to the entire system. In addition, interference between signals on the same channel may cause packet loss due to the complicated network environment [59]. AdaWiFi take these factors into consideration and adjust the weights of low-quality sensors accordingly to avoid interference. In previous work, their designs only focus on considering changes of room layout and ignore changes of device placement and introduction of low-quality signals. However, our proposed technique of contribution-aware sensor reweighting can handle both above challenges. Even though there are random devices are unable to transmit signal normally due to interference, AdaWiFi can automatically find them and minimize their impact on the entire system explicitly. For other baselines, only one low-quality sensor will poisoned the entire system. Therefore, our system can quickly and efficiently adapt to new environments with very limited samples.

Widar3.0 Dataset. We also examine whether our system can achieve better cross-environment adaptation effectiveness on the public dataset Widar3.0. The results are shown in Table IV. We can notice that AdaWiFi achieves very high accuracy in both 1-shot and 10-shot settings, 85.00% and 91.67% respectively. Meanwhile, the baselines including ProtoNet, EI, and CADA also get good performance as compared with the results on our self-collected dataset. AdaWiFi seems to be less competitive than other baseline models because the sensing environments of Widar3.0 are relatively clean and similar. Specifically, Widar3.0 has a similar relative position between devices and sensing area in every room, which makes cross-environmental challenges smaller than our dataset. Meanwhile, current baselines have considered these simple deployments and assumed no low-quality sensors, so that all can obtain decent results. This phenomenon demonstrates the importance of taking the noisy and heterogeneous characteristics of real sensing environments into consideration when designing cross-environment sensing systems.

Cross-dataset. To further investigate the adaptability of AdaWiFi, a comprehensive cross-dataset scenario is conducted. Specifically, the model is trained with Ada dataset (a proprietary dataset) and is subsequently tested with Widar3.0 dataset under both 1-shot and 10-shot conditions, and vice versa. Parallel experiments are also conducted with baseline models. This particular experiment poses heightened challenges due to the disparate origins of the datasets, collected by devices featuring distinct chip architectures (*e.g.* Widar3.0 dataset gathered via an Intel 5300 wireless NIC and Ada dataset via a TP-LINK WDR4310 equipped with an Atheros AR9344 SoC), and within entirely dissimilar environments. Notably, prior studies only explored cross-environment scenarios within the same dataset. As illustrated in Fig 7, the performance across varied conditions exhibits a notable decline, particularly under the 1-shot setting, where all models, except for AdaWiFi, achieve only approximately 20% accuracy. In contrast, AdaWiFi demonstrates accuracies of 30.09% and 26.33% under Widar-to-Ada and Ada-to-Widar3.0 setups, respectively. This discrepancy can be attributed to the substantial differences in data distributions between the two datasets. The efficacy of AdaWiFi becomes more pronounced under the 10-shot setting, where all models exhibit improved



(a) Accuracy of cross-dataset with 1-shot setting.

(b) Accuracy of cross-dataset with 10-shot setting.

Fig. 7: The cross-dataset adaptation performance of different settings of various models.

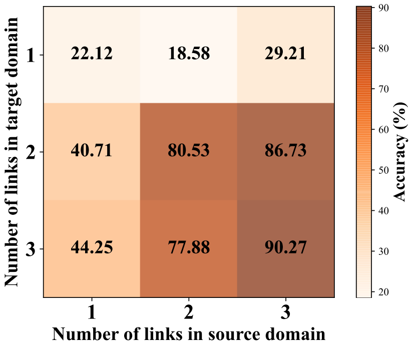


Fig. 8: The cross-environment adaptation accuracy achieved with different numbers of Wi-Fi links.

