BeamSense: Rethinking Wireless Sensing with MU-MIMO Wi-Fi Beamforming Feedback

Khandaker Foysal Haque, *Graduate Student Member, IEEE*, Milin Zhang, *Graduate Student Member, IEEE*, Francesca Meneghello, *Member, IEEE*, and Francesco Restuccia, *Senior Member, IEEE*

***Abstract*—In this paper, we propose `BeamSense`, a completely novel approach to implement standard-compliant Wi-Fi sensing applications. Wi-Fi sensing enables game-changing applications in remote healthcare, home entertainment, and home surveillance, among others. However, existing work leverages the manual extraction of channel state information (CSI) from Wi-Fi chips to classify activities, which is not supported by the Wi-Fi standard and hence requires the usage of specialized equipment. On the contrary, `BeamSense` leverages the standard-compliant beamforming feedback information (BFI) to characterize the propagation environment. Conversely from CSI, the BFI (i) can be easily recorded without any firmware modification, and (ii) captures the multiple channels between the access point and the stations, thus providing much better sensitivity. `BeamSense` includes a novel cross-domain few-shot learning (FSL) algorithm to handle unseen environments and subjects with few additional data points. We evaluate `BeamSense` through an extensive data collection campaign with three subjects performing twenty different activities in three different environments. We show that our BFI-based approach achieves about 10% more accuracy when compared to CSI-based prior work, while our FSL strategy improves accuracy by up to 30% and 80% when compared with state-of-the-art cross-domain algorithms.**

***Index Terms*—Wi-Fi sensing, IEEE 802.11ac, SU-MIMO, MU-MIMO, beamforming, beamforming feedback angles**

I. Introduction

Since 1990, Wi-Fi has become the technology of choice for Internet connectivity in indoor environments [1]. Beyond connectivity, Wi-Fi signals can be used as sounding waveforms to perform activity recognition [2], health monitoring [3], and human presence detection [4], among others [5]. The intuition behind Wi-Fi sensing is that humans act as obstacles to the propagation of radio signals in the environment. Specifically, when encountering the human body, the radio waves undergo reflections, diffractions and scattering that make the signals collected at the Wi-Fi receiver differ from the transmitted ones. Wi-Fi sensing aims at detecting the changes in the Wi-Fi signals and associating them to the way the subject stays/moves in the environment, thus realizing device-free monitoring solutions. To date, the vast majority of Wi-Fi sensing systems – discussed in Section II – leverage channel measurements obtained from pilot symbols as sensing primitive. Such measurements are usually referred to as channel state information (CSI) and describe the way the signals propagate in the environment. Despite leading to good performance, CSI-based techniques require extracting and recording the CSI estimated by the Wi-Fi devices involved in the sensing activities, and such operations are currently not supported by the IEEE 802.11 standard. This has led to the introduction of custom-tailored firmware modifications to extract the CSI [6], [7], [8], [9], [10], which makes the sensing process not scalable. Such CSI extraction tools only provide support for single-user multiple-input multiple-output (MIMO) sensing as the channel is sounded on the link between the transmitter and the device implementing the extraction tool. Therefore, Wi-Fi sensing approaches relying on CSI extraction tools cannot benefit from the spatial diversity that can be gained through multi-user MIMO (MU-MIMO) transmissions. Spatial diversity may be achieved considering multiple CSI collectors but this would increase the computation burden as synchronization among the devices would be needed. Moreover, even if CSI extraction could be supported in the future without the need for custom-tailored firmware modifications, it would require additional processing to extract the data from the chip, thus increasing energy consumption. Therefore, we argue that more suitable approaches to Wi-Fi sensing should be put forward.

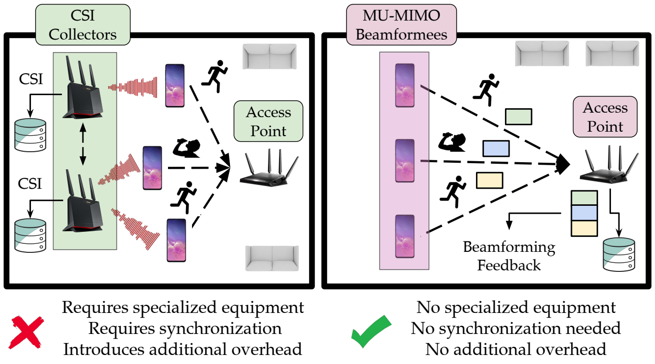


Fig. 1: CSI-based vs BFI-based Wi-Fi sensing.

In this paper, we propose `BeamSense`, an entirely new approach to Wi-Fi sensing that leverages the MU-MIMO capabilities of Wi-Fi to drastically increase sensing performance while substantially reducing sensing overhead. As shown in Figure 1, `BeamSense` leverages the beamforming feedback information (BFI) – traditionally used to beamform transmissions – to estimate the propagation environment between the access point (AP) and the connected stations (STAs). In stark contrast with CSI-based sensing, `BeamSense` (i) does not need firmware modifications, since any off-the-shelf Wi-Fi device can capture BFI packets, which are sent unencrypted to keep the processing delay below a few milliseconds [11]; and (ii) does not require synchronization among receivers, since a single BFI report contains the information about all the

Khandaker Foysal Haque, Milin Zhang and Francesco Restuccia are with the Institute for the Wireless Internet of Things, Northeastern University, United States, e-mail: {haque.k, zhang.mil, frestuc}@northeastern.edu.

F. Meneghello is with the Department of Information Engineering, University of Padova, Italy, e-mail: francesca.meneghello.1@unipd.it.

MIMO channels established between the AP and the STAs. In fact, while devices empowered with CSI extraction tools allow obtaining information on a single MIMO channel, when capturing the BFI we obtain the channel information associated with all the STAs involved in a MU-MIMO transmission. Thus, multiple spatially diverse channel information is collected with a single capture. For this reason, `BeamSense` exhibits far better performance in challenging environments, as shown in Section IV.

This paper provides the following contributions:

- We propose `BeamSense`, a new approach to Wi-Fi sensing where the standard-compliant BFI routinely sent in MU-MIMO Wi-Fi networks is used to characterize the propagation environment between the MU-MIMO users and the AP. To the best of our knowledge, this is the first work proposing the utilization of BFI to perform Wi-Fi sensing;
- We propose a deep learning (DL)-based Fast and Adaptive Micro Reptile Sensing (FAMReS) algorithm to perform activity classification based on BFI. We chose DL since it has shown remarkable performance in classifying activities in Wi-Fi sensing settings [12]. However, it is well-known that DL models may perform poorly when tested in different settings [13]. For this reason, FAMReS leverages few-shot learning (FSL) to quickly generalize to different subjects and environments with few additional data points;
- We extensively evaluate `BeamSense` through a comprehensive data collection campaign, with three subjects performing twenty different activities in three different environments. For that, we built a reconfigurable IEEE 802.11ac MU-MIMO network with three STAs and one AP. The Wi-Fi network was also synchronized with a camera-based system that records the ground truth for our experiments and a secondary IEEE 802.11ac network empowered with Nexmon CSI [8] to concurrently collect the CSI measurements used for comparative analysis. We show that our BFI-based approach combined with a traditional convolutional neural network (CNN) without pre-processing achieves about 10% more accuracy when compared to state-of-the-art CSI-based techniques, which uses pre-processing. Moreover, FAMReS improves accuracy by up to 30% and 80% when compared with state-of-the-art cross-domain algorithms. **For reproducibility, we will release the entirety of our 800 GB dataset and our code.**

The rest of the article is organized as follows. In Section II we review the existing literature in the area. The `BeamSense` Wi-Fi sensing system is illustrated in Section III whereas the performance evaluation of the system is presented in Section IV. Section V concludes the discussion.

II. Related Work

Over the last ten years, a lot of efforts have been made to explore wireless sensing, which is summarized by Liu et al. in [14]. The first Wi-Fi sensing approaches were based on the received signal strength indicator (RSSI) [15], [16], [17], [18], [19], [20]. More recently, researchers have focused on the more fine-grained CSI information that describes how the wireless channel modifies signals at different frequencies rather than providing a cumulative metric on the signal attenuation as the RSSI does. Passive Wi-Fi radar (PWR)-based approaches [21], [22], [23], [24], [25] have also been proposed in the literature. However, such an approach requires specialized hardware (software defined radio (SDR)) to analyze the collected signal. In the rest of the section, we focus on CSI-based sensing, and summarize the main research on the topic.

Background on CSI-based Sensing. The term CSI can refer both to the time-domain channel impulse response (CIR) or the frequency-domain channel frequency response (CFR). Specifically, the CIR encodes the information about the multipath propagation of the transmitted signal: each peak in the CIR represents a propagation path characterized by a specific time delay (linked with the length of the path) and an attenuation. Multipath propagation is a typical phenomenon of indoor environments, where obstacles (objects, people, animals) in the surroundings act as reflectors/diffractors/scatterers for the irradiated wireless signals. In turn, the receiver collected different copies of the transmitted signal each associated with a different propagation, or, equivalently, an obstacle in the environment. The CFR represents the Fourier transform of the CIR and describes how the environment modifies signals transmitted with different carrier frequencies. Specifically, indicating with $\mathbf{x}(f,t)$ and $\mathbf{y}(f,t)$ the frequency domain representation of the transmitted and received signals at time t and frequency f respectively, and with $\mathbf{h}(f,t)$ the CFR, we have that $\mathbf{y}(f,t) = \mathbf{h}(f,t) \times \mathbf{x}(f,t)$ [26]. Considering the $M \times N$ MIMO orthogonal frequency-division multiplexing (OFDM) system, with K sub-channels, and M and N transmitting and receiving antennas respectively, the CFR is a $K \times M \times N$-dimensional matrix providing the amplitude and phase information over each OFDM sub-channel for any given pair of transmitting and receiving antenna.

