Efficient One-shot Gesture Recognition for WiFi ISAC via Aug-meta Learning

Jianwei Liu, Jiantao Yuan, *Member, IEEE,* Guanding Yu, *Senior Member, IEEE,* Jinsong Han, *Senior Member, IEEE*

***Abstract*—WiFi-based gesture recognition (WGR) has emerged as a promising technology due to its potential for integration with communication systems under the concept of integrated sensing and communication (ISAC). However, current WGR systems face two primary challenges: limited scalability for recognizing new gestures and poor compatibility with ISAC. These systems typically require extensive data collection and retraining for each new gesture and struggle to handle the dimensional variability of channel state information (CSI) caused by fluctuating data traffic in communication networks. To overcome these limitations, we introduce OneSense, a one-shot WGR system designed for seamless integration with communication systems. OneSense designs a data enrichment technique based on the law of signal propagation to generate virtual gestures. Based on enriched dataset, OneSense leverages an aug-meta learning (AML) framework to facilitate efficient and scalable FSL. OneSense also incorporates a data cropping strategy to enhance gesture feature prominence and a dynamic size-adaptive backbone model that ensures compatibility with CSI samples exhibiting dimensional inconsistencies. Experimental results show that OneSense achieves over 94% accuracy in one-shot gesture recognition. A case study further illustrates its effectiveness in ISAC contexts. Furthermore, our proposed AML framework reduces pre-training latency by more than 86% compared to conventional meta-learning approaches.**

***Index Terms*—WiFi, Gesture Recognition, Few-shot Learning, Integrated Sensing and Communication.**

I. Introduction

IN the past decade, WiFi-based gesture recognition (WGR) has garnered significant attention due to its multiple compelling properties, such as visual privacy preserving and device-free operation. Most importantly, WGR can be implemented by reusing existing WiFi communication infrastructures [2]–[5], making it highly promising for seamless integration into existing WiFi systems, also known as integrated sensing and communication (ISAC) [6]–[12], for wide and rapid deployment. Existing WGR approaches typically rely on deep learning models to associate channel state information (CSI) with gesture classes. While these methods demonstrate high gesture recognition accuracy, they suffer from two major flaws concerning system scalability and compatibility, which hinder their practical adoption in real-world communication contexts.

On the one hand, deep learning-based sensing models generally require each gesture class to provide dozens or even hundreds of samples per gesture class for training, making data collection significantly time-consuming and labor-intensive. Furthermore, traditional sensing models fix the gesture classes during the initial system implementation. To recognize new gestures, the model must be re-trained to adapt, which introduces massive extra computational overhead. As the demand for gesture categories continues to grow [13], these limitations make existing systems struggle to fulfill flexible gesture recognition tasks. On the other hand, previous works often require the input samples for the sensing model to have a consistent size. However, in communication scenarios, the packets are often irregular in time, and different gestures may vary in duration. Therefore, it is challenging to maintain a consistent input size. This inconsistency complicates their integration into communication networks. In addition, enforcing a unified size for different gesture classes fails to fully capture their unique characteristics and results in unnecessary computational costs.

Recently, few-shot learning (FSL) techniques have shown potential in addressing the aforementioned drawbacks. Previous works [14]–[16] in computer vision (CV) field have demonstrated the feasibility of few-shot image recognition through *meta-learning*. The core idea behind meta-learning is to pre-train a general model using a large number of similar few-shot tasks, enabling the model to quickly adapt to new tasks with only a limited number of labeled samples. However, applying meta-learning to WGR is non-trivial due to the following challenges. (1) Training a general model requires a dataset containing a wide range of gesture classes (referred to as seen classes). Collecting enough samples to create such a diverse dataset is time-consuming and labor-intensive. (2)

Manuscript received November 30, 2024; revised April 14, 2025; accepted May 27, 2025. This work is supported by the National Natural Science Foundation of China under grant No. U21A20462 and 62372400, 2025 Annual Scientific Research Cultivation Plan of Hangzhou City University-Special Project of Urban Development and Strategy Research Institute Research on the Construction of a Potential Risk Assessment and Early Warning System for Immovable Cultural Relics in Zhejiang During the Flood Season Project (Grant No. C-202409), Science and Technology Program for Emergency Management Research and Development by the Zhejiang Provincial Department of Emergency Management (Grant No. 2025YJ033), and the Postdoctoral Fellowship Program of CPSF under grant No. GZC20241488. Part of this paper was presented in INFOCOM'24 [1]. *(Corresponding authors: Jiantao Yuan and Jinsong Han.)*

J. Liu is with Zhejiang University, Hangzhou 310027, China, and also with the School of Information and Electrical Engineering, Hangzhou City University, Hangzhou 310015, China (e-mail: jianweiliu@zju.edu.cn).

J. Yuan is with the School of Information and Electrical Engineering, Hangzhou City University, Hangzhou 310015, China, and also with Zhengzhou Digital Industry Institute, Hangzhou City University, Zhengzhou 450046, China, and also with the Academy of Edge Intelligence, Hangzhou City University, Hangzhou 310015, China (e-mail: yuanjt@hzcu.edu.cn).

G. Yu is with the State Key Laboratory of Ocean Sensing, Zhoushan 316021, China, and also with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: yuguanding@zju.edu.cn).

J. Han is with Zhejiang University, Hangzhou 310027, China (e-mail: hanjinsong@zju.edu.cn).

The meta-learning process involves generating numerous few-shot tasks for pre-training, which demands significantly more computational resources and time compared to conventional supervised learning. (3) Common gesture recognition models, such as convolutional neural networks (CNNs), rely on fully connected layers (FCs) to map extracted gesture features to gesture classes. However, FCs can only handle inputs of fixed size. Given the irregular network traffic in communication environments and the varying durations of different gestures, maintaining a consistent input size is a challenge.

To tackle these challenges, we propose OneSense, a dynamic size-adaptive one-shot WiFi-based gesture recognition system. OneSense significantly reduces the manpower, time, and computational overhead required from both developers and users, while also accommodating variable input sizes for seamless integration into communication systems. First, developers collect a small set of real samples, which are then used to generate a large number of virtual samples. By combining both real and virtual samples, developers can efficiently train a general model. Subsequently, users can deploy OneSense within an off-the-shelf communication system. By fine-tuning the general model with just one real sample from each unseen class, users can achieve accurate one-shot gesture recognition in ISAC environments.

Specifically, we first propose a data enrichment algorithm to enhance the diversity of our training dataset, minimizing the need for extensive real-world data collection. Particularly, developers only need to gather a small set of real-world samples for a few base gesture classes. Our algorithm then generates a wealth of new gestures (referred to as virtual gestures) by applying the principle of signal propagation in the physical world. This approach allows OneSense to construct a diverse dataset for training the general model, significantly saving both time and manpower compared to vanilla data collection methods.

Then, we devise a novel one-shot learning framework called *aug-meta learning (AML)*, which combines the advantages of both traditional supervised learning and meta-learning. The framework comprises two pre-training stages: *aug-training* and *meta-training*. In the aug-training stage, the backbone model engages in conventional supervised learning using virtual samples, enhancing its deep feature extraction capabilities. In the subsequent meta-training stage, the AML framework applies classical meta-learning techniques to adapt to few-shot scenarios and generate a generalized model. This dual-stage pre-training process accelerates the convergence of the backbone model, significantly reducing both computational and time overhead

Finally, to empower OneSense with the ability to process the samples with varying dimensionality, we integrate a spatial pyramid pooling (SPP) component into the backbone model. This component segments the feature maps generated by the convolutional layer into spatial bins using multi-level pooling, effectively capturing both fine and coarse features. The results from all pooling levels are concatenated to form a fixed-length feature vector for the FC layer. This retains more contextual information and minimizes information loss, thereby improving feature extraction. Additionally, now that the backbone model can adapt to dynamic input sizes, we develop a data cropping method tailored to the characteristics of human gestures. This method focuses on retaining only the most representative features of the gesture within the CSI sample. The subsequent experiments cobfirm that the data cropping and dynamic size-adaptive model work in tandem to not only improve the compatibility with ISAC but also enhance the recognition performance.