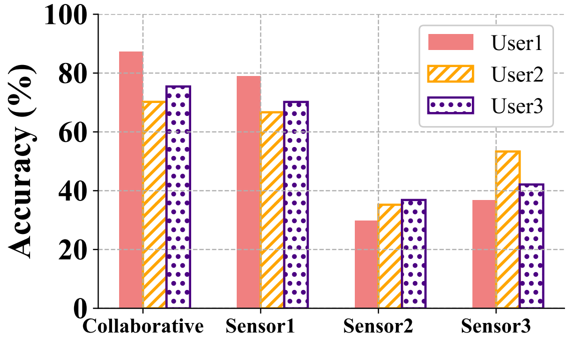


Fig. 9: The accuracy achieved with different individual devices and their combination.

accuracies. Notably, AdaWiFi outperforms its counterparts by a margin of at least 25%, showcasing its robust adaptability. An interesting observation is the generally superior performance of the Widar-to-Ada dataset compared to the Ada-to-Widar dataset. The analysis is that the scale of Widar3.0 dataset is larger and simpler than Ada dataset so that the base models trained with Widar3.0 dataset can obtain a better generalization ability. On the flip side, the Ada-to-Widar setting is more sophisticated which requires a higher adaptation ability and feature extraction capability for models from limited data. Cross-dataset experiments inspire us with two perspectives, collecting more samples and considering more complex scenarios (*e.g.* different environments and different Wi-Fi routers), to improve the generalization ability of models.

C. *Effectiveness of Multi-Sensor Collaboration*

A key advantage of our system is enabling flexible collaboration of multiple sensors. In this experiment, we show the effectiveness of such multi-sensor collaboration by testing our system with different number of Wi-Fi links.

We randomly select one to three links in the source and target environments and measure the adaptation accuracy with the selected links. The results, as shown in Fig 8, reveal that the gesture classification accuracy increases with the number of links. However, when there is only one link in the target environment, the accuracy of the system is poor (below 30%), even if three links are available in the source environment for training. This is because the single link may not well depict the sensing target and generate enough virtual intermediate domains. Similarly, when there is only one link on the source domain, the accuracy is still low since the system cannot learn enough domain-independent knowledge to transfer to the target environment. When the model has more than two links, the accuracy soars to above 80% and eventually achieves 90% with three links.

This experiment also provides empirical evidence that our system can allow adding or removing sensing devices flexibly. In our training and inference processes, the amount and order of the sensors do not affect the model prediction. This characteristic aligns better with real scenarios where the amount and order of sensors may be changed at any time due to dynamic enrollment or disconnection of IoT devices.

D. *Robustness against Low-Quality Sensors*

In this part, we analyze the influence of low-quality signals on the sensing systems. We first analyze the contribution of each Wi-Fi link in our self-collected environment. In Fig 9, we obtain the adaptation accuracy achieved with each individual sensor and all three sensors collaboratively for three users. It shows that the accuracy achieved with Sensor 2 is the lowest (approximately only 33%) among the three sensors. It is because it was positioned in the corner of the room farthest from the target user. Instead, Sensor 1 has the best location and the strongest sensing signal, which leads to the highest accuracy (around 70%).

Our approach is able to coordinate multiple sensors for collaborative sensing, which results in about 10% to 40% accuracy improvement as compared to using an individual (high-quality or low-quality) sensor. The improved accuracy of AdaWiFi is mainly due to the ability to accurately analyze the

TABLE V: Sensing adaptation accuracy of different methods on Widar3.0 datasets with a synthetic low-quality sensor. *Random Noise* and *Packet Loss* are two strategies to simulate the low-quality data.

Method	Random Noise		Packet Loss	
	1 shot	10 shot	1 shot	10 shot
MetaSense	50.68%	55.53%	54.00%	62.65%
RFNet	46.89%	62.50%	56.25%	67.19%
ProtoNet	68.73%	79.74%	62.07%	82.96%
EI	70.00%	83.33%	58.33%	71.67%
CADA	54.00%	80.00%	56.67%	70.00%
AdaWiFi	86.67%	86.67%	85.00%	90.00%

quality of different signals and generate a reasonable weight distribution for them. Meanwhile, we note that under such conditions, the accuracy of recognition can still be further improved by using collaborative sensing. In other words, low-quality signals can still supplement a certain amount of information to the system.