Existing Research on CSI-based Sensing. Over the last decade, CSI-based sensing has been proposed for a wide variety of applications. Among the most compelling, we mention person detection and identification [27], [28], [29], crowd counting [30], [18], respiration monitoring [31], baggage tracking [32], smart homes [33], [34], human pose tracking [35], [36], [37], [38], patient monitoring [39], [40], with most of the previous research activities focusing on human activity recognition (HAR) and human gesture recognition (HGR) [41], [42], [43], [44], [13], [45]. ***The above list is definitely not exhaustive.*** For excellent survey papers on the topic, we refer the reader to [46], [5], [2], [47]. In the following, we just summarize the most recent approaches that are most related to the work conducted in this article. Guo et al. presented WiAR [48], a CSI-based system achieving up to 90% accuracy in the recognition of 16 human activities. Similarly, a meta-learning-based approach called RF-Net was presented in [49] based on the usage of recurrent neural networks with long short-term memory (LSTM) cells. However, only six activities were considered in the evaluation. Regarding HGR, [43] and [44] presented Widar 3.0 and OneFi, respectively considering six and forty gestures. The authors in [43] proposed to use a body velocity profile (BVP) measure which has been shown to improve the generalization capability of the classification algorithm. The authors of [44] used one-shot learning to classify unseen gestures with few labeled samples. The majority of previous work has been evaluated on 802.11n channel data while, to the best of our knowledge, only two works considered HAR in the context of 802.11ac [13], [12]. Meneghello et al. proposed to use the Doppler shift estimated through the CSI to obtain an algorithm that generalizes to different environments [13]. Bahadori et al. use instead few-shot learning to achieve environmental robustness [12].

Limitations of CSI-based Sensing. Since the CSI is computed at the physical layer (PHY), it is not readily available with off-the-shelf network interface cards (NICs). Although CSI can be extracted with SDR implementations, which only support up to 40 MHz of bandwidth, being only IEEE 802.11 a/g/p/n compliant [50], [12]. Moreover, SDRs are costly specialized hardware that may be unavailable in real-life situations and require expert knowledge to be used. To overcome such limitations, in recent years, researchers have developed some CSI extraction tools that run on commercial Wi-Fi NICs. Two of them, namely Linux CSI [6] and Atheros CSI [7], target IEEE 802.11n compliant NICs (up to 40 MHz bandwidth). The third one, Nexmon CSI [8], allows extracting the CFR from some IEEE 802.11ac compliant devices, supporting bandwidths up to 80 MHz. The most recent one, AX CSI [10] is designed for IEEE 802.11ax devices and provides CFR measurements also on 160 MHz bandwidth channels. These tools, however, need non-trivial firmware modifications of the NICs. Moreover, they do not provide support for estimating the channel on MU-MIMO channels. Both when the CSI extractor tool is implemented on one receiving Wi-Fi device or on another monitor device, only the MIMO links between the transmitter and the CSI collector is monitored, i.e., only SU-MIMO mode is supported. This is a limitation of CSI-based systems as MU-MIMO systems can provide way richer information than SU-MIMO ones as they capture the correlation of the propagated signal from different STAs relative to the sensed subject. As a last consideration, Wang et al. [51] recently pointed out the importance of the placement of the CSI extractor device. Specifically, they showed that accurate placement of the sensing devices can enhance the sensing coverage by mitigating severe interference. Non-calibrated placement of the sensing devices can severely hamper the sensing quality.

Advantages of `BeamSense`. Our approach addresses these challenges by exploiting the MU-MIMO beamforming feedback to sense the environment. The collection of the MU-MIMO beamforming feedback packets can be done with any standard-compliant 802.11 ac/ax device, and it does not need any close proximity or direct access to the sensed subject. As our system does not need any specific hardware or infrastructure, it facilitates mass deployment. Moreover, since it utilizes the aggregated feedback from different users placed at different locations, `BeamSense` is less sensitive to the accurate placement of the STAs.

III. The BeamSense Wi-Fi Sensing System

Figure 2 shows a high-level overview of `BeamSense`, which leverages the channel estimation mechanism standardized in IEEE 802.11 to sound the physical environment. The

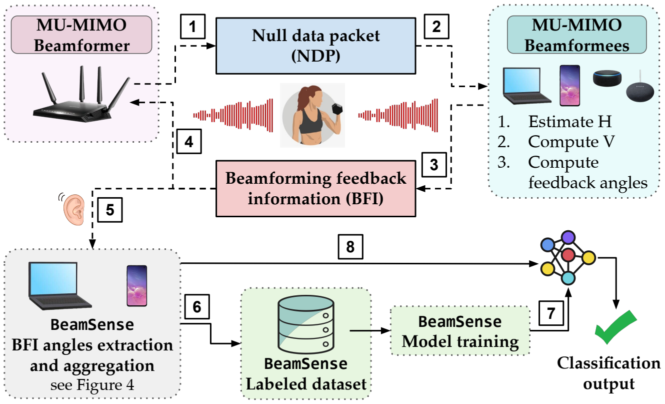

MU-MIMO Beamformer
1
Null data packet (NDP)
2
MU-MIMO Beamformees
1. Estimate H
2. Compute V
3. Compute feedback angles
4
3
Beamforming feedback information (BFI)
5
8
6
BeamSense BFI angles extraction and aggregation see Figure 4
BeamSense Labeled dataset
BeamSense Model training
7
Classification output

Fig. 2: The `BeamSense` Wi-Fi sensing system.

channel estimation is performed on the STAs (beamformees) and is reported to the AP (beamformer) that uses it to properly beamform MU-MIMO transmissions. The report is referred to as the BFI and is transmitted over the air in clear text. Since the AP continuously triggers the channel estimation procedure on the connected STAs, *the BFI contains very rich, reliable, and spatially diverse information*. Moreover, the BFI *can be collected with a single capture* by the AP or any other Wi-Fi-compliant device, thus reducing the system complexity.

`BeamSense` Technical Challenges. `BeamSense` is a completely novel way to perform Wi-Fi sensing. While previous work in the literature deal with the well-known CSI data, we instead consider the BFI as a sensing primitive. We stress that BFI represents a completely new type of data. While CSI consists of complex I/Q-values, BFI is expressed in terms of compressed rotational matrices. In this respect, the first challenge we need to address is the design and implementation of a novel tool to extract the BFI data embedded within Wi-Fi frames transmitted from the beamformees to the beamformer as part of the channel sounding procedure. On top of that, the second challenge concerns the implementation of a new data processing pipeline for the new data type that effectively performs activity classification based on BFI data and provides environment adaptation features. The third challenge to be addressed is the setup of an extensive experimental testbed to implement and assess the performance of the new Wi-Fi sensing approach in a real-world scenario with commercial Wi-Fi devices.

In the following, we thoroughly detail the `BeamSense` sensing system. We use the superscripts T and † to denote the transpose and the complex conjugate transpose (i.e., the Hermitian). We define with $\angle \mathbf{C}$ the matrix containing the phases of the complex-valued matrix $\mathbf{C}$. Moreover, $\text{diag}(c_1, \dots, c_j)$ indicates the diagonal matrix with elements $(c_1, \dots, c_j)$ on the main diagonal. The (c_1, c_2) entry of matrix $\mathbf{C}$ is defined by $[\mathbf{C}]_{c_1,c_2}$, while $\mathbb{I}_c$ refers to an identity matrix of size $c \times c$ and $\mathbb{I}_{c \times d}$ is a $c \times d$ generalized identity matrix.

A. BeamSense: A Walkthrough

The `BeamSense` sensing system entails eight steps, as depicted in Figure 2. The process stems from the way beamforming is implemented in IEEE 802.11 networks. Specifically, the beamformer (AP) uses a matrix $\mathbf{W}$ of pre-coding weights – called steering matrix – to linearly combine the signals to be simultaneously transmitted to the different beamformees (STAs). The steering matrix is derived from the CFR matrices $\mathbf{H}$ estimated by each of the beamformee and that describe how the environment modifies the irradiated signals in their path to the receivers. The estimation process is called *channel sounding* and is triggered by the AP which periodically broadcasts a null data packet (NDP) (**step 1** in Figure 2) that contains sequences of bits – named long training fields (LTFs) – the decoded version of which is known by the beamformees. Since its purpose is to sound the channel, the NDP *is not beamformed* by the AP. *This is particularly advantageous for sensing purposes*, since the resulting CFR estimation will not be affected by inter-stream or inter-user interference. The LTFs are transmitted over the different beamformer antennas in subsequent time slots, thus allowing each beamformee to estimate the CFR of the links between its receiving antennas and the beamformer transmitting antennas. The LTFs are modulated – as the data fields – through OFDM by dividing the signal bandwidth into K partially overlapping and orthogonal sub-channels spaced apart by $1/T$. The input bits are grouped into OFDM symbols, $\mathbf{a} = [a_{-K/2}, \dots, a_{K/2-1}]$, where a_k is named OFDM sample. These K OFDM samples are digitally modulated and transmitted through the K OFDM sub-channels in a parallel fashion thus occupying the channel for T seconds. The transmitted LTF signal is

$$s_{\text{tx}}(t) = e^{j2\pi f_c t} \sum_{k=-K/2}^{K/2-1} a_k e^{j2\pi kt/T}, \tag{1}$$

where f_c is the carrier frequency. The NDP is received and decoded by each STA (**step 2**) to estimate the CFR $\mathbf{H}$. The different LTFs are used to estimate the channel over each pair of transmitting (TX) and receiving (RX) antennas, for every OFDM sub-channel. This generates a $K \times M \times N$ matrix $\mathbf{H}$ for each beamformee, where M and N are respectively the numbers of TX and RX antennas. We refer the reader to Section II for additional details about the CFR. Next, the CFR is compressed – to reduce the channel overhead – and fed back to the beamformer. Using $\mathbf{H}_k$ to identify the $M \times N$ sub-matrix of $\mathbf{H}$ containing the CFR samples related to sub-channel k, the *compressed beamforming feedback* is obtained as follows ([52], Chapter 13). First, $\mathbf{H}_k$ is decomposed through singular value decomposition (SVD) as