We build a prototype of OneSense and conduct extensive experiments to evaluate its performance across four real environments. The results indicate that OneSense achieves a high one-shot recognition accuracy of 94.7%, surpassing existing WiFi-based few-shot gesture recognition approaches. A robustness study reveals that the recognition performance of OneSense remains satisfactory when facing varied environments, user locations, and user orientations. A case study in ISAC context indicates that OneSense can accommodate to irregular data traffic while exhibiting better performance than conventional interpolation-based solutions. Further experiments over four open-source datasets highlight the superiority of our data cropping method dynamic size-adaptive sensing model. Additionally, the proposed AML framework reduces the pre-training time cost by over 86% compared to conventional meta-learning approaches.

The contributions of this paper are summarized as follows:

- We propose OneSense, the first WiFi-based one-shot gesture recognition system capable of adapting to natural size-dynamic CSI in ISAC context.
- We design a data enrichment algorithm based on signal propagation laws to expand the training dataset, significantly reducing the manpower and time overhead of real data collection.
- We propose AML framework to enable efficient and scalable FSL. This framework holds great potential for a wide range of sensing tasks, such as person identification.
- We conduct extensive experiments in real-world environments, and the results demonstrate that OneSense achieves 94.7% one-shot gesture recognition accuracy. Additionally, OneSense exhibits robustness to variations in environment, user location, and user orientation.

II. Related Work

This work is primarily concerned with two types of techniques: gesture recognition and FSL.

A. Gesture Recognition

Gesture recognition is a critical and active research area that enables a wide range of applications, such as smart shopping [17] and virtual reality [5]. Traditional gesture recognition approaches typically rely on cameras [18], [19], wearable devices [20], [21], or sonars [22], [23]. Although these approaches achieve high recognition accuracy, they also have inherent limitations. Camera systems depend on optimal lighting and clear lines of sight, which raises privacy concerns. Wearable devices, while providing on-body sensing, can be cumbersome for users. Sonar-based solutions are limited by their effective sensing range. To overcome these challenges,

researchers have explored the use of WiFi signals for gesture recognition [5], [24]–[35]. WiFi-based sensing offers several compelling advantages, including enhanced privacy, robustness to occlusion, and a larger sensing range [36]–[43]. Most importantly, WiFi sensing can leverage existing communication infrastructure, making it a promising candidate for integration into current communication systems for widespread deployment.

Existing WiFi-based approaches typically extract features from CSI and map them to gesture classes using deep learning models. However, they face two significant challenges that hinder practical implementation. First, these methods require extensive training data to establish the association between CSI and gesture classes, leading to considerable data collection overhead. Additionally, they necessitate that model inputs maintain a consistent size. Given the variability in data packets, achieving a uniform size for all CSI samples is challenging. In recent years, some WiFi-based studies have explored few-shot gesture recognition, aiming to identify gestures with only a limited number of labeled samples [13], [44]–[46]. However, these approaches still encounter high overhead and suboptimal accuracy. For instance, OneFi [13] employs virtual sample generation and transfer learning to achieve few-shot recognition of unseen gestures. Nonetheless, this method requires estimating velocity distributions from CSI using at least three receivers, making it both time-consuming and resource-intensive. Another solution, WiGr [46], utilizes a modified prototypical network to enhance recognition performance in few-shot scenarios. However, when the number of gesture classes changes, WiGr must re-train the entire model from scratch, leading to additional computational overhead. Furthermore, the one-shot recognition accuracies of existing methods [47], [48] are low (falls short of 90%), rendering them not ready for practical application. Also worthy of attention is the fact that existing literature lacks investigations into size-dynamic gesture recognition within communication contexts.

To address these challenges, we propose OneSense, a WiFi-based one-shot gesture recognition approach that can be seamlessly integrated into communication systems. OneSense tackles the overhead and accuracy issues encountered in existing few-shot methods through virtual gesture generation and innovative deep learning framework designs, offering a promising solution for real-world deployment. Additionally, our deep learning framework incorporates a SPP component, enabling the backbone model to adapt to varying sample sizes. We also develop a data cropping method that retains the most representative segments of the signal, further enhancing recognition performance while minimizing the computational overhead.

B. Few-shot Learning

FSL aims to enable models to learn from a minimal number of labeled samples, often just one or a few per class, making it particularly valuable in situations where data collection is costly or impractical. Existing FSL methods can be categorized into four main types [49]: data augmentation [50], [51], transfer learning [52], [53], multimodal learning [54], [55], and meta-learning [56], [57]. Data augmentation enhances the data distribution by simulating various scenarios through either metric-based or generative techniques. However, when the available data samples are limited, the model struggles to accurately assess the true data distribution based solely on a small number of samples, leading to bias and a tendency to overfit. Transfer learning primarily relies on pre-training and fine-tuning to leverage knowledge from extensive auxiliary datasets. However, its effectiveness can diminish in the absence of relevant domains or large auxiliary datasets. Multimodal learning aims to integrate different forms of information, such as text, images, and audio, to address the challenge of limited useful information in real-world applications of FSL. Yet, effectively combining data from various modalities remains a significant hurdle in supporting FSL. Currently, meta-learning is recognized as the leading strategy for tackling FSL challenges. It enables models to quickly adapt to new tasks by utilizing knowledge gained from previously learned tasks, enhancing generalization and efficiency with limited data. During the training of the meta-learner across a range of tasks, it samples not only from the data space but also from the task space itself. By continually adjusting to the specifics of each individual task, the network develops a more abstract learning capability.

Given these advantages, we apply meta-learning to achieve few-shot gesture recognition. However, pre-training the meta-learner is often time-consuming and computationally intensive. To address this issue, we propose a novel FSL framework called aug-meta learning, which significantly reduces computational costs and accelerates training without compromising inference performance.

III. Primer

We achieve few-shot gesture recognition using WiFi CSI. This section begins by providing some foundational knowledge about WiFi CSI, followed by an introduction to the principles of FSL.

A. WiFi CSI

Current WiFi-based sensing techniques primarily rely on extracting CSI from WiFi packets [58]. WiFi CSI represents the channel frequency response of each OFDM subcarrier [59], capturing how WiFi signals propagate from the transmitter to the receiver after undergoing amplitude attenuation, phase shifts, and the addition of noise at the physical layer [60]. Each CSI entry with carrier frequency f_c at time t can be formulated as:

$$H(f_c, t) = \sum_{k=1}^{K} \alpha_k(t) e^{-j2\pi f_c \tau_k(t)} + N \tag{1}$$

where K is the number of multipaths. α_k and τ_k are the amplitude attenuation factor and propagation delay for the k-th path, respectively. N is the additive white Gaussian noise.