To further examine the ability of AdaWiFi in tolerating low-quality signals when implementing cross-environment adaptation. We add a synthetic noisy Wi-Fi link to the Widar3.0 dataset to simulate the low-quality signals. Specifically, we adopt two strategies to simulate the low-quality signals. One is *Random Noise*, where the signal values of the added synthetic link are sampled from a standard Gaussian distribution. It simulates the case that the device is completely uninformative. Another is *Packet Loss*, where a random portion (50%) of the signals in the time domain are masked (*i.e.* set to zero) to simulate the case of unstable connection.

We test the performance of different adaptation methods on the modified dataset, and the results are shown in Table V. The accuracies of the baseline methods drop significantly as compared with the values in Table IV. Specifically, all of them experience 10%-20% accuracy decrease. On the contrary, the cross-environment adaptation accuracy of AdaWiFi is not significantly affected. In 1-shot setting, the accuracy of AdaWiFi remains around 85%, similar to the results on the original data. While in the 10-shot setting, there is only a 1%-5% degradation as compared with the clean data, still better than other models. These results demonstrate the ability of AdaWiFi to mitigate the influence of low-quality signals. Meanwhile, it also suggests that the other methods that do not consider the influence of noise may not be sufficient to deal with the situations in real homes, where the data quality may be influenced by the hardware, network, and deployment.

E. System Overhead

We further analyze the overhead of our system. including the training overhead and the runtime inference overhead.

With three Wi-Fi links, it takes around 830 ms to train our model for one epoch on NVIDIA 3090 Ti. To adapt the model between environments, it takes about 1,640 ms for each epoch. In our experiments, training with 300 epochs and adapting with 120 epochs are enough to achieve a high adaptation accuracy.

TABLE VI: Runtime overhead of AdaWiFi under different configurations. T_{client} and T_{master} stand for the processing delay on the client device and the master device in milliseconds, D_{trans} represents the amount of data transmitted between them. We use three clients and one master, they are all Raspberry Pis. The processing window size is 1 second and the Wi-Fi packet rate is 1K Hz.

Configuration	T_{client}	T_{master}	D_{trans}
Preprocess @ Client Encode+Predict @ Master	375	97	240 KB
Preprocess+Encode @ Client Predict @ Master	406	13	94 KB
All @ Master	–	$2,348$	21 MB

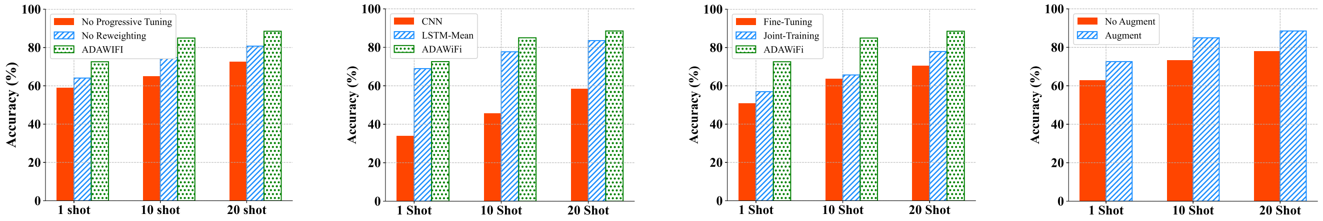
We further test the runtime system overhead of AdaWiFi with the Raspberry Pi. The workload of AdaWiFi mainly consists of three parts, including data preprocessing, encoding, and prediction after aggregation. Since our system allows different parts to be placed on either master or client devices (as mentioned in Section IV-B), we consider three workload distribution strategies and evaluate their performance respectively. Specifically, we calculate the delay on both the client and master devices, as well as the amount of data transmitted between them on the Raspberry Pi to make a prediction, as shown in Table VI. According to the results, configuration #2, *i.e.* the sensing signals are preprocessed and encoded on the client device before sending to the master device for aggregation and prediction, provides the lowest total delay, around 419 (406+13) ms, and the smallest data transmission (98 KB). This is because configuration #2 places the encoding part on the client device, thereby utilizing each device's computing ability and reducing the computational load of the master device. Furthermore, the data after encoding is reduced in dimension, enabling rapid aggregation and prediction. Instead, configuration #3 imposes an excess burden, as the client device is only responsible for receiving CSI data and sending the raw data to the master device for extensive centralized processing.