$$\mathbf{H}_k^T = \mathbf{U}_k \mathbf{S}_k \mathbf{Z}_k^{\dagger}, \tag{2}$$

where $\mathbf{U}_k$ and $\mathbf{Z}_k$ are, respectively, $N \times N$ and $M \times M$ unitary matrices, while the singular values are collected in the $N \times M$ diagonal matrix $\mathbf{S}_k$. Using this decomposition, the complex-valued beamforming matrix $\mathbf{V}_k$ is defined by collecting the first $N_{\text{SS}} \leq N$ columns of $\mathbf{Z}_k$. Such a matrix is used by the beamformer to compute the pre-coding weights for the N_{SS} spatial streams directed to the beamformee. Hence, $\mathbf{V}_k$ is converted into polar coordinates as detailed in Algorithm 1 to avoid transmitting the complete matrix. The output is matrices $\mathbf{D}_{k,i}$ and $\mathbf{G}_{k,\ell,i}$, defined as

$$\mathbf{D}_{k,i} = \begin{bmatrix} \mathbb{I}_{i-1} & 0 & & \cdots & 0 \\ 0 & e^{j\phi_{k,i,i}} & 0 & \cdots & \vdots \\ \vdots & 0 & \ddots & 0 & \\ & \vdots & 0 & e^{j\phi_{k,M-1,i}} & 0 \\ 0 & \cdots & & 0 & 1 \end{bmatrix}, \tag{3}$$

$$\mathbf{G}_{k,\ell,i} = \begin{bmatrix} \mathbb{I}_{i-1} & 0 & & \cdots & 0 \\ 0 & \cos\psi_{k,\ell,i} & 0 & \sin\psi_{k,\ell,i} & \vdots \\ \vdots & 0 & \mathbb{I}_{\ell-i-1} & 0 & \\ & -\sin\psi_{k,\ell,i} & 0 & \cos\psi_{k,\ell,i} & 0 \\ 0 & \cdots & & 0 & \mathbb{I}_{M-\ell} \end{bmatrix}, \tag{4}$$

that allow rewriting $\mathbf{V}_k$ as $\mathbf{V}_k = \tilde{\mathbf{V}}_k \tilde{\mathbf{D}}_k$, with

$$\tilde{\mathbf{V}}_k = \prod_{i=1}^{\min(N_{\text{SS}},M-1)} \left(\mathbf{D}_{k,i} \prod_{l=i+1}^{M} \mathbf{G}_{k,l,i}^T \right) \mathbb{I}_{M \times N_{\text{SS}}}, \tag{5}$$

where the products represent matrix multiplications. In the $\tilde{\mathbf{V}}_k$ matrix, the last row – i.e., the feedback for the M-th transmitting antenna – consists of non-negative real numbers by construction. Using this transformation, the beamformee is only required to transmit the ϕ and ψ angles to the beamformer as they allow reconstructing $\tilde{\mathbf{V}}_k$ precisely. Moreover, it has been proved (see [52], Chapter 13) that the beamforming performance is equivalent at the beamformee when using $\mathbf{V}_k$ or $\tilde{\mathbf{V}}_k$ to construct the steering matrix $\mathbf{W}$. In turn, the feedback for $\tilde{\mathbf{D}}_k$ is not fed back to the beamformer. The angles are quantized using $b_\phi \in \{7, 9\}$ bits for ϕ and $b_\psi = b_\phi - 2$ bits for ψ, to further reduce the channel occupancy. The quantized values – $q_\phi = \{0, \dots, 2^{b_\phi} - 1\}$ and $q_\psi = \{0, \dots, 2^{b_\psi} - 1\}$ – are packed into the compressed beamforming frame (**step 3**) and such *beamforming feedback information* (BFI) is transmitted to the AP (**step 4**) in *clear text*. Each BFI contains A number of angles for each of the K OFDM sub-channels for a total of $(K \cdot A)$ angles each. In Figure 3, we show an example of how beamforming is conducted in a 3×2 MIMO system.

Algorithm 1: $\mathbf{V}_k$ matrix decomposition

```
Require: $\mathbf{V}_k$;
$\tilde{\mathbf{D}}_k = \text{diag}(e^{j\angle[\mathbf{V}_k]_{M,1}}, \dots, e^{j\angle[\mathbf{V}_k]_{M,N_{\text{SS}}}})$ ;
$\mathbf{\Omega}_k = \mathbf{V}_k \tilde{\mathbf{D}}_k^{\dagger}$;
for $i \leftarrow 1$ to $\min(N_{\text{SS}}, M-1)$ do
    $\phi_{k,\ell,i} = \angle\,[\mathbf{\Omega}_k]_{\ell,i}$ with $\ell = i, \dots, M-1$;
    compute $\mathbf{D}_{k,i}$ through Eq. (3);
    $\mathbf{\Omega}_k \leftarrow \mathbf{D}_{k,i}^{\dagger} \mathbf{\Omega}_k$;
    for $\ell \leftarrow i+1$ to $M$ do
        $\psi_{k,\ell,i} = \arccos\left(\frac{[\mathbf{\Omega}_k]_{i,i}}{\sqrt{[\mathbf{\Omega}_k]_{i,i}^2 + [\mathbf{\Omega}_k]_{\ell,i}^2}}\right)$;
        compute $\mathbf{G}_{k,\ell,i}$ through Eq. (4);
        $\mathbf{\Omega}_k \leftarrow \mathbf{G}_{k,\ell,i} \mathbf{\Omega}_k$;
```

`BeamSense` captures the BFI reports (**step 5**), and uses the channel estimation data to perform Wi-Fi sensing. We remark that, since MU-MIMO requires fine-grained channel

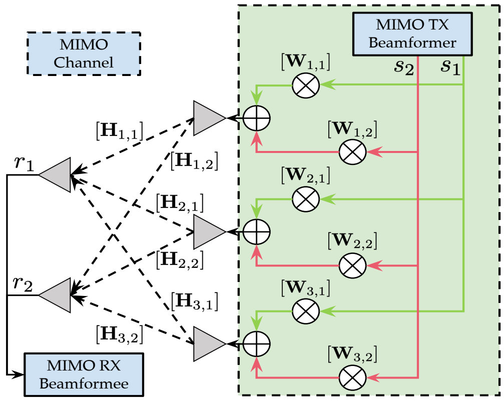


Fig. 3: Example of 3×2 MIMO system. s_1, s_2 and r_1, r_2 are respectively the transmitted and received signals. The symbol $\mathbf{W}$ indicates the steering matrix, while $\mathbf{H}$ is the CFR.

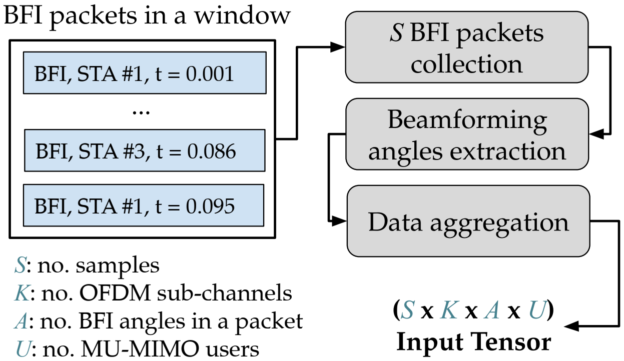


Fig. 4: BFI data processing. The processing is applied to each observation window of W seconds.

sounding – every around 10 milliseconds to account for user mobility, according to [53] – it is fundamental to process the BFI in a fast manner at the AP. For this reason, and since cryptography would lead to excessive delays, the angles are currently sent unencrypted. Therefore, the BFI reports are exposed to and can be read by any device that can access the wireless channel. Specifically, `BeamSense` relies on the BFI transmitted by all the beamformees in the environment and captured during a time window of W seconds to reliably estimate the activity being performed by a human moving within the propagation environment. This is done by using the BFI samples collected within the window as input for a learning-based algorithm (detailed in Section III-B). Note that, as `BeamSense` leverages ongoing MU-MIMO transmissions, there is no guarantee that the same number of BFI frames are collected within a specific time interval of W seconds. This is related to the fact that we have no control on when the beamformer triggers the channel sounding procedure that generates BFI data. Therefore, as the neural network-based classification algorithm requires the input to be of a fixed dimension, we need to determine a fixed-size input that represents the BFI reports captured during the time window. The processing is applied just after having collected the data on the wireless channel (grey box in Figure 2) and is summarized in Figure 4. Specifically, we consider the average number S of BFI packets counted (at training time) in each window during an activity recording. Windows having less than S packets are padded with BFI packets containing zero-valued angles while packets exceeding such threshold are discarded. Hence, the $K \times A$ BFI angles contained in each packet are extracted and the final tensor is obtained by aggregating the $S \times K \times A$ angles for all the U MU-MIMO users for which the BFI data have been captured in the observation window. Note that even if it would be possible to define learning algorithms that accept input of different sizes, this would lead to an increase in the complexity of the approach, both from the training and inference perspective. Therefore, to keep the model simple for implementation on memory- and battery-constrained devices, we decided to follow a fixed-input approach.

To obtain the training data, the $S \times K \times A \times U$ tensors derived from the BFI packets captured during the data collection phase are stored in a dataset, together with their associated activity and/or phenomenon, and a timestamp (**step 6** in Figure 2). This phase can be performed offline by sensing application vendors without requiring the users' cooperation. The trained model (**step 7**) is then used for online sensing (**step 8**). As mentioned in [53], the MU-MIMO sounding procedure should be performed at least every 10 ms, which corresponds to 100 BFI measurements/second. Since the frequency of channel sounding is not specified in the standard and since the sounding measurement lasts approximately 500 microseconds, *the BFI rate can theoretically reach 2000 BFI per second.*

Example. Let assume the activity recording is 300 seconds long, and W is 0.1 seconds. Then, 3000 windows are present in the recording. Let us assume that the average number of packets in the considered windows is S = 10. The windows presenting less than 10 packets are zero-padded. Considering a bandwidth of 80 MHz, according to the IEEE 802.11 standard, four angles describe each of the K = 234 sub-channels where sounding is performed, i.e., the total number of OFDM sub-channels (256) minus pilots and control sub-channels that are excluded from the sounding procedure. Assuming that U = 3 users are connected to the AP, the resulting input tensor has dimensions $10 \times 234 \times 4 \times 3$, and presents a total size of $10 \cdot 234 \cdot 4 \cdot 3 = 28080$.