These K paths can be categorized into two types: *static* and *dynamic*. The static paths encompass direct propagation from the transmitter to the receiver, as well as reflections from

stationary objects in the environment. In contrast, dynamic paths involve reflections from moving objects. Accordingly, each CSI entry can also be divided into a static component, denoted as $H_s(f_c)$, and a dynamic component, represented as $H_d(f_c, t)$. Taking into account the additional phase offsets introduced by hardware and software errors, the estimated raw CSI entry can be expressed as:

$$H(f_c, t) = (H_s(f_c) + H_d(f_c, t))e^{j\theta(f_c,t)} + N \quad (2)$$

where $e^{j\theta(f_c,t)}$ represents the random extra phase offset, which encompasses timing alignment offsets, sampling frequency offsets, and carrier frequency offsets.

In the context of human gesture recognition, the movement of the human body induces variations in the amplitude and phase of the CSI multipath channels, particularly in dynamic scenarios. Consequently, we can extract gesture-related dynamic features from the CSI to enable WiFi-based gesture recognition.

B. Meta-Learning

WiFi-based gesture recognition systems typically utilize supervised learning to map features extracted from CSI into gesture classes. This approach necessitates that users collect a substantial number of samples for each class, which can be labor-intensive. To address this challenge, we implement a meta-learning-based FSL techniques to minimize data collection efforts.

In meta-learning approach [61], the learning problem is organized into a series of N-way K-shot tasks $\{T_i\}_{i=1}^{I}$, where N represents the number of classes, K indicates the number of samples available for each class, and I denotes the total number of tasks. Each task T_i comprises a support set S_i and a query set Q_i. The support set S_i provides a limited number of labeled samples to train the model, while the query set Q_i is used to evaluate the model's performance after training on S_i. The support set S_i consists of $N \times K$ samples, with N classes randomly selected from the dataset and K samples drawn for each new class. The query set Q_i contains samples from the same N classes as in S_i, but with different instances. By structuring the learning process around these tasks and training the model on various combinations of classes and sample, FSL enables the model to generalize effectively and make accurate predictions on new tasks with limited labeled data.

In this work, we focus on minimizing the data collection overhead to achieve one-shot gesture recognition using meta-learning techniques. Moreover, we introduce AML, a novel FSL framework, to further alleviate the computational demands associated with traditional meta-learning methods.

IV. System Overview

This paper proposes a one-shot ISAC-friendly WiFi-based gesture recognition system, namely OneSense. As shown in Fig. 1, OneSense primarily consists of four modules: data collection, data pre-processing, data enrichment, and gesture recognition. The deployment of OneSense can be divided into two phases: *bootstrapping phase* (developer edge) and *deployment phase* (user edge). Our key idea is that, the developers pre-train a recognition model based on the pre-collected samples (base gestures) as well as the virtual ones to initialize the system. Then, on the user edge, only a few labeled samples are required to fine-tune the pre-trained model for deployment on the customized task (unseen gestures).

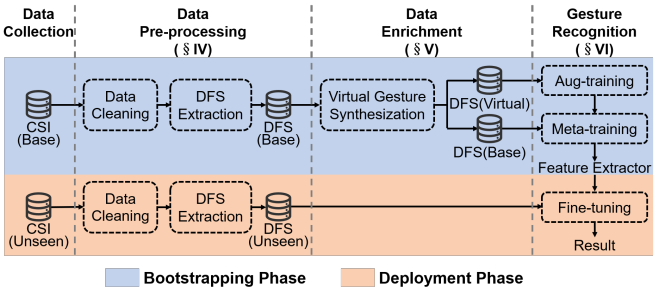


Fig. 1: Architecture of OneSense.

Bootstrapping phase. In this phase, OneSense aims to pre-train a feature extractor that serves as the foundation for one-shot recognition. Specifically, OneSense begins by collecting a batch of CSI samples of seen gestures to create a base dataset in the data collection module. The data pre-processing module then eliminates gesture-irrelevant components (e.g., noise) from the raw CSI measurements. With the cleaned CSI data, OneSense extracts Doppler frequency shift (DFS) as environment-independent gesture features. Next, in the data enrichment module, OneSense employs a virtual gesture construction algorithm to generate a large quantity of samples of virtual gestures based on the base dataset. Thereafter, OneSense perform data cropping to improve the feature prominence. Finally, in the gesture recognition module, OneSense trains the feature extractor using both the virtual dataset and the base dataset through AML.

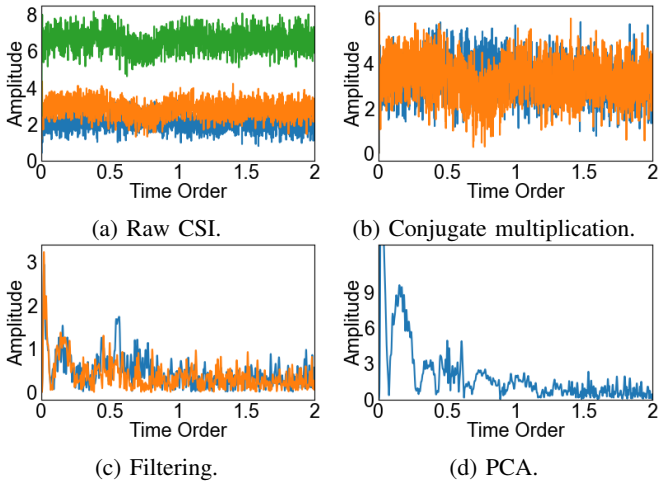
Deployment phase. In this phase, OneSense develops an accurate classifier for unseen gestures. Specifically, OneSense first collects a single sample for each unseen gesture in the data collection module. The collected samples then undergo the same pre-processing as in the bootstrapping phase. Following this, OneSense leverages the feature extractor pre-trained during the bootstrapping phase and attaches a classifier to it. OneSense fine-tunes the classifier using the pre-processed samples of unseen gestures, resulting in a model capable of accurate one-shot recognition for these gestures. After that, OneSense can monitor the WiFi traffic in off-the-shelf communication contexts and perform real-time gesture recognition.

V. Data Pre-processing

This section describes how OneSense removes gesture-irrelevant components from raw CSI measurements and extracts environment-independent DFS as gesture features. For clarity, we illustrate this process using three randomly selected subcarriers, as depicted in Fig. 2.

A. Data Cleaning

As mentioned in Sec. III-A, raw CSI measurements (Fig. 2(a)) contain many components irrelevant to the ges-



(a) Raw CSI. (b) Conjugate multiplication.

(c) Filtering. (d) PCA.

Fig. 2: Effectiveness of signal pre-processing.

ture information, such as phase offset, static components, and noise. These elements can degrade gesture recognition performance. To mitigate their effects, we employ a series of signal processing techniques to clean the raw CSI, including conjugate multiplication [62], frequency-based filtering, and principal component analysis (PCA) [63].

Conjugate multiplication. The slight out-of-sync between the transmitter and receiver can introduce a time-varying random phase offset $e^{j\theta(f_c,t)}$ (Eq. 2). Fortunately, since the antennas on the same receiver share the same RF oscillator, their phase offsets can be considered consistent. Based on this characteristic, we can eliminate such phase offset by performing conjugate multiplication between the CSI of two antennas on the same receiver, as illustrated in Fig. 2(b).

Frequency-based filtering. In addition to the phase offset, the received CSI also contains static components and white Gaussian noise. To eliminate their influences, we first apply high-pass filtering (with a 2 Hz cut-off frequency) to remove low-frequency components caused by static paths. Next, we perform low-pass filtering (with a 60 Hz cut-off frequency) to eliminate high-frequency noise. As shown in Fig. 2(c), the filtered CSI traces become smoother.