Table VI implies that utilizing a distributed scheme can reasonably allocate resources and increase computing efficiency. Nevertheless, the optimal configuration depends on the computation power of devices. For instance, if the master device is powerful, configuration #1 or #3 may get better. Due to the design of AdaWiFi, our system can flexibly deal with dynamic resource scheduling, and the performance optimization will be further revealed with the increase of devices.

F. Variation Analysis

We further analyze the performance of different potential variations of AdaWiFi to understand the contribution of each component in our design.

Effectiveness of Proposed Adaptation Technologies. Our primary novel adaptation techniques encompass progressive tuning with virtual domains and sensor reweighting. To evaluate their efficacy within AdaWiFi, we conduct separate analyses by disabling each method. As depicted in Fig 10a, the accuracy of both techniques exhibit 8%-20% decrease. Significantly, the decrease in performance is more conspicuous for progressive tuning, amounting to a minimum of 15%, in



(a) Accuracy w/o key adaptation technologies.

(b) Accuracy with different model backbones.

(c) Accuracy with different adaptation methods.

(d) Accuracy w/o sample-level data augmentation.

Fig. 10: The cross-environment adaptation performance of different variations of ADAWIFI.

contrast to sensor reweighting, which registers a minimum of 8%. This disparity can be attributed to the construction of virtual domains, enabling the generation of additional virtual samples that facilitate a smoother transition from the source domain to the target domain. In contrast, the reweighting process is less impacted due to the inherent capacity of deep learning models to implicitly discern sensor contributions. Nonetheless, in the absence of our proposed explicit weight assignment, a significant performance degradation is observed. Subsequent experiments involving the removal of both components will be discussed in Fig 10c under a fine-tuning case. Therefore, both two components are indispensable for effective adaptation.

Comparison of Different Model Backbones. We introduce a tailored model architecture for the collective sensing problem. In this study, we aim to examine the effectiveness of this architecture by replacing it with other alternatives. Specifically, we consider a Convolutional Neural Network (CNN) backbone and a LSTM-Mean backbone. The CNN backbone takes the stacked the DFS profile image as the input and makes prediction with four convolutional layers and two fully connected layers. The LSTM-Mean backbone utilizes Long Short-Term Memory (LSTM) to extract features and aggregate sensor embeddings by computing their mean value. As shown in Fig 10b, the CNN backbone achieves an accuracy of only 58% in 20-shot setting, with further decreases in precision as the number of samples reduced. Although the LSTM-Mean backbone demonstrates the ability to solve cross-domain problems to a certain extent, it still lag behind ADAWIFI by 5% in all N-shot settings. Therefore, ADAWIFI has the better architecture to deal with sensing adaptation challenges.

Comparison of Common Adaptation Methods. We further analyze the effectiveness of the adaptation method in ADAWIFI by replacing it with two common transfer learning methods, fine-tuning and joint-training [60], [61]. Fine-tuning adjusts the parameters of the pre-trained model using a small set of labeled samples from the target environment, while joint-training utilizes the data from both the source and the target domains to jointly optimize the model which can achieve better performance in the target domain and avoid forgetting the knowledge in the source domain. As shown in Fig 10c, joint-training outperforms fine-tuning slightly. This is due to the fact that joint-training can better align the samples from the two domains and thus compensate for the data gap. However, ADAWIFI still achieves 10-20% higher accuracy over them. This experiment illustrates the effectiveness of designing a sensing-specific adaptation method to address complex adaptation issues.