B. The FAMReS Classification Algorithm

Existing research in CSI-based sensing has exposed that designing classifiers that are robust to changing the subject performing the activity (i.e., different people) and the environment where the activity is performed (i.e., different rooms) is very challenging [43], [44], [13], [12]. On the other hand, it is hardly feasible to collect a large amount of data for all possible scenarios. To address this key issue, we propose a deep learning (DL)-based algorithm for BFI-based activity classification called *Fast and Adaptive Micro Reptile Sensing* (FAMReS), which is a few-shot learning (FSL) algorithm based on Reptile [54] which needs a limited set of new input data to generalize to unseen environments.

FSL is a DL technique that leverages only small amounts of additional data to adapt to classes that are unseen at training time. Specifically, in K-way-N-shot FSL, the model is trained on a set of mini-batches of data that only have K different classes (ways) and N samples (shots) of each class. The key

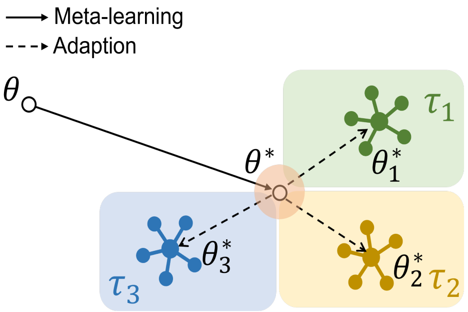


Fig. 5: Example of Few-Shot Learning.

idea is that by feeding less data, the model is spurred to rapidly adapt to new tasks. This unique property makes FSL a strong candidate to tackle the diversity of environments. FSL can be categorized into embedding learning [55], [56], and meta-learning [54], [57], among others. Specifically, Reptile is a gradient-based meta-learning algorithm that learns the model parameter initialization for rapid fine-tuning. The key idea is that there are some common features between different tasks that can be learned through meta-learning. Therefore, the model can be fine-tuned on a new task faster with the meta-learned weights instead of training it from the beginning. To find the initialization weights θ^*, Reptile minimizes the expectation of the loss function L_τ with respect to the different tasks τ, i.e.,

$$\theta^* = \min_\theta \quad \mathbb{E}_\tau \left\{ L_\tau \left[f\left(x, y|\theta\right)\right]\right\}, \tag{6}$$

where $f(x, y|\theta)$ is the model functional approximation between input data x and output y obtained with parameters θ. This is equivalent to finding the θ^* that satisfies $\mathbb{E}_\tau \left\{\nabla_\theta \left(L_\tau \left[f\left(x, y|\theta\right)\right]\right)\right\} = 0$ via, e.g., stochastic gradient descent (SGD). SGD finds θ^* through an iterative procedure, by subsequently updating the value of θ with a new value θ' based on the gradient information:

$$\theta' = \theta - \beta \frac{1}{n} \sum_{\tau=1}^{n} \left(\frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \left(L_\tau \left[f\left(x_i, y_i|\theta\right)\right]\right)\right) \tag{7}$$

$$= \theta - \beta \frac{1}{n} \sum_{\tau=1}^{n} \left(\theta - \tilde{\theta}\right), \tag{8}$$

where n and m denote the number of tasks and sampled data points of each task, respectively, β is a scalar denoting the step size, and $\tilde{\theta} = \theta - \alpha \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \left(L_\tau \left[f\left(x_i, y_i|\theta\right)\right]\right)$ are the updated weights using m sampled data from τ, where α denotes the learning rate. $\tilde{\theta}$ can be easily obtained using any deep learning API such as TensorFlow and PyTorch. The meta-learning proceeds through the following steps: (i) sample n new tasks $\{\tau\}$ with m data of each task (for K-way-N-shot, m is the product of K and N); (ii) compute $\tilde{\theta}$; (iii) update θ with Equation 8; (iv) iterate (ii) and (iii) until the loss function stops decreasing. Figure 5 shows how FSL is implemented through the Reptile algorithm: once obtained the initialization weights θ^* through meta-learning, the model is fine-tuned on each different task.

1) FAMReS Algorithm: The original purpose of Reptile is to extract meta-features from a large dataset so that it can be quickly fine-turned when a new task is sampled from the given dataset. However, *Reptile requires the inference and meta-learning data to be sampled from the same dataset*. Such a dataset should contain as many classes as possible so that the meta-learner can extract the general characteristics and fine-tune a task with fewer classes. Since this is unfeasible in BFI-based sensing, we find some common ground between meta-learning and general DL. The aim of learning is trying to approach the ground truth between different sampled data, while meta-learning is to find shared features between various tasks. Thus, if we consider each batch of training data as a new task in meta-learning, *the learning problem can be converted into a meta-learning problem*. Formally, we aim to find a set of parameters θ^* that minimize the loss function L on training data x_i and y_i:

$$\theta^* = \min_\theta \quad \mathbb{E}_i \left\{ L \left[f\left(x_i, y_i|\theta\right)\right]\right\}. \tag{9}$$

By plugging the derivative $\mathbb{E}_i \left\{\nabla_\theta \left(L \left[f\left(x_i, y_i|\theta\right)\right]\right)\right\}$ to the SGD optimizer, the optimization problem can be solved as

$$\tilde{\theta} = \theta - \alpha \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \left(L \left[f\left(x_i, y_i|\theta\right)\right]\right). \tag{10}$$

By comparing Equation 7 with 10, we can easily find that if we set $n = 1$ in Equation 7, the only difference between these two equations is a constant scalar. Based on this observation, we note that Reptile learns common ground from different mini-batch of data. The meta-learning rate β, which is usually a scalar less than 1, is to adjust the step size of the learning, making it less likely to overfit the mini-batch data. This meta-learning process can be regarded as a warm-up phase before learning, which makes the parameters θ closer to the ground truth in the hyperspace than random initial weights.

Inspired by this idea, FAMReS is divided into two stages: (i) meta-learning stage; and (ii) micro-learning stage. In stage (i), the model utilizes a small portion of data to learn the shared features. In stage (ii), the same micro dataset is used for training. The complete FAMReS workflow is reported in Algorithm 2. **We stress the difference between the original Reptile and FAMReS**: we only use a small portion of data in meta-learning and micro-learning and use other unseen data for testing. On the contrary, Reptile uses the same dataset for both learning and inference. Although we have only done experiments offline in this work, FAMReS is a strong candidate for online learning. The algorithm can run the meta-learning phase while collecting new data. Once there is enough data, it can move on to the next stage. Therefore, we define a time variable δ in experiments to simulate the real-time implementation. We use the data collected within the δ time window for learning and the other for inference. FAMReS is an empirical risk minimizer that can be unstable when using small values for δ, depending on the distribution of training data. Meta-learning on the micro dataset can only bring the initial parameters closer to the ground truth point in the hyperspace, but the final parameters still depend on the training set. Thanks to the high stability of the BFI data, we can always get a reasonable accuracy in the experiments unless δ is extremely small.

Algorithm 2: The FAMReS Algorithm

```
Require: step size $\beta$, micro dataset $\mathbb{D}$;
Initialize: a set of parameters $\theta$;
for $iteration = 1, 2, ...$ do
    sample k points of data from $\mathbb{D}$ ;  /*stage i*/
    compute $\tilde{\theta}$ using the SGD formulation;
    update the parameters: $\theta \leftarrow \theta + \beta \left( \tilde{\theta} - \theta \right)$;
for $epoch = 1, 2, ...$ do
    update $\theta$ running SGD on $\mathbb{D}$;     /*stage ii*/
```

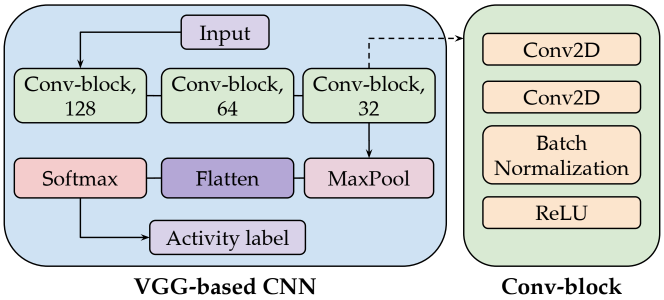


Fig. 6: Learning-based activity classifier.

2) Learning Architecture: In the last decade, convolutional neural networks (CNNs) have achieved tremendous success in computer vision [58], [59], [60]. The convolution layer, the basis of CNNs, can efficiently extract features by performing convolution operations on the elements of the input data. Given that in this article our aim is to investigate the effectiveness of BFI-based sensing as compared to CSI-based sensing, we propose to use a VGG-based [59] CNN architecture as the human activity classifier. The network is depicted in Figure 6 and entails stacking three convolutional blocks (`conv-block`) and a max-pooling (`MaxPool`) layer. Softmax is applied to the flattened output to obtain the probability distribution over the activity labels.

The `conv-block` is a stack of two convolution two-dimensional (2D) layers. Following the design of VGG [59], each convolution layer has a kernel size of 3×3 and a step size of 1. To introduce non-linearity in the model, we apply a rectified linear units (ReLU) activation function at the end of each `conv-block`. Batch normalization is also used in `conv-blocks` to avoid gradient explosion or vanishing. Our VGG-based CNN consists of three `conv-blocks` with 128, 64 and 32 filters, respectively. We choose a descending order of filters to reduce the model size since features in lower layers are usually sparser and thus require extracting more activation maps to be properly captured.

IV. Performance Evaluation

A. Experimental Setup and Data Collection

We collected experimental data in three environments: a kitchen, a living room, and a classroom, as depicted in Figure 7. We considered three human subjects and twenty different activities: *jogging, clapping, push forward, boxing, writing, brushing teeth, rotating, standing, eating, reading a book, waiving, walking, browsing phone, drinking, hands-up-down, phone call, side bend, check the wrist (watch), washing hands, and browsing laptop*. The activities are performed independently by each subject within a designated rectangular region in each of the three environments. Both BFI and CSI data is collected for the same duration of 300 seconds for each of the twenty activities. **To create the ground truth, we captured the synchronous video streams of the subjects performing the activity**. The video streams are synchronized with the data to show what the subject is doing during the transmission of the NDP frame triggering the BFI computation. As an example, three frames from the captured video streams are shown in Figure 8.