PCA. Ultimately, we apply PCA to the filtered CSI and extract the first principal component. This not only enhances the prominence of gesture-related features but also reduces any remaining noise. As shown in Fig. 2(d), the CSI after PCA is much cleaner. The envelope—representing the variation trend of the CSI profile—becomes clearer, enabling OneSense to extract representative and high-quality gesture features.

B. DFS Extraction

Due to the multipath effect, CSI measurements are influenced not only by the dynamics of the human body but also by the surrounding environment. As a result, the CSI profiles of the same gesture collected in different environments may vary. Furthermore, a gesture recognition model trained in one environment may perform inadequately in another. To address this issue, we choose to extract DFS from the CSI as gesture features. Since DFS is environment-independent, a model based on DFS can maintain robust performance across different environments. In the following sections, we will demonstrate how to extract DFS from WiFi CSI in accordance with the Doppler effect [64].

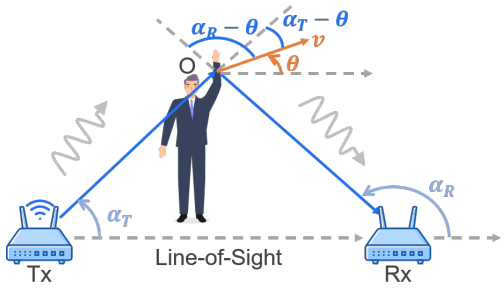


Fig. 3: WiFi signal propagation model in physical world.

A scenario of WiFi-based human gesture recognition is shown in Fig. 3. If we consider the target user O as a point, the motion of O will induce a frequency change between the transmitter Tx and the receiver Rx, due to Doppler Effect. Such frequency variation is referred to as DFS, which can be calculated by:

$$f_D = -f_0 \frac{v}{c}(cos(\theta - \alpha_R) + cos(\theta - \alpha_T)) \tag{3}$$

where f_0 is the frequency of the WiFi signal transmitted by Tx, v is the speed of the target user, c is the speed of light, α_T is the angle of departure (AoD), and α_R is the angle of arrival (AoA).

However, in reality, a human body cannot be simply represented by a single point, especially in near-field scenarios where the distance between the human subject and the transmitter/receiver is short. In fact, different body parts generate varying velocity components, resulting in distinct DFS. To address this issue, we introduce the Doppler spectrogram as a means to represent the intensity of DFS components over time, encompassing the entire body. The Doppler spectrogram can be estimated from CSI measurements using time-frequency analysis techniques, such as the short-time Fourier transform (STFT). Hence, after data cleaning, we first apply STFT to derive the Doppler spectrogram from the CSI, and then select the portion of the spectrogram that best reflects the dynamics of the human body as input for the subsequent sensing model.

Specifically, let $x(t)$ be the time-series CSI data and $w(t)$ represent a window function in STFT. The Doppler spectrogram at a particular frequency f and time t can be obtained by calculating the square of the STFT magnitude, expressed as:

$$S(f,t) = \left| \int_{-\infty}^{\infty} x(\tau) w(t-\tau) \mathrm{e}^{-2\pi i f \tau} d\tau \right|^2. \tag{4}$$

Next, we retain the portion of the spectrogram that best captures the Doppler shift, specifically the range of frequencies between [-60Hz, 60Hz] [13]. The resulting Doppler spectrogram, illustrated in Fig. 4(a) and (b), offers valuable insights into human body movements.

C. DFS Cropping

While the DFS can provide a wealth of gesture features, directly using the DFS matrix as input for the subsequent recognition model presents two problems. First, as illustrated in Fig. 4(a) and (b), certain frequencies exhibit low energy (blue areas). Feeding the entire DFS matrix into the recognition model thus wastes computational resources and time. Second, since different gestures may occupy distinct frequency ranges, retaining the full frequency range (i.e., [-60Hz, 60Hz]) for each gesture class can obscure the unique frequency characteristics of individual gestures, ultimately limiting recognition performance. To enhance computational efficiency and improve recognition accuracy, we propose a DFS cropping method that focuses on the most prominent frequency components.

Specifically, before calculating the DFS, we first identify most significant frequencies based on their power. More specifically, we begin by computing the power spectrum $P(f)$ of the signal $x(\tau)$, defined as:

$$P(f) = \left| \int_{-\infty}^{\infty} x(\tau) e^{-2\pi i f \tau} d\tau \right|^2, \tag{5}$$

where $P(f)$ represents the power spectrum at frequency f and τ is the time variable. Next, we establish a threshold T to filter out less significant frequencies:

$$T = k \cdot \max(P(f)) \tag{6}$$

Here, k is a factor less than 1 (for instance, 0.2) to select the sufficiently strong frequencies. Subsequently, we define the set of selected frequencies f_{selected} as:

$$f_{\text{selected}} = \{f \mid P(f) > T\}. \tag{7}$$

This set includes all frequencies where the power exceeds the established threshold. Finally, we perform the STFT only for the frequencies in the selected set:

$$S(f,t) = \begin{cases} \left| \int_{-\infty}^{\infty} x(\tau) w(t-\tau) e^{-2\pi i f \tau} d\tau \right|^2 & \text{if } f \in f_{\text{selected}} \\ \text{discard} & \text{otherwise} \end{cases} \tag{8}$$

The above method ensures that only significant frequency components are analyzed, enhancing the efficiency and performance of OneSense.

VI. Data Enrichment

As mentioned in Sec. III-B, users must prepare a dataset containing a diverse range of gesture classes to pre-train the model when employing meta-learning for one-shot recognition. Collecting real gesture samples in physical world is laborious. To tackle this problem, Xiao et al. [13] propose generating virtual gestures by simulating the rotation of real gestures in a two-dimensional plane, i.e., altering the orientation of the original gestures. While this approach does increase the size of the support set, it does not fundamentally introduce new gesture classes; a virtual gesture remains identical to the real gesture from which it is derived. Consequently, this virtual gesture generation scheme may lead the model to confuse gestures from the same class—albeit with slightly different orientations—for distinct classes.

In this section, we propose a novel virtual gesture synthesis method to generate new gesture classes that have not been explicitly observed during data collection. A virtual gesture sample is a synthesized gesture that retains the temporal characteristics and movement patterns of multiple source gestures while introducing new combinations. For instance, if we have source gestures 'L' and 'I', we can connect them to create a virtual sample '⊔'. However, achieving an effective synthesis is difficult, as directly concatenating multiple real gestures presents several limitations: (1) Taking two source samples as examples, splicing them nearly doubles the duration of the generated sample, resulting in temporal differences. In this case, the model tends to distinguish between the gestures based on the duration differences, overlooking their intrinsic features. (2) If we discard half of each source sample before concatenation, the duration of the new sample remains relatively unchanged, but essential gesture information is lost, violating the physical laws of signal propagation. (3) Simply combining two samples fails to introduce new velocity distributions, limiting the recognition model's ability to extract features effectively. To address these issues, we propose a solution that offers three key benefits: it approximately maintains the duration of new samples, adheres to physical laws, and introduces new velocity distributions. The core idea of our virtual gesture generation method is to accelerate the samples from multiple real gesture classes in accordance with the principles of the signal propagation and then combine them to create a new class.