Effectiveness of Data Augmentation. This experiment investigates the effectiveness of our proposed data augmentation technique. Fig 10d shows that there is at least a 10% reduction in the adaptation accuracy if the model is adapted without the sample-level data augmentation, which demonstrates the effectiveness of our proposed data processing technique.

VI. Discussions

We discuss several limitations of ADAWIFI and our future work. First, receiving CSI data at a high sampling rate may cause interference in signals on the same channel. The situation would be even worse for a multi-link sensing system like ours. Simultaneous communicating and sensing is an important research direction to push the Wi-Fi sensing techniques into real deployments. Second, in ADAWIFI, we mainly use supervised learning to train our sensing model. The training samples are manually segmented and labeled. How to effectively leverage the unlabeled raw sensing streams (*e.g.* using self-supervised learning) would be an interesting research question. Third, there usually exist other modalities of wireless signals in many sensing environments. Due to the distributed design, it is possible for our system to use different modalities of signals to achieve more robust sensing adaptation. We leave it as our future work.

VII. Conclusion

In this paper, we have proposed a learning-based Wi-Fi sensing framework that leverages the collaboration of IoT devices to achieve more effective cross-environment adaptation. We have designed a model architecture that utilizes complementary information of distinct devices and an accompanying model adaptation technique that can transfer the sensing model to new environments with limited data. Experiments on both custom and public datasets have demonstrated that ADAWIFI is able to achieve significantly better adaptation accuracy than strong baselines. Our techniques would help push Wi-Fi sensing to more practical smart-home applications.

References

[1] H. Abdelnasser, M. Youssef, and K. A. Harras, "Wigest: A ubiquitous wifi-based gesture recognition system," in *2015 IEEE conference on computer communications (INFOCOM)*. IEEE, 2015, pp. 1472–1480.

[2] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals," in *Proceedings of the 19th annual international conference on Mobile computing & networking*, 2013, pp. 27–38.

[3] W. He, K. Wu, Y. Zou, and Z. Ming, "Wig: Wifi-based gesture recognition system," in *2015 24th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2015, pp. 1–7.

[4] X. Zhang, C. Tang, K. Yin, and Q. Ni, "Wifi-based cross-domain gesture recognition via modified prototypical networks," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8584–8596, 2021.

[5] X. Liu, J. Cao, S. Tang, and J. Wen, "Wi-sleep: Contactless sleep monitoring via wifi signals," in *2014 IEEE Real-Time Systems Symposium*. IEEE, 2014, pp. 346–355.

[6] F. Lin, Y. Zhuang, C. Song, A. Wang, Y. Li, C. Gu, C. Li, and W. Xu, "Sleepsense: A noncontact and cost-effective sleep monitoring system," *IEEE transactions on biomedical circuits and systems*, vol. 11, no. 1, pp. 189–202, 2016.

[7] J. Liu, Y. Chen, Y. Wang, X. Chen, J. Cheng, and J. Yang, "Monitoring vital signs and postures during sleep using wifi signals," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2071–2084, 2018.

[8] H. Wang, D. Zhang, Y. Wang, J. Ma, Y. Wang, and S. Li, "Rt-fall: A real-time and contactless fall detection system with commodity wifi devices," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 511–526, 2016.

[9] Y. Wang, K. Wu, and L. M. Ni, "Wifall: Device-free fall detection by wireless networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 581–594, 2016.

[10] S. Palipana, D. Rojas, P. Agrawal, and D. Pesch, "Falldefi: Ubiquitous fall detection using commodity wi-fi devices," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 4, pp. 1–25, 2018.

[11] Z. Chen, H. Zou, H. Jiang, Q. Zhu, Y. C. Soh, and L. Xie, "Fusion of wifi, smartphone sensors and landmarks using the kalman filter for indoor localization," *Sensors*, vol. 15, no. 1, pp. 715–732, 2015.

[12] M. Abbas, M. Elhamshary, H. Rizk, M. Torki, and M. Youssef, "Wideep: Wifi-based accurate and robust indoor localization system using deep learning," in *2019 IEEE International Conference on Pervasive Computing and Communications (PerCom*. IEEE, 2019, pp. 1–10.