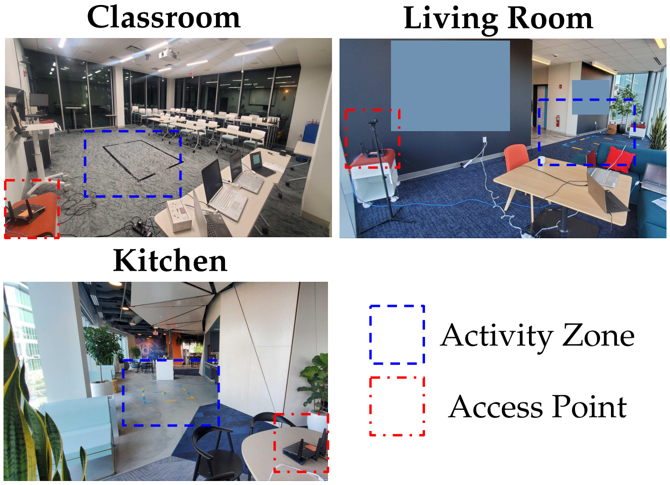


Fig. 7: Sites of experimental data collection.

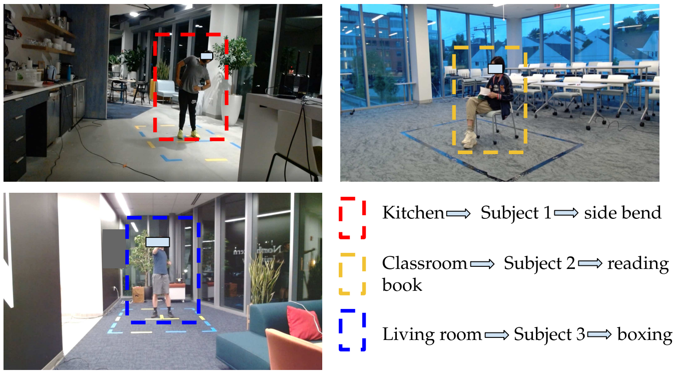
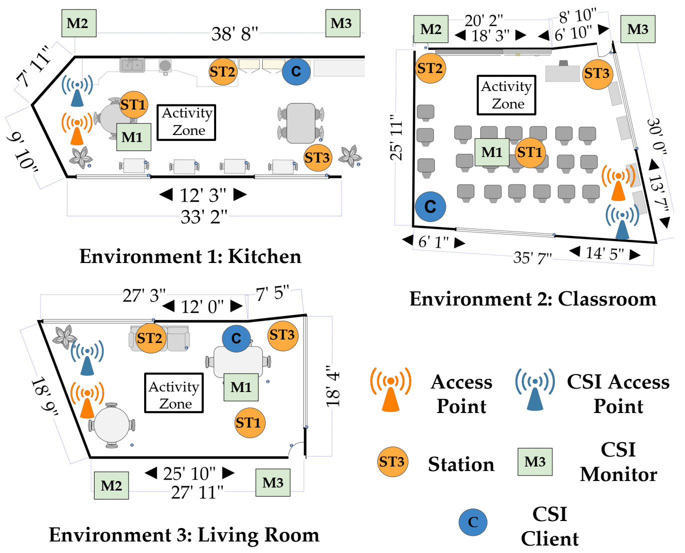

Kitchen⟹ Subject 1⟹ side bend
Classroom⟹ Subject 2⟹ reading book
Living room⟹ Subject 3⟹ boxing

Fig. 8: Sample frames from the video capture.

MU-MIMO Setup and Equipment. We set up an 802.11ac MU-MIMO network operating on channel 153 with center frequency f_c=5.77 GHz and 80 MHz bandwidth. This allows sounding K=234 sub-channels, i.e., 256 available sub-channels on 80 MHz channels minus 14 control sub-channels and 8 pilots. We use one AP (beamformer) and three STAs (beamformees), as depicted in Figure 9 in orange. The AP and the STAs are implemented through Netgear Nighthawk X4S AC2600 routers with M=3 and N=1 antennas enabled respectively for the AP and each of the STAs. The three STAs are served with $N_{ss} = 1$ spatial stream each and placed at three different heights and significantly spaced from each other to form a 3×3 MU-MIMO system. According to the IEEE 802.11ac standard, four beamforming feedback angles (two

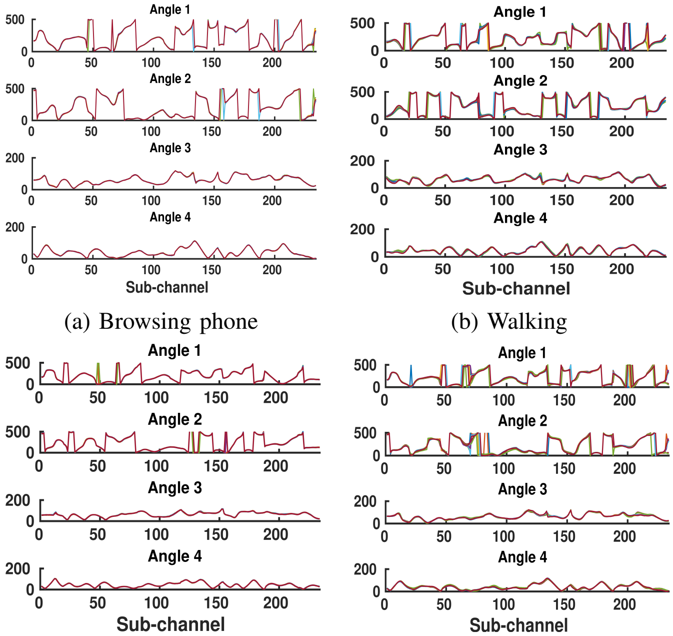


Environment 1: Kitchen

Environment 2: Classroom

Environment 3: Living Room

Fig. 9: Experimental setups for data collection.



(a) Browsing phone

(b) Walking

(c) Drinking water

(d) Boxing

Fig. 10: BFI angles for each sub-channel for four activities. Each plot shows the values of 10 different packets (superimposed lines with different colors). The x-axis reports the indices of the sensed sub-channels.

ϕ and two ψ) are needed to represent each of the 3×1 channels between the AP and the STAs. In our setup, the angle quantization process uses b_ϕ= 9 bits and b_ψ =7 bits for the feedback angles ϕ and ψ respectively. UDP data streams are sent from the AP to the STAs in the downlink direction to trigger the channel sounding. The BFI frames are captured with the Wireshark network protocol analyzer running on an off-the-shelf laptop equipped with an Intel 9560NGW wireless-AC NIC set in monitor mode. However, note that any IEEE 802.11ac-compliant NIC set in monitor mode could be used for this purpose. Moreover, notice that the frame-capturing device does not need any direct link with the AP or the STAs. The only requirement is that the capture is performed on the wireless channel where the Wi-Fi network is operating. From the captured frames, the ϕ and the ψ angles are extracted for each of the STAs and used as input to the `BeamSense` learning framework (see Section III-B). Figure 10 shows a sample taken from our dataset. We plot the magnitude of the four collected beamforming angles for each of the 234 available sub-channels, for ten different packets and four activities. Figure 10 remarks that the absolute values of the angles change quite significantly among different activities, while do not change significantly among different packets. This indicates that BFI-based sensing is a stable measurement of the channel propagation environment and thus, a strong candidate to be used within Wi-Fi sensing systems.

CSI Network Setup and Equipment. For comparative studies, CSI data has also been collected concurrently with the BFI frame capture. For this purpose, a Wi-Fi network consisting of an AP (referred to as *CSI AP*) and a single STA (referred to as *CSI client*) has been set up within the same environments, as depicted in Figure 9 in blue. The network operates on the IEEE 802.11ac channel 42, i.e., the center frequency is f_c = 5.21 GHz and the bandwidth is 80 MHz. The AP is implemented with a Netgear Nighthawk X4S AC2600 router, while the CSI client is a PC APU2 board equipped with an Intel 9560NGW wireless-AC NIC. For the CSI extraction, three IEEE 802.11ac-compliant Asus RT-AC86U routers (referred to as *CSI monitors*) equipped with the Nexmon CSI extraction tool ([8]) have been deployed, as depicted in Figure 9 in green. To have the same setup as in the MU-MIMO network, the CSI AP is enabled with M = 3 antennas whereas the CSI monitors are set up to sense the channel through N = 1 antenna over N_{ss}= 1 spatial stream each. UDP packets are sent from the CSI AP to the CSI client to trigger the channel estimation on the three CSI monitors.

Note that, as shown in Figure 9, the CSI AP and one of the CSI monitors (M1) are respectively placed at the same location as the MU-MIMO AP and one of the stations (ST1) to allow for baseline performance comparison. To show the challenges of using CSI-based sensing, we place both the BFI capturing device and the CSI monitors M2 and M3 beyond the wall of the activity zone. *The CSI monitor captures the channel between itself and the CSI AP*, and, in turn, the performance decrease when CSI collectors are placed far from the monitored environment, as detailed in Section IV-B1.

B. Performance Analysis

In the following, all the results are obtained with a time window size of 0.1 s with ten packets/sample with the data of three subjects combined, unless specified otherwise.

1) Comparison between BFI and CSI-based Sensing: Figure 11 shows the classification accuracy of `BeamSense` as compared to the state-of-the-art CSI-based SignFi algorithm [61] in the three environments. For a baseline comparison, we only consider M1 and ST1 as the CSI collection device and BFI STA respectively which are co-located. We first

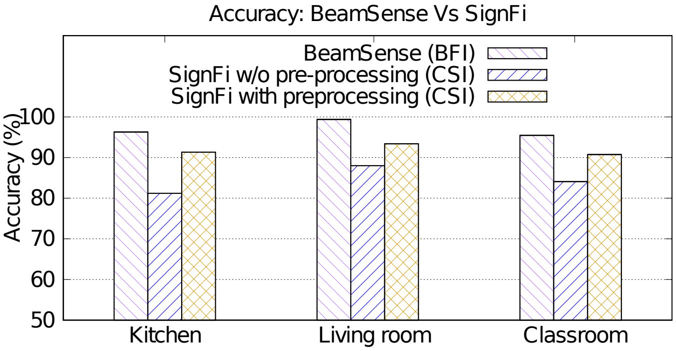


Fig. 11: BeamSense (BFI) vs SignFi (CSI) performance.