Gesture acceleration. To shorten the duration of the sample and introduce new velocity distributions, we accelerate gesture samples based on the physical laws of signal propagation. For a source sample of a base gesture, if we accelerate it to only $\frac{1}{n}$ of the original time, the position of the moving target at time t after acceleration will correspond to its position at time nt before acceleration. Thus, we establish the following relations:

$$\begin{aligned} &s_{acc}(t) = s(nt), \quad \theta_{acc}(t) = \theta(nt), \\ &\alpha_{R_{acc}}(t) = \alpha_R(nt), \quad \alpha_{T_{acc}}(t) = \alpha_T(nt), \end{aligned} \tag{9}$$

where s, θ, α_R, and α_T represent the passed distance, moving direction, AoA, and AoD of the source sample, respectively. The terms s_{acc}, θ_{acc}, $\alpha_{R_{acc}}$, and $\alpha_{T_{acc}}$ correspond to these measurements after acceleration. Consequently, the velocity at time t after acceleration can be calculated as follows:

$$v_{acc}(t) = \frac{\mathrm{d}s_{acc}(t)}{\mathrm{d}t} = \frac{\mathrm{d}s(nt)}{\mathrm{d}t} = n\frac{\mathrm{d}s(t)}{\mathrm{d}t} = n \cdot v(nt) \tag{10}$$

By combining this equation with Eq. 3, we can determine the DFS of the accelerated gesture at time t:

$$\begin{aligned} f_{D_{acc}}(t) &= -f_0 \frac{v_{acc}(t)}{c} (cos(\theta_{acc}(t) - \alpha_{R_{acc}}(t)) \\ &\quad + cos(\theta_{acc}(t) - \alpha_{T_{acc}}(t))) \\ &= -f_0 \frac{n \cdot v(nt)}{c} (cos(\theta(nt) - \alpha_R(nt)) \\ &\quad + cos(\theta(nt) - \alpha_T(nt))) \\ &= n \cdot f_D(nt) \end{aligned} \tag{11}$$

Algorithm 1 Virtual gesture generation by accelerating two real samples.

Require: Source gesture samples $S_A(f,t)$ and $S_B(f,t)$
Ensure: Virtual gesture spectrogram $S_{A+B}(f,t)$
Step 1: Gesture Acceleration
for each source gesture $G \in \{A, B\}$ **do**
  for each time step $t \in [0, T]$ **do**
    Compute accelerated spectrogram: $S_{G_{acc}}(f,t) = S_G(f/2, 2t)$
  end for
end for
Step 2.1: Frequency Alignment
Determine the maximum frequency range: $F_{\max} = \max(F_A, F_B)$
for each source gesture $G \in \{A, B\}$ **do**
  if $F_G < F_{\max}$ **then**
    Pad $S_{G_{acc}}(f,t)$ in frequency domain to match $F_{\max}$
  end if
end for
Step 2.2: Virtual Gesture Generation
for each time step $t \in [0, T]$ **do**
  if $0 \le t < \frac{T}{2}$ **then**
    $S_{A+B}(f,t) = S_A(f/2, 2t)$
  else
    $S_{A+B}(f,t) = S_B(f/2, 2t - T)$
  end if
end for
return $S_{A+B}(f,t)$

Let denote the Doppler spectrogram of the source sample as $S(f,t)$, where $f \in [-F, F]$, $t \in [0, T]$. Then, for the accelerated gesture, the Doppler spectrogram can be expressed as:

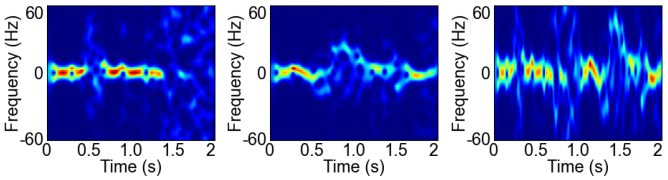
$$S_{acc}(f,t) = S(\frac{f}{n}, nt), \tag{12}$$

where $f \in [-nF, nF]$ and $t \in [0, \frac{T}{n}]$.

Virtual gesture generation. Thereafter, we merge the accelerated real samples to create virtual samples. Since DFS cropping may cause inconsistencies in the frequency-domain dimension, we apply padding before concatenation to ensure that all real samples have the same number of frequency components. As shown in Alg. 1, when synthesizing a new gesture from two source gestures, A and B, we can generate the Doppler spectrogram sample of the virtual gesture A+B as follows:

$$S_{A+B}(f,t) = \begin{cases} S_A\left(\frac{f}{2}, 2t\right) & 0 \le t < \frac{T}{2} \\ S_B\left(\frac{f}{2}, 2t - T\right) & \frac{T}{2} \le t \le T \end{cases} \tag{13}$$

The process outlined above illustrates how to generate a single virtual gesture using two real gestures. In fact, this method can be easily extended to incorporate more than two real gestures. Theoretically, users can create an infinite number of virtual gesture classes, along with corresponding virtual samples, to enrich the training set. Figure 4(c) displays the Doppler spectrograms of a virtual gesture synthesized by two source gestures shown in Fig. 4(a) and (b), demonstrating the effectiveness of our virtual gesture generation approach.



(a) Drawing '1' (b) Drawing '2'. (c) Virtual gesture.

Fig. 4: Generating Doppler spectrograms of virtual gesture (c) by combining real ones (a) and (b).

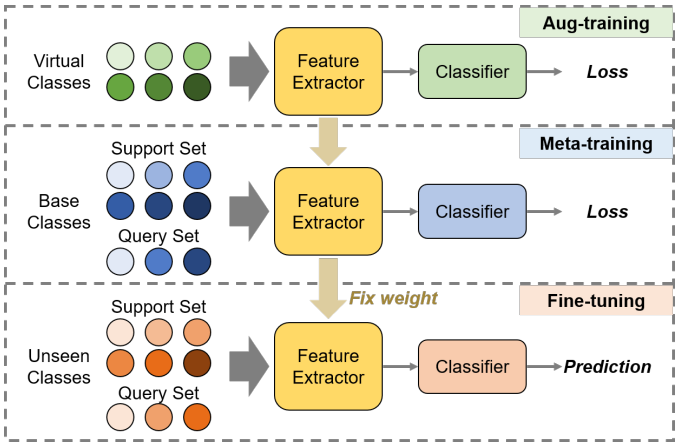


Fig. 5: Aug-meta learning framework.

VII. Aug-Meta Learning

OneSense aims to leverage meta-learning for one-shot recognition of unseen gestures. However, we have observed that the pre-training phase in meta-learning introduces significant time costs and computational overhead. In this section, we present a new FSL framework called AML, which enables efficient and rapid pre-training while ensuring high performance in one-shot gesture recognition[1].

As illustrated in Fig. 5, our AML framework consists of three stages, namely *aug-training*, *meta-training*, and *fine-tuning*. The key idea is that: in the *aug-training* stage, AML enhances the backbone model's deep feature extraction capabilities through traditional supervised learning using virtual gestures. Next, during the *meta-training* stage, AML pre-trains the backbone model employing a meta-learning technique on real collected gestures, thereby improving the model's generalization and adaptability in few-shot scenarios. Finally, in the *fine-tuning* stage, AML fine-tunes the pre-trained model with samples of unseen gestures to improve its ability to recognize these new gestures. The backbone model is composed of a feature extractor followed by a classifier. In the following, we will detail the three stages of the AML learning framework and then elaborate on the architecture of the backbone model.

[1]We theoretically analyze the superiority of AML compared to traditional meta-learning, in Appendix A.