[13] Y. Shu, C. Bo, G. Shen, C. Zhao, L. Li, and F. Zhao, "Magicol: Indoor localization using pervasive magnetic field and opportunistic wifi sensing," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 7, pp. 1443–1457, 2015.

[14] F. Wang, J. Feng, Y. Zhao, X. Zhang, S. Zhang, and J. Han, "Joint activity recognition and indoor localization with wifi fingerprints," *IEEE Access*, vol. 7, pp. 80 058–80 068, 2019.

[15] J. Liu, C. Xiao, K. Cui, J. Han, X. Xu, and K. Ren, "Behavior privacy preserving in rf sensing," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 1, pp. 784–796, 2023.

[16] P. Zhao, W. Liu, G. Zhang, Z. Li, and L. Wang, "Preserving privacy in wifi localization with plausible dummy locations," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 11 909–11 925, 2020.

[17] M. A. Tuan Tran, T. N. Le, and T. P. Vo, "Smart-config wifi technology using esp8266 for low-cost wireless sensor networks," in *2018 International Conference on Advanced Computing and Applications (ACOMP)*, 2018, pp. 22–28.

[18] M. Won, S. Sahu, and K.-J. Park, "Deepwitraffic: Low cost wifi-based traffic monitoring system using deep learning," in *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 2019, pp. 476–484.

[19] C. Li, Z. Cao, and Y. Liu, "Deep ai enabled ubiquitous wireless sensing: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–35, 2021.

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

[21] H. Jiang, C. Cai, X. Ma, Y. Yang, and J. Liu, "Smart home based on wifi sensing: A survey," *IEEE Access*, vol. 6, pp. 13 317–13 325, 2018.

[22] Y. Ma, G. Zhou, and S. Wang, "Wifi sensing with channel state information: A survey," *ACM Comput. Surv.*, vol. 52, no. 3, jun 2019. [Online]. Available: https://doi.org/10.1145/3310194

[23] Z. Zhou, F. Wang, J. Yu, J. Ren, Z. Wang, and W. Gong, "Target-oriented semi-supervised domain adaptation for wifi-based har," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 420–429.

[24] G. Yin, J. Zhang, G. Shen, and Y. Chen, "Fewsense, towards a scalable and cross-domain wi-fi sensing system using few-shot learning," *IEEE Transactions on Mobile Computing*, 2022.

[25] H. Zou, J. Yang, Y. Zhou, L. Xie, and C. J. Spanos, "Robust wifi-enabled device-free gesture recognition via unsupervised adversarial domain adaptation," in *2018 27th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2018, pp. 1–8.

[26] Y. Zheng, Y. Zhang, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Zero-effort cross-domain gesture recognition with wi-fi," in *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, 2019, pp. 313–325.

[27] R. Gao, M. Zhang, J. Zhang, Y. Li, E. Yi, D. Wu, L. Wang, and D. Zhang, "Towards position-independent sensing for gesture recognition with wi-fi," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 2, pp. 1–28, 2021.

[28] W. Jiang, C. Miao, F. Ma, S. Yao, Y. Wang, Y. Yuan, H. Xue, C. Song, X. Ma, D. Koutsonikolas *et al.*, "Towards environment independent device free human activity recognition," in *Proceedings of the 24th annual international conference on mobile computing and networking*, 2018, pp. 289–304.

[29] H. Zou, Y. Zhou, J. Yang, H. Liu, H. P. Das, and C. J. Spanos, "Consensus adversarial domain adaptation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 5997–6004.

[30] Z. Wang, S. Chen, W. Yang, and Y. Xu, "Environment-independent wi-fi human activity recognition with adversarial network," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3330–3334.

[31] S. Ding, Z. Chen, T. Zheng, and J. Luo, "Rf-net: A unified meta-learning framework for rf-enabled one-shot human activity recognition," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, 2020, pp. 517–530.