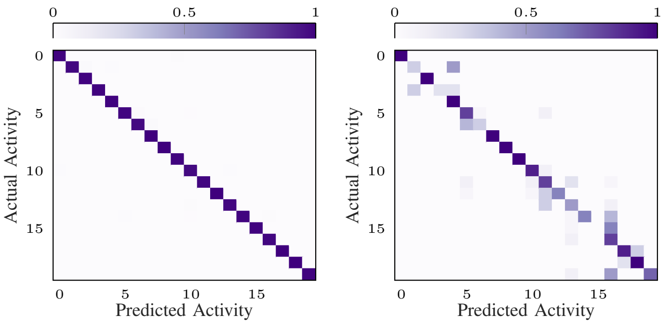


Fig. 12: Conf. matrices for BeamSense and SignFi.

evaluate the performance of BFI and CSI-based sensing using the minimalist data processing and the CNN architecture as referenced in Figure 4 and Figure 6 respectively. The accuracy of BeamSense in the kitchen, living room and classroom is respectively 96%, 99%, and 95.47% whereas SignFi reaches 81.19%, 87.99%, and 84.08% of accuracy respectively, resulting in a 12.6% accuracy decrease on average. We also show the performance of SignFi with the processing pipeline presented in [61], which unwraps the phase of each collected signal and then removes the phase noise by multiple linear regression based on the unwrapped phase across all sub-carriers and antennas. The classification accuracy improves to 91.34%, 93%, and 90% in the kitchen, living room, and classroom environments, respectively. **Yet, BeamSense achieves better performance with no data preprocessing.**

To shed light on which classes are the hardest to classify with CSI-based sensing, Figure 12 shows the confusion matrices obtained in the kitchen using BeamSense and SignFi without the custom pre-processing. The bottom five classes are browsing laptop (index 20), phone call (16), hands-up-down (15), clapping (02), and boxing (04), which are indeed among the hardest classes to distinguish.

Figure 13 shows the performance of BeamSense and SignFi with pre-processing evaluated in the kitchen as a function of the CSI capture location, the BFI capture location, and the window size W. Whereas CSI data acquired through M1 provides acceptable results since M1 is very close to the activity zone, data acquired with M2 and M3 provides poor results as M2 and M3 are far from the activity zone and beyond a wall. Specifically, the accuracy drops by 94.13% considering an observation window size of W = 0.1 s. On the contrary, the performance of BeamSense does not change with the location of the BFI collector. Moreover, in the case of CSI-based sensing, we also observe a significant performance variation when varying the window size W. For SignFi, with the variation of the window size, the accuracy varies by 47.37% on average, which is only 1.36% for BeamSense. **This proves that feedback angles are a much more stable and reliable measurement than CSI.**

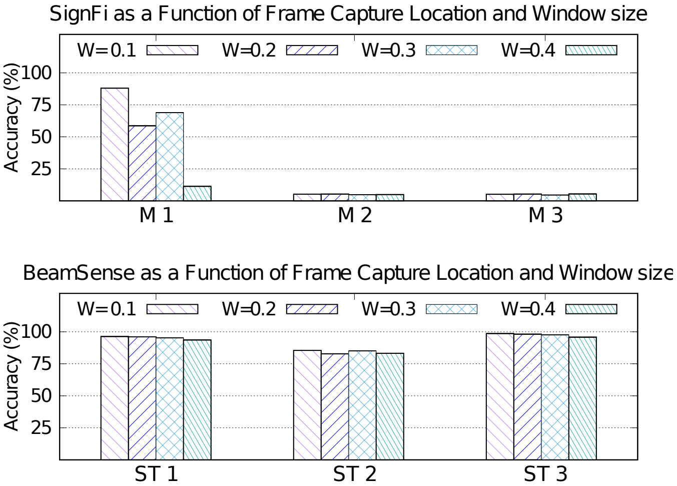


Fig. 13: BeamSense and SignFi performance changing the capture location and the window size.

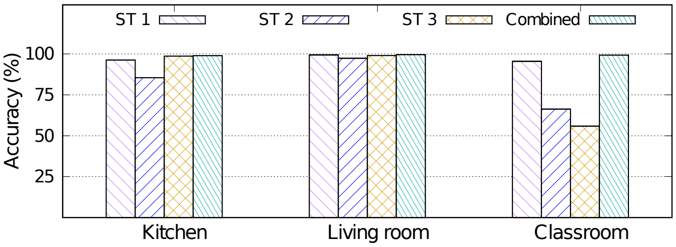
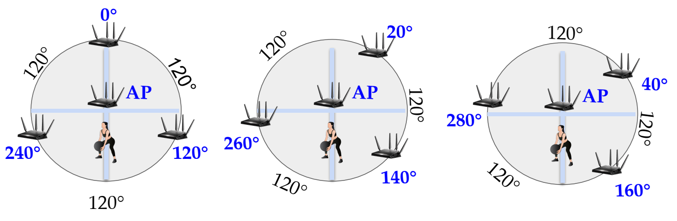


Fig. 14: Impact of the spatial diversity.

2) Performance as a Function of the Spatial Diversity: Figure 14 presents the performance of BeamSense when trained with data from a single STA and with the combined data. First, we notice that the single STA data is almost always a very stable measurement, with the accuracy remaining high in most of cases. However, we notice that some STAs perform worse than others, especially ST2 in the kitchen, and ST2 and ST3 in the classroom. Indeed, due to the physical location of these STAs, the communication channels between them and the AP might be in deep fade causing BeamSense to perform poorly. However, by aggregating the spatially diverse STA data, **the overall accuracy is improved by up to 43.81%** in the classroom. Given the variability of the Wi-Fi channel, considering different STA locations imply obtaining completely different angles for the same activity, even in the same environment, as shown in Figure 14. To further investigate the sensing performance as a function of the STA location, we conduct an experiment in the kitchen entailing three different STA locations as depicted in Figure 15. The first placement is referred to as "Setup 1" while "Setup 2" and "Setup 3" are obtained by physically rotating each STA by 20°clockwise, which corresponds to placing the STA



Setup 1 Setup 2 Setup 3

Fig. 15: Different setup / orientation of the STAs.

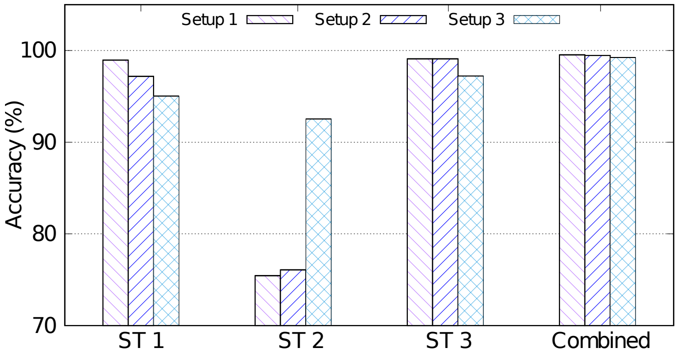


Fig. 16: BeamSense Accuracy vs STAs Location.

around 2 meters away from the previous location. Figure 16 shows the accuracy of BeamSense in the kitchen when using data collected through each of the three setups. BeamSense performs very well when combining all the STAs: the accuracy is 99.53%, 99.46%, and 99.23% respectively in Setup 1, Setup 2 and Setup 3. Therefore, **multi-STA sensing should be preferred over single-STA sensing whenever possible**.

3) Evaluation of Angle and Sub-Channel Resolution: It is known that Wi-Fi sensing performs worse when lowering the number of sub-channel considered in the sensing process [62], [31]. Extensive feature extraction or higher sampling frequency can be utilized, at the cost of increasing the computational burden and intensifying pre-processing steps, as well as increasing the computational complexity of the learning process. For this reason, we investigate the trade-off between the number of angles and sub-channels considered for sensing and the sensing performance.

Figure 17 shows the accuracy of BeamSense as a function of the number of sub-channels utilized in the learning process. To down-sample the sub-channels, we take the first 20, 40, 80, and 160 sub-channels, to emulate sensing systems with smaller available bandwidths. As expected, the accuracy decreases by 6.31%, 3.80%, and 3.46% respectively for the kitchen, living room, and classroom when we switch from 234 to 20 sub-channels. However, notice that this operation drastically decreases the input tensor dimension from $10 \times 234 \times 12 = 28080$ to $10 \times 20 \times 12 = 2400$, implying that sub-channel resolution decreases the computational burden by 10× while maintaining the accuracy above 92% in all the considered scenarios.

Figure 18 shows BeamSense performance as a function of the number of angles considered for sensing. STA1 is considered for angle 1, angle 2, angle 3, angle 4, and the combination of four angles, whereas STA1 and STA2 are considered for the combination of eight angles, and all three stations are considered for the combination of 12 angles. Figure 18 shows that the accuracy decreases by 1.98%, 0.16%, and 2.22% in the kitchen, living room and classroom respectively when considering a single angle with respect to the combination of 12 angles. Even though the above results show no significant variation in performance even if the angle resolution is decreased from 12 angles combined to any individual angle, we suggest aggregating at least the angles of two spatially diverse STAs to obtain a robust algorithm.

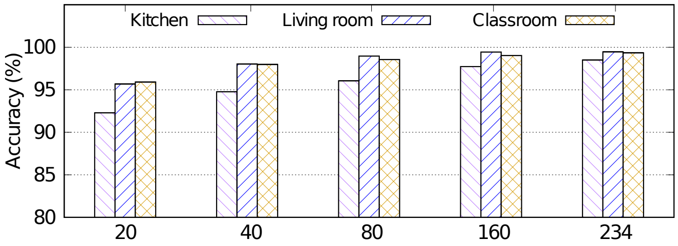


Fig. 17: BeamSense accuracy as a function of the number of sensed sub-channels.

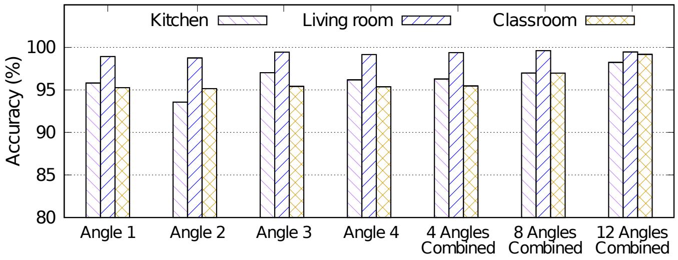


Fig. 18: BeamSense accuracy as a function of the number of the angles considered.