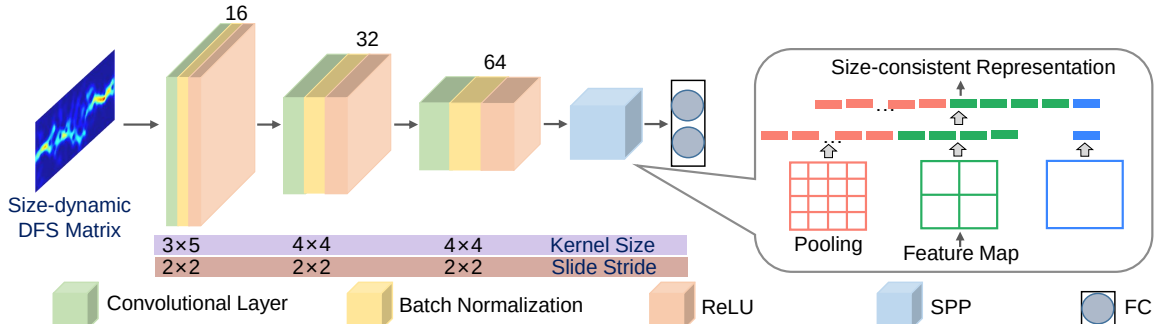


Fig. 6: Dynamic size-adaptive backbone model.

TABLE I: Details of the gesture classes considered in the evaluation. (CCW means counterclockwise.)

Base Gestures	Draw '1'	Draw '2'	Draw '3'	Draw '4'	Draw '5'
	Draw '6'	Draw '7'	Draw '8'	Draw '9'	Draw '0'
	Draw '0' (CCW)	Draw 'U'	Draw 'U' reverse	Draw 'N'	Draw 'N' reverse
	Swing left and right	Swing right and left	Draw Rectangle	Draw Rectangle (CCW)	Dig
Unseen Gestures	Push and Pull	Sweep	Slide	Clap	Draw zig-zag
	Draw Triangle	Draw 'a'	Draw 'b'	Draw 'c'	Draw 'd'
	Draw 'e'	Draw 'f'	Draw 'g'	Draw 'h'	Draw 'i'
	Draw 'j'	Draw 'k'	Draw 'l'	Draw 'm'	Draw 'n'

A. AML Framework

1) Aug-training: In the aug-training stage, we pre-train a feature extractor f_θ and a classifier c_ω using the generated virtual dataset D_{virtual} through classical supervised learning. We begin by randomly initializing the parameters of f_θ and c_ω. Next, we optimize the parameter sets θ and ω by minimizing the following loss function:

$$\mathcal{L}_{D_{\text{virtual}}}(\theta,\omega)=\frac{1}{|D_{\text{virtual}}|}\sum_{(x,y)\in D_{\text{virtual}}} l(c_\omega(f_\theta(x)),y) \quad (14)$$

where l represents the cross-entropy loss [65]. x and y denote a gesture sample and its corresponding label, respectively.

After pre-training, the feature extractor will acquire the capability of deep feature extraction. Since the number of virtual gesture classes may differ from that of the dataset used for meta-learning in the next stage, the classifier c_ω will be discarded after aug-training.

2) Meta-training: In the meta-training stage, we employ FSL techniques to sample a set of n-way 1-shot tasks $\{T_i\}_{i=1}^{I}$ from the real collected dataset, i.e., the base dataset D_{base}. Here the integer n can be selected from the interval $[2, N_{\text{base}}]$, where N_{base} is the number of classes in D_{base}. We begin training by loading the parameters of the feature extractor f_θ pre-trained in the previous stage, while randomly initializing a new classifier c_ϕ. For each task $T_i=(S_i,Q_i)$, we optimize the parameters ϕ using the support set S_i, by minimizing the loss $\mathcal{L}_{S_i}(\theta,\phi)$. Similar to Eq. 14, the loss $\mathcal{L}_{S_i}(\theta,\phi)$ is calculated as follows:

$$\mathcal{L}_{S_i}(\theta,\phi)=\frac{1}{|S_i|}\sum_{(x,y)\in S_i} l(c_\phi(f_\theta(x)),y) \quad (15)$$

During the above optimization process, we do not update the model parameters directly. Instead, we record the optimized parameters as ϕ_i for each task T_i. Then, we calculate the loss on the query set Q_i using the optimized parameters ϕ_i, denoted as $\mathcal{L}_{Q_i}(\theta,\phi_i)$.

Once we have completed training on all the tasks in $\{T_i\}_{i=1}^{I}$, we proceed to adapt the parameter sets θ and ϕ of the model. This is achieved by minimizing the accumulated loss $\sum_{i=1}^{I}\mathcal{L}_{Q_i}(\theta,\phi_i)$. This stage updates the feature extractor to learn superior representations from real-world samples, enhancing its ability to handle few-shot scenarios. The optimized feature extractor will be directly used in the subsequent stage, while the classifier will be discarded once more.

3) Fine-tuning: In the final stage, we generate an N_{unseen}-way 1-shot support set S from the dataset D_{unseen}, where N_{unseen} represents the number of classes in D_{unseen}. This support set consists of only one labeled sample for each unseen class. We begin by loading the pre-trained feature extractor f_θ from the previous stage and fixing its parameters. Then, we attach a new classifier c_ψ to the tail of f_θ and fine-tune it using the one-shot support set S. After fine-tuning, the feature extractor and classifier collaborate to recognize unseen gestures with high accuracy.

B. Dynamic Size-Adaptive Backbone Model

Traditional deep learning-based gesture recognition models employ fully connected (FC) layers to map extracted gesture representations to the confidence scores of each gesture class. In OneSense, the FC layer serves as the classifier. However, due to the fixed number of neurons in the FC layer, it can only accept size-consistent feature maps as input. When integrating OneSense into communication systems, the unevenness of the packets leads to changes in the duration of the DFS. Additionally, our data cropping method in Sec. V-C results in a variable dimension of the frequency. Consequently, the feature extractor cannot maintain a size-fixed feature map for the FC layer. To address this issue, we design a dynamic size-adaptive backbone model to ensure OneSense's compatibility with existing communication systems.

As illustrated in Fig. 6, the backbone model consists of a feature extractor and a classifier (i.e., FC layer). The feature

extractor includes three convolutional layers with channel counts of 16, 32, and 64, and kernel sizes of (3, 5), (4, 4), and (4, 4), respectively. To provide a length-fixed representation for the classifier, we integrate a SPP component after the last convolutional layer. SPP divides the feature map into multiple levels of sub-regions; for instance, the first layer might create 1×1 regions, the second layer 2×2, and the third layer 4×4. Within each region, SPP performs pooling operations (such as max pooling), aggregating the features into a single value. This ensures that the resulting feature vector maintains a consistent dimension, enabling effective processing in the subsequent FC layer and ultimately yielding a fixed-length feature representation

$$v = [\text{Pool}_{1\times1}(FM), \text{Pool}_{2\times2}(FM), \text{Pool}_{4\times4}(FM)], \quad (16)$$

where v represents the size-fixed gesture representation and FM is the output of the last convolutional layer. Through multi-level pooling, SPP captures features at various sizes and scales, enhancing the model's adaptability and accuracy in complex gesture recognition scenarios. This design allows OneSense to be seamlessly integrated into off-the-shelf communication environments, achieving accurate one-shot sensing.