[32] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[33] T. Gong, Y. Kim, J. Shin, and S.-J. Lee, "Metasense: few-shot adaptation to untrained conditions in deep mobile sensing," in *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, 2019, pp. 110–123.

[34] S. Tan, Y. Ren, J. Yang, and Y. Chen, "Commodity wifi sensing in ten years: Status, challenges, and opportunities," *IEEE Internet of Things Journal*, vol. 9, no. 18, pp. 17 832–17 843, 2022.

[35] X. Zhu, Y. Feng *et al.*, "Rssi-based algorithm for indoor localization," *Communications and Network*, vol. 5, no. 02, p. 37, 2013.

[36] S. Sigg, U. Blanke, and G. Tröster, "The telepathic phone: Frictionless activity recognition from wifi-rssi," in *2014 IEEE international conference on pervasive computing and communications (PerCom)*. IEEE, 2014, pp. 148–155.

[37] A. Zhuravchak, O. Kapshii, and E. Pournaras, "Human activity recognition based on wi-fi csi data-a deep neural network approach," *Procedia Computer Science*, vol. 198, pp. 59–66, 2022.

[38] J. Yang, H. Zou, H. Jiang, and L. Xie, "Device-free occupant activity sensing using wifi-enabled iot devices for smart homes," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3991–4002, 2018.

[39] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of wifi signal based human activity recognition," in *Proceedings of the 21st annual international conference on mobile computing and networking*, 2015, pp. 65–76.

[40] D. Zhang, H. Wang, and D. Wu, "Toward centimeter-scale human activity sensing with wi-fi signals," *Computer*, vol. 50, no. 1, pp. 48–57, 2017.

[41] X. Li, D. Zhang, Q. Lv, J. Xiong, S. Li, Y. Zhang, and H. Mei, "Indotrack: Device-free indoor human tracking with commodity wi-fi," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, pp. 1–22, 2017.

[42] J. Zhang, Z. Tang, M. Li, D. Fang, P. Nurmi, and Z. Wang, "Crosssense: Towards cross-site and large-scale wifi sensing," in *Proceedings of the 24th annual international conference on mobile computing and networking*, 2018, pp. 305–320.

[43] F. Wang, J. Liu, and W. Gong, "Wicar: Wifi-based in-car activity recognition with multi-adversarial domain adaptation," in *Proceedings of the International Symposium on Quality of Service*, 2019, pp. 1–10.

[44] B.-B. Zhang, D. Zhang, Y. Li, Y. Hu, and Y. Chen, "Unsupervised domain adaptation for device-free gesture recognition," *arXiv preprint arXiv:2111.10602*, 2021.

[45] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[46] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016.

[47] G. Csurka, “Domain adaptation for visual applications: A comprehensive survey,” *arXiv preprint arXiv:1702.05374*, 2017.

[48] A. Farahani, S. Voghoei, K. Rasheed, and H. R. Arabnia, “A brief review of domain adaptation,” *Advances in Data Science and Information Engineering: Proceedings from ICDATA 2020 and IKE 2020*, pp. 877–894, 2021.

[49] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, “Meta-learning in neural networks: A survey,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 9, pp. 5149–5169, 2021.

[50] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *International conference on machine learning*. PMLR, 2017, pp. 1126–1135.

[51] A. Antoniou, H. Edwards, and A. Storkey, “How to train your maml,” *arXiv preprint arXiv:1810.09502*, 2018.

[52] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, “A closer look at few-shot classification,” *arXiv preprint arXiv:1904.04232*, 2019.

[53] Y. Guo, N. C. Codella, L. Karlinsky, J. V. Codella, J. R. Smith, K. Saenko, T. Rosing, and R. Feris, “A broader study of cross-domain few-shot learning,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*. Springer, 2020, pp. 124–141.

[54] Y. Ma, G. Zhou, and S. Wang, “Wifi sensing with channel state information: A survey,” *ACM Computing Surveys (CSUR)*, vol. 52, no. 3, pp. 1–36, 2019.