4) Evaluation of CNN Filter Size: To further investigate the trade-off between computation complexity and accuracy, we introduce a width multiplier $\alpha \in (0, 1]$ to each layer of the CNN-based classifier. For a given number of input channels C and output channels Z, they become αC and αZ after applying the multiplier. Hence, the computation complexity will be reduced by α^2 roughly. Applying the width multiplier α to BeamSense, the channel size of each conv-block becomes $\alpha \times 128$, $\alpha \times 64$, $\alpha \times 32$, respectively. Figure 19 shows how the accuracy changes when applying width multiplier $\alpha \in \{0.07, 0.13, 0.25, 0.5, 0.75\}$. BeamSense accuracy, averaged over the three environments, is 97.22%, 98.01%, 98.62%, 98.88%, and 99.02%, respectively. **As the CNN width decreases from 0.75 to 0.07, the accuracy drops marginally by 1.8%**. This observation indicates that BeamSense can adapt to limited computation resources and latency-sensitive cases by sacrificing little accuracy.

C. Evaluation of FAMReS Algorithm

To address the challenge of generalization to unseen environments and subjects, we have proposed FAMReS in Section III-B1. We compare the performance of FAMReS with the state-of-the-art FSL algorithm OneFi[44] and the transfer learning (TL) algorithm presented in WiTransfer[63] for cross-domain WiFi sensing. Figure 20(a) shows that with only 15 s of new data, FAMReS can adapt to new environments with

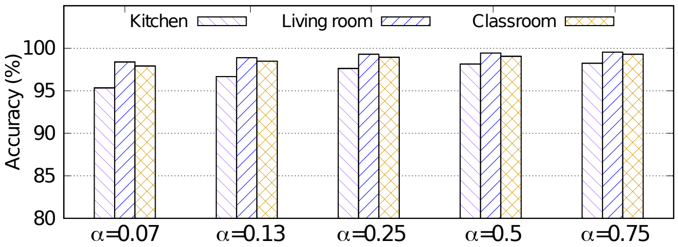
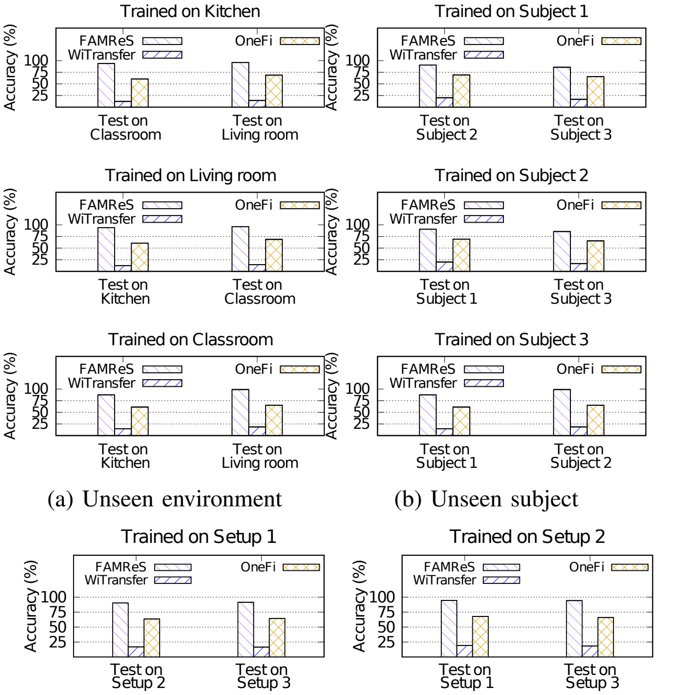


Fig. 19: `BeamSense` accuracy as a function of the number of the CNN filter sizes.

an average accuracy of 94.97%, 90.51% and 93.09% when trained in the kitchen, living room, and classroom respectively. On the other hand, WiTransfer achieves 13.4%, 18.02%, and 16.52% respectively, which is 76.88% less than FAMReS. The reason is that the WiTransfer pre-trained model is optimized for a specific task. Conversely, transfer learning approaches usually require more data to get rid of the data bias and 15s of new data is not enough for WiTransfer to achieve satisfactory accuracy. OneFi achieves an accuracy of 64.72%, 63.36%, and 63.24% respectively in the kitchen, living room, and classroom. Although it can generalize to new environments to some extent, FAMReS performs better since it can fine-tune the whole model and learn shared information across different tasks by meta-learning. On the contrary, OneFi utilizes information from one task and only fine-tunes the classifier. Figure 20(b) shows a similar trend, where FAMReS is 73.41% better than WiTransfer and 24.81% better when compared to OneFi on average. We also evaluated the performance of FAMReS as a function of different setups as discussed in Section IV-B2. Figure 20(c) shows that FAMReS achieves an accuracy of 90.93%, 94.38%, and 93.20% when trained in setup 1, setup 2, and setup 3 respectively, and tested in the other setups. FAMReS supersedes WiTransfer and OneFi by 74.88% and 27.28% respectively with new unseen setups too.

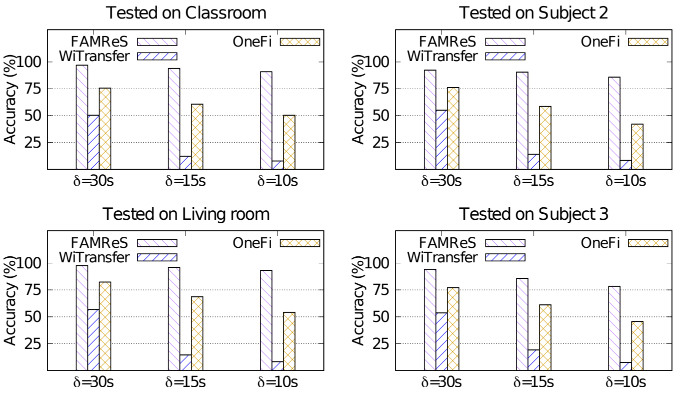
Finally, we investigate the performance of FAMReS as a function of the additional micro-dataset δ required to generalize to new environments and/or subjects. Figure 21 shows the performance of the different considered sensing algorithms as a function of the micro dataset size δ. The results show that as δ decreases from 30 s to 10 s, the accuracy of FAMReS only drops by 5.30% and 11.13% on average when tested in unseen environments and subjects respectively. On the contrary, the performance of WiTransfer drops significantly when the duration of micro dataset δ is reduced to 10 s, showing that without the meta-learning phase, transfer learning requires more data for adaptation. Although OneFi is more stable than WiTransfer, the accuracy decreases to only 52.26% and 43.92% respectively with unseen environments and subjects, which is 39% less than FAMReS. This proves the performance gain that FAMReS achieves by fine-tuning the whole network rather than fine-tuning only the classifier like OneFi.



(a) Unseen environment (b) Unseen subject

(c) Unseen orientation

Fig. 20: Comparative analysis of `BeamSense` in unseen environments, subjects and orientations.



(a) Unseen environment (b) Unseen subject

Fig. 21: Comparative analysis of `BeamSense` as a function of micro dataset, δ.

V. Conclusions and Remarks

In this article, we have proposed `BeamSense`, a novel approach to Wi-Fi sensing based on the usage of MU-MIMO beamforming feedback information (BFI). Conversely from CSI-based approaches, (i) the BFI can be easily recorded by off-the-shelf devices without MIMO capabilities and without any firmware modification; (ii) the BFI captures in a single packet the multiple channels between the AP and the STAs, thus achieving a much better sensing granularity. `BeamSense` includes a few-shot learning (FSL)-based classification algorithm to adapt to new environments and subjects with few additional data. We have evaluated `BeamSense` through an extensive data collection campaign involving three subjects performing twenty different activities in three indoor environments. We have compared our approach with traditional CSI-based sensing approaches and show that `BeamSense` improves the accuracy by 25% on the average, while our FSL-based approach improves accuracy by up to 51% when compared with state-of-the-art domain adaptive sensing models. We hope that this work will pave the way for additional research on BFI-based Wi-Fi sensing.

References

[1] Wi-Fi Alliance, "The Economic Value of Wi-Fi: A Global View (2018 and 2023)," https://tinyurl.com/EconWiFi, 2021.

[2] Y. Ma, S. Arshad, S. Muniraju, E. Torkildson, E. Rantala, K. Doppler, and G. Zhou, "Location- and Person-Independent Activity Recognition with WiFi, Deep Neural Networks, and Reinforcement Learning," *ACM Trans. Internet of Things*, vol. 2, no. 1, Jan. 2021.

[3] X. Wang, C. Yang, and S. Mao, "TensorBeat: Tensor Decomposition for Monitoring Multiperson Breathing Beats with Commodity WiFi," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 9, no. 1, pp. 1–27, 2017.

[4] 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 Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1090–1103, 2017.

[5] 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.

[6] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool Release: Gathering 802.11n Traces with Channel State Information," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, p. 53, 2011.

[7] Y. Xie, Z. Li, and M. Li, "Precise Power Delay Profiling with Commodity Wi-Fi," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, 2015.

[8] F. Gringoli, M. Schulz, J. Link, and M. Hollick, "Free Your CSI: A Channel State Information Extraction Platform For Modern Wi-Fi Chipsets," in *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*. New York, NY, USA: Association for Computing Machinery, 2019, p. 21–28.

[9] Z. Jiang, T. H. Luan, X. Ren, D. Lv, H. Hao, J. Wang, K. Zhao, W. Xi, Y. Xu, and R. Li, "Eliminating the Barriers: Demystifying Wi-Fi Baseband Design and Introducing the PicoScenes Wi-Fi Sensing Platform," *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4476–4496, 2022.

[10] F. Gringoli, M. Cominelli, A. Blanco, and J. Widmer, "AX-CSI: Enabling CSI Extraction on Commercial 802.11ax Wi-Fi Platforms," in *Proceedings of the 15th ACM Workshop on Wireless Network Testbeds, Experimental Evaluation & CHaracterization*. New York, NY, USA: Association for Computing Machinery, 2022, p. 46–53.