VIII. Evaluation

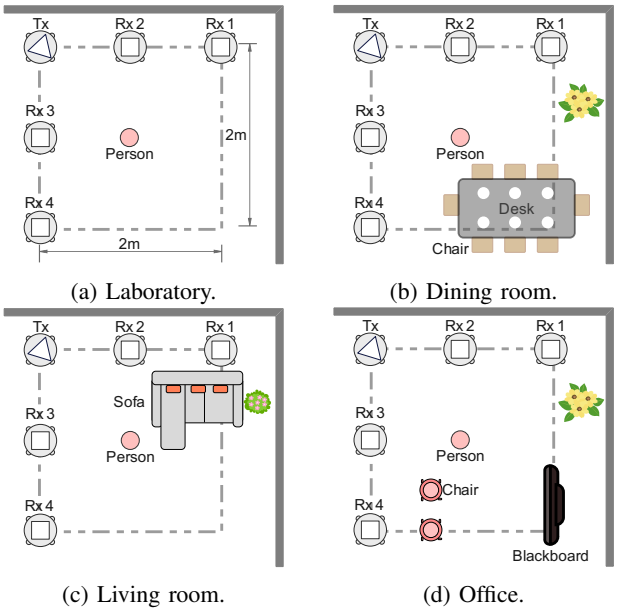
This section presents the real-world implementation and evaluation results of OneSense.

Experiment setup. We conduct experiments in four environments, including a laboratory, a dining room, a living room, and an office, as shown in Fig. 7. Our assessment follows a widely adopted WiFi sensing setup [5], [13], i.e., utilizing one transmitter and four receivers to collect data within a $2m \times 2m$ square, all of which are commercial off-the-shelf (COTS) laptops equipped with Intel 5300 network interface cards (NICs). The transmitter, which has one antenna, sends WiFi packets at a rate of 1000 packets per second, while each receiver is configured in monitor mode with three antennas. We install the Linux 802.11n CSI Tool [66] to extract CSI from WiFi data packets. The signals operate within the standard communication frequency band of 5.54 GHz.

Data collection. We invite 10 volunteers to perform gestures in our experiments, comprising six males and four females. We define 40 gestures, as outlined in Tab. I, with 20 serving as base gestures and the remaining 20 classified as unseen gestures. Each base gesture is performed at least 30 times, resulting in a total of over 2900 gesture samples collected. In the default setting, we use 20 base gestures along with 6 unseen gestures ('push and pull', 'sweep', 'slide', 'clap', 'draw zig-zag', and 'draw triangle') performed in the laboratory environment for evaluation. For K-shot recognition, K samples are randomly selected for fine-tuning and the remaining samples are used for testing.

Metric. We define accuracy to quantify the performance of gesture recognition. Accuracy reflects the probability that a sample is correctly recognized and is calculated using the formula:

$$accuracy = \frac{N_{cor}}{N_{all}}, \quad (17)$$



(a) Laboratory. (b) Dining room.

(c) Living room. (d) Office.

Fig. 7: Experiment setup in four environments.

where N_{cor} and N_{all} denote the number of correctly identified samples and the number of all tested samples, respectively.

A. Overall Performance

In this part, we conduct three experiments: (1) comparing OneSense with three existing works; (2) exploring the efficacy of the training strategies in AML framework; and (3) validating the effectiveness of the data enrichment.

Comparison with existing works. To highlight the superiority of OneSense, we compare it with two classical methods (OneFi [13] and WiGr [46]) and a state-of-the-art (SOTA) method (CorssFi [67]). Like OneSense, OneFi employs virtual gesture generation for data augmentation, followed by transfer learning to adapt to unseen gestures. In contrast, WiGr enhances few-shot gesture recognition by modifying the prototypical network. Meanwhile, CrossFi leverages a Siamese network to compare unseen samples with stored templates for gesture classification. We evaluate its performance using the publicly available code provided with the original paper. The accuracies of these methods are presented in Fig. 8. As shown, OneSense achieves 94.7%, 98.9%, and 99.1% accuracies in 1-shot, 3-shot, and 5-shot settings, respectively. In comparison, OneFi achieves 84.2%, 94.2%, and 95.8%, while WiGr attains 81.7%, 92.3%, and 93.9% under the same settings. CrossFi, on the other hand, achieves 48.4%, 89.8%, and 97.8% accuracies. These results clearly indicate that increasing the number of samples for fine-tuning improves performance. More importantly, OneSense consistently outperforms all other methods across different shot configurations. Although CrossFi's 5-shot accuracy surpasses those of OneFi and WiGr, its 1-shot accuracy is significantly lower, which could be attributed to the difficulty of finding an optimal template for the Siamese network with just a single data collection. *Overall, OneSense not only surpasses existing methods in performance but also significantly reduces data collection overhead and*

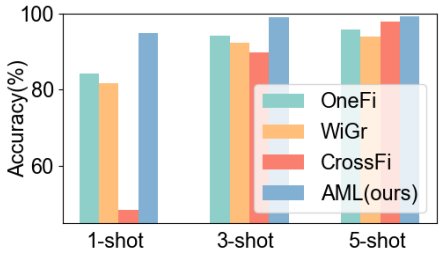


Fig. 8: Comparison with existing works.

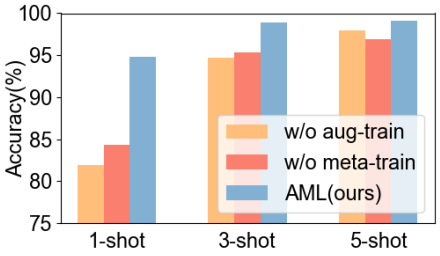


Fig. 9: Effect of AML framework.

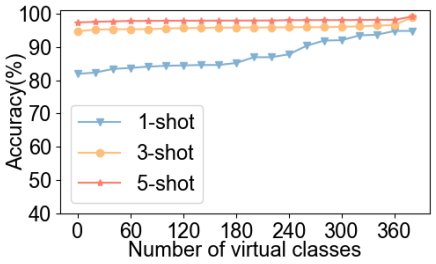


Fig. 10: Effect of data enrichment.

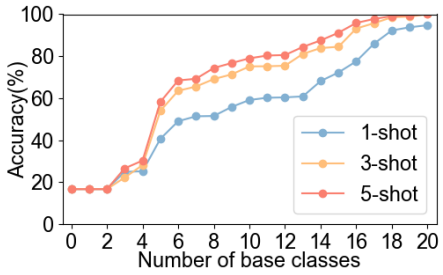


Fig. 11: Effect of no. of base classes.

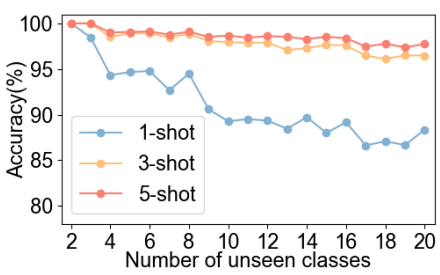


Fig. 12: Effect of no. of unseen classes.

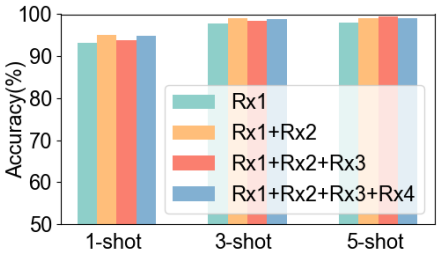


Fig. 13: Effect of no. of receivers.

TABLE II: Comparison with standard few-shot learning.