[55] Y. Yu, X. Si, C. Hu, and J. Zhang, “A review of recurrent neural networks: Lstm cells and network architectures,” *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[56] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, “Recent advances in recurrent neural networks,” *arXiv preprint arXiv:1801.01078*, 2017.

[57] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[58] Y. Xie, Z. Li, and M. Li, “Precise power delay profiling with commodity wifi,” in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, ser. MobiCom ’15. New York, NY, USA: ACM, 2015, p. 53–64. [Online]. Available: http://doi.acm.org/10.1145/2789168.2790124

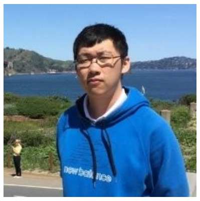
[59] Y. Zheng, C. Wu, K. Qian, Z. Yang, and Y. Liu, “Detecting radio frequency interference for csi measurements on cots wifi devices,” in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[60] Z. Li and D. Hoiem, “Learning without forgetting,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[61] R. Caruana, “Multitask learning,” *Machine learning*, vol. 28, pp. 41–75, 1997.

Naiyu Zheng is a Ph.D. student at the department of computer science, City University of Hong Kong. He received the Bachelor degree in computer science from Beijing University of Posts and Telecommunications (BUPT) in 2023. His research interests include wireless sensing and edge computing.

Yuanchun Li is a research assistant professor at the Institute for AI Industry Research (AIR), Tsinghua University. Before joining AIR, he got his Ph.D. and B.S. in Computer Science from Peking University, and was a Senior Researcher at Microsoft Research Asia. His research interests lie in the efficiency and reliability of edge AI systems. His work won the UbiComp Honorable Mention Award and IS-EUD Best Paper Award, and the related systems and tools are widely used in the open-source community. He is a member of ACM and a member of IEEE.



Shiqi Jiang is the Senior Researcher with Systems and Networking Research Group, Microsoft Research Asia (MSRA). He received the Ph.D. degree in computer science from Nanyang Technological University in 2018, and the Bachelor degree in computer engineering from Zhejiang University.

Yuanzhe Li is a postdoc at the Institute for AI Industry Research (AIR), Tsinghua University. He received his Ph.D. in 2022 from the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications (BUPT). His work won the best paper award from CollaborateCom 2022. His research interests include mobile edge computing, cloud computing and service computing.

Rongchun Yao is a software development engineer at Tencent. He received his bachelor degree and master degree from Nanjing University.

Chuchu Dong received his Ph.D. degree in computer science from the University of Chinese Academy of Sciences, China. In 2018, He joined the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences as a postdoc. He is a researcher at Midea Group. His research interests include IoT, wireless networking, and intelligent home system.

Ting Chen is a deputy general manager of R&D center at Radar Automotive, Geely Group. He received his Ph.D. degree from Technical University of Hamburg, Germany. Before 2022, he was the director of IoT technology at Midea Group. His research interests are mobile communications and IoT applications in smart home and intelligent automotive. He was the vice chairman of User Experience Task Group at Wi-Fi Alliance.

Yubo Yang is an embedded software development engineer in the intelligent wireless communication industry. He received a master's degree from King's College London. His research interests are mobile communications, smart home, and Internet of Things.

Zhimeng Yin is an assistant professor at the Department of Computer Science at the City University of Hong Kong. He obtained his Ph.D. degree from the University of Minnesota in 2020. Before that, He received my bachelor's degree and master's degree from Huazhong University of Science and Technology in 2011 and 2014, respectively.

Yunxin Liu (Senior Member, IEEE) is a Guoqiang Professor at Institute for AI Industry Research (AIR),Tsinghua University. He received his Ph.D. degree from Shanghai Jiao Tong University (SJTU). His research interests are mobile computing and edge computing. He received MobiSys 2021 Best Paper Award, SenSys 2018 Best Paper Runner-up Award, MobiCom 2015 Best Demo Award, and PhoneSense 2011 Best Paper Award.