[11] E. Aryafar, N. Anand, T. Salonidis, and E. W. Knightly, "Design and Experimental Evaluation of Multi-User Beamforming in Wireless LANs," in *Proc. of the 16th Annual International Conference on Mobile Computing and Networking (MobiCom)*, New York, NY, USA, 2010.

[12] N. Bahadori, J. Ashdown, and F. Restuccia, "ReWiS: Reliable Wi-Fi Sensing Through Few-Shot Multi-Antenna Multi-Receiver CSI Learning," in *Proceedings of the IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Los Alamitos, CA, USA, jun 2022.

[13] F. Meneghello, D. Garlisi, N. Dal Fabbro, I. Tinnirello, and M. Rossi, "SHARP: Environment and Person Independent Activity Recognition with Commodity IEEE 802.11 Access Points," *IEEE Transactions on Mobile Computing*, pp. 1–16, 2022.

[14] 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.

[15] C.-F. Hsieh, Y.-C. Chen, C.-Y. Hsieh, and M.-L. Ku, "Device-free indoor human activity recognition using Wi-Fi RSSI: machine learning approaches," in *2020 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan)*. IEEE, 2020, pp. 1–2.

[16] 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.

[17] M. Zhang, Z. Fan, R. Shibasaki, and X. Song, "Domain Adversarial Graph Convolutional Network Based on RSSI and Crowdsensing for Indoor Localization," *arXiv preprint arXiv:2204.05184*, 2022.

[18] S. Depatla and Y. Mostofi, "Crowd Counting through Walls Using WiFi," in *Proceedings of the IEEE international conference on pervasive computing and communications (PerCom)*, Athens, Greece, 2018.

[19] P. Ssekidde, O. Steven Eyobu, D. S. Han, and T. J. Oyana, "Augmented CWT features for deep learning-based indoor localization using WiFi RSSI data," *Applied Sciences*, vol. 11, no. 4, p. 1806, 2021.

[20] N. Singh, S. Choe, and R. Punmiya, "Machine learning based indoor localization using Wi-Fi RSSI fingerprints: an overview," *IEEE Access*, 2021.

[21] W. Li, M. J. Bocus, C. Tang, S. Vishwakarma, R. J. Piechocki, K. Woodbridge, and K. Chetty, "A Taxonomy of WiFi Sensing: CSI vs passive Wi-Fi Radar," in *2020 IEEE Globecom Workshops (GC Wkshps*. IEEE, 2020, pp. 1–6.

[22] W. Li, R. J. Piechocki, K. Woodbridge, C. Tang, and K. Chetty, "Passive WiFi Radar for Human Sensing Using a Stand-alone Access Point," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 3, pp. 1986–1998, 2020.

[23] C. Tang, W. Li, S. Vishwakarma, F. Shi, S. Julier, and K. Chetty, "People counting using multistatic passive WiFi radar with a multi-input deep convolutional neural network," in *Radar Sensor Technology XXVI*. SPIE, 2022.

[24] C. Tang, W. Li, S. Vishwakarma, K. Chetty, S. Julier, and K. Woodbridge, "Occupancy detection and people counting using WiFi passive radar," in *2020 IEEE Radar Conference (RadarConf20)*. IEEE, 2020, pp. 1–6.

[25] B. Huang, G. Mao, Y. Qin, and Y. Wei, "Pedestrian flow estimation through passive wifi sensing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1529–1542, 2019.

[26] Q. Bu, X. Ming, J. Hu, T. Zhang, J. Feng, and J. Zhang, "TransferSense: towards environment independent and one-shot wifi sensing," *Personal and Ubiquitous Computing*, vol. 26, no. 3, pp. 555–573, 2022.

[27] B. Korany, H. Cai, and Y. Mostofi, "Multiple People Identification Through Walls Using Off-the-Shelf WiFi," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6963–6974, 2021.

[28] Y. Zeng, P. H. Pathak, and P. Mohapatra, "WiWho: WiFi-based Person Identification in Smart Spaces," in *Proceedings of ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2016, pp. 1–12.

[29] E. Soltanaghaei, R. A. Sharma, Z. Wang, A. Chittilappilly, A. Luong, E. Giler, K. Hall, S. Elias, and A. Rowe, "Robust and practical WiFi human sensing using on-device learning with a domain adaptive model," in *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, 2020, pp. 150–159.

[30] S. Liu, Y. Zhao, F. Xue, B. Chen, and X. Chen, "DeepCount: Crowd counting with WiFi via deep learning," *arXiv preprint arXiv:1903.05316*, 2019.

[31] Y. Zeng, D. Wu, J. Xiong, J. Liu, Z. Liu, and D. Zhang, "MultiSense: Enabling Multi-person Respiration Sensing with Commodity WiFi," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, vol. 4, no. 3, pp. 1–29, 2020.

[32] C. Shi, T. Zhao, Y. Xie, T. Zhang, Y. Wang, X. Guo, and Y. Chen, "Environment-independent In-baggage Object Identification Using WiFi Signals," in *Proceedings of IEEE International Conference on Mobile Ad Hoc and Smart Systems (MASS)*. IEEE, 2021.

[33] Y. Ren, S. Tan, L. Zhang, Z. Wang, Z. Wang, and J. Yang, "Liquid level sensing using commodity wifi in a smart home environment," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 1, pp. 1–30, 2020.

[34] Y. He, Y. Chen, Y. Hu, and B. Zeng, "WiFi vision: Sensing, recognition, and detection with commodity MIMO-OFDM WiFi," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8296–8317, 2020.

[35] Y. Ren, Z. Wang, S. Tan, Y. Chen, and J. Yang, "Winect: 3D Human Pose Tracking for Free-form Activity Using Commodity WiFi," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 4, pp. 1–29, 2021.

[36] ——, "Tracking Free-Form Activity Using WiFi Signals," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 816–818.

[37] W. Jiang, H. Xue, C. Miao, S. Wang, S. Lin, C. Tian, S. Murali, H. Hu, Z. Sun, and L. Su, "Towards 3D Human Pose Construction Using Wifi," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1–14.

[38] M. Zhao, T. Li, M. Abu Alsheikh, Y. Tian, H. Zhao, A. Torralba, and D. Katabi, "Through-wall human pose estimation using radio signals," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7356–7365.

[39] M. Muaaz, A. Chelli, M. W. Gerdes, and M. Pätzold, "Wi-Sense: A passive human activity recognition system using Wi-Fi and convolutional neural network and its integration in health information systems," *Annals of Telecommunications*, vol. 77, no. 3, pp. 163–175, 2022.

[40] Y. Ge, A. Taha, S. A. Shah, K. Dashtipour, S. Zhu, J. M. Cooper, Q. Abbasi, and M. Imran, "Contactless WiFi Sensing and Monitoring for Future Healthcare-Emerging Trends, Challenges and Opportunities," *IEEE Reviews in Biomedical Engineering*, 2022.

[41] B. Korany, C. R. Karanam, H. Cai, and Y. Mostofi, "Teaching RF to Sense without RF Training Measurements," in *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, no. 4, 2020.

[42] B. Wei, W. Hu, M. Yang, and C. T. Chou, "From Real to Complex: Enhancing Radio-based Activity Recognition Using Complex-Valued CSI," *ACM Transactions on Sensor Networks (TOSN)*, vol. 15, no. 3, pp. 1–32, 2019.

[43] 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 ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2019.

[44] R. Xiao, J. Liu, J. Han, and K. Ren, "OneFi: One-Shot Recognition for Unseen Gesture via COTS WiFi," in *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2021, pp. 206–219.

[45] H. F. T. Ahmed, H. Ahmad, and C. Aravind, "Device free human gesture recognition using Wi-Fi CSI: A survey," *Engineering Applications of Artificial Intelligence*, vol. 87, p. 103281, 2020.

[46] 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, 2020.

[47] I. Nirmal, A. Khamis, M. Hassan, W. Hu, and X. Zhu, "Deep Learning for Radio-Based Human Sensing: Recent Advances and Future Directions," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 995–1019, 2021.

[48] L. Guo, L. Wang, C. Lin, J. Liu, B. Lu, J. Fang, Z. Liu, Z. Shan, J. Yang, and S. Guo, "Wiar: A Public Dataset for WiFi-based Activity Recognition," *IEEE Access*, vol. 7, pp. 154 935–154 945, 2019.

[49] 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 (SenSys 2020)*. New York, NY, USA: Association for Computing Machinery, 2020, p. 517–530.

[50] B. Bloessl, M. Segata, C. Sommer, and F. Dressler, "An IEEE 802.11 a/g/p OFDM Receiver for GNU Radio," in *Proceedings of the second workshop on Software radio implementation forum*, 2013, pp. 9–16.

[51] X. Wang, K. Niu, J. Xiong, B. Qian, Z. Yao, T. Lou, and D. Zhang, "Placement Matters: Understanding the Effects of Device Placement for WiFi Sensing," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 1, pp. 1–25, 2022.

[52] E. Perahia and R. Stacey, *Next Generation Wireless LANs: Throughput, Robustness, and Reliability in 802.11n*. Cambridge Univ. Press, 2008.

[53] M. S. Gast, *802.11 ac: A Survival Guide: Wi-Fi at Gigabit and Beyond*. " O'Reilly Media, Inc.", 2013.

[54] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *arXiv preprint arXiv:1803.02999*, 2018.

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

[56] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," *Advances in neural information processing systems*, vol. 29, 2016.

[57] 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.

[58] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[59] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[60] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[61] Y. Ma, G. Zhou, S. Wang, H. Zhao, and W. Jung, "SignFi: Sign language recognition using WiFi," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 1, pp. 1–21, 2018.

[62] S. Shi, Y. Xie, M. Li, A. X. Liu, and J. Zhao, "Synthesizing wider WiFi bandwidth for respiration rate monitoring in dynamic environments," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 181–189.

[63] Y. Fang, B. Sheng, H. Wang, and F. Xiao, "WiTransfer: A cross-scene transfer activity recognition system using WiFi," in *Proceedings of the ACM Turing Celebration Conference-China*, 2020, pp. 59–63.