Method	ML	TL	MB	GB	**OneSense**
1-shot accuracy	81,8%	65.5%	64.7%	29.4%	**94.8%**
3-shot accuracy	94.7%	76.5%	74.8%	66.7%	**98.9%**
5-shot accuracy	98.0%	82.7%	80.0%	76.7%	**99.1%**

offers greater flexibility in accommodating unseen gestures. Furthermore, when the data is not cropped and the SPP component is removed from the backbone model, OneSense achieves 1-shot accuracy of 93.0%. *This suggests that the data cropping method enhances feature prominence, while the dynamic-size adaptive backbone model effectively improves recognition accuracy.*

Efficacy of aug-meta learning. To reduce the time cost associated with primitive meta-learning without compromising recognition performance, we propose an AML framework consisting of two pre-training stages. To evaluate the efficacy of each stage, we sequentially remove one stage at a time and recalculate the accuracies under 1-shot, 3-shot, and 5-shot settings. We refer to these as "w/o aug-train" and "w/o meta-train", respectively. The results are presented in Fig. 9. As observed, regardless of the number of shots, the complete AML framework consistently outperforms both "w/o aug-train" and "w/o meta-train". *This demonstrates that both the aug-training and meta-training stages significantly enhance few-shot recognition performance. Furthermore, as the training time of traditional mete learning is much longer than that of AML (see Sec. VIII-G), our design achieves the best of both worlds—reducing time cost while maintaining accurate one-shot recognition.*

Effectiveness of data enrichment. Meta-learning would require the user to prepare a large quantity of samples for pre-training. To alleviate the data collection overhead, we propose a data enrichment method to generate virtual classes through physical modeling. To validate its effectiveness, we evaluate the few-shot recognition accuracy with the inclusion of 0 to 380 virtual gesture classes. As shown in Fig. 10, the accuracy generally increases with the number of virtual gestures. *This means that data enrichment can effectively enhance the feature extraction capabilities of the backbone model, resulting in improved recognition performance.*

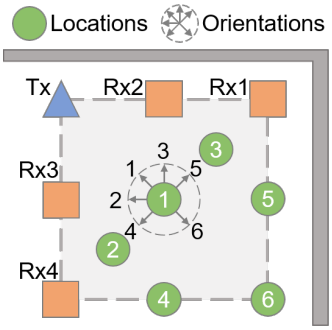


Fig. 14: Different locations and orientations.

Comparison with standard few-shot learning. To further demonstrate the superiority of AML, we compare it with four standard few-shot learning methods, including meta-learning (ML) [56], transfer learning (TL) [52], metric-based method (MB) [68], and generative-based method (GB) [69]. ML leverages base gestures to train a general model, which is then adapted to unseen gestures using only a few samples. TL trains a gesture recognition model with strong feature extraction capabilities on base gestures and fine-tunes only the final layers on unseen gestures. MB employs relation networks to learn gesture representations optimized for comparison, determining labels by computing similarities between unseen representations and templates. GB utilizes base gestures to train a generative adversarial network (GAN) that synthesizes new samples, which are then used to train gesture recognition models for unseen gestures. Experimental results, presented in Tab. II, reveal that the GB method performs the worst. This is likely due to the difficulty of generating high-fidelity DFS with limited samples of both base and new gestures. In contrast, ML

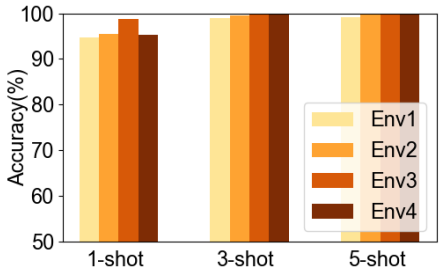


Fig. 15: Cross-environment accuracies.

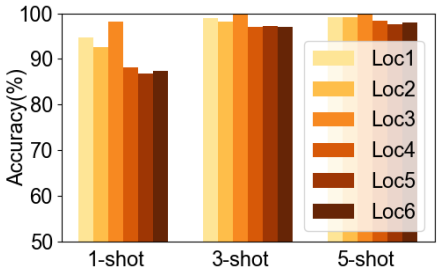


Fig. 16: Cross-location performance.

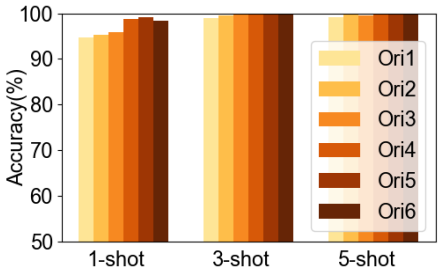


Fig. 17: Cross-orientation performance.

and TL achieve strong performance, second only to OneSense. Notably, OneSense exhibits the best overall performance by effectively combining the advantages of both ML and TL.

B. Effect of Hyper-parameters

In this part, we examine the impact of the hyper-parameters in OneSense, including the number of base classes, unseen classes, as well as receivers.

Number of base classes. OneSense generates virtual classes by concatenating base classes. In theory, increasing the number of base classes should result in a greater number of virtual classes for augmentation during aug-training. To investigate the impact of the number of base classes, we vary the number from 0 to 20 and re-evaluate the performance. The 1-shot, 2-shot, and 3-shot recognition accuracies are presented in Fig. 11. As observed, the accuracy begins to converge when the number of base classes reaches 16. Meanwhile, these results confirm that increasing the number of base classes indeed leads to higher accuracy.

Number of unseen classes. OneSense leverages fine-tuning to enable the backbone model to adapt to arbitrary unseen classes. To investigate how accuracy changes with the number of unseen classes, we vary the number from 2 to 20. The experimental results in Fig. 12 show that recognition accuracies can reach 100% when only two unseen classes are considered. Generally, the accuracies for 1-shot, 3-shot, and 5-shot settings decrease as the number of unseen classes increases, which is expected, as distinguishing among more classes becomes more challenging. However, the 1-shot accuracy remains above 85% when the number of unseen classes exceeds 16. Moreover, the accuracies surpass 95% when more samples (e.g., 3-shot) are used for fine-tuning. These results suggest that OneSense demonstrates exceptional scalability to new gestures.

Number of receivers. In our default setting, four receivers are used to capture WiFi packets, providing comprehensive coverage of the sensing area. Intuitively, increasing the number of receivers should improve recognition performance. To validate this hypothesis, we test OneSense with different numbers of receivers, ranging from 1 to 4. The resulting accuracies are shown in Fig. 13. It can be observed that OneSense achieves the lowest accuracy when only one receiver is used, but it still exceeds 90%, demonstrating the high flexibility of OneSense in terms of transceiver deployment. As expected, accuracy improves with the addition of more receivers. Interestingly, the accuracies with three receivers are lower than with two under 3-shot and 5-shot settings. This may be due to that the superposition of the signals between two receivers interfere each other, which could diminish the gesture features. Overall, while using more receivers generally leads to higher accuracy, even a single receiver can achieve acceptable performance.

C. Cross-domain Assessment

In practice, OneSense could be pre-trained by the developer in one domain and fine-tuned by the user in another. In this part, we evaluate the cross-domain performance of OneSense. The domains include three types: (1) four environments in which OneSense is deployed, as illustrated in Fig.7; (2) six user locations shown in Fig.14; and (3) six user orientations relative to the transmitter, as depicted in Fig. 14.

Cross-environment evaluation. As mentioned previously, we collect data in four different environments: laboratory, dining room, living room, and office, labeled as Env 1, 2, 3, and 4, respectively. In this experiment, the backbone model is pre-trained using data from Env 1 and tested on data from all four environments (Env 1, 2, 3, and 4). The experimental results, shown in Fig. 15, demonstrate that a greater number of fine-tuning samples leads to higher accuracy. Although the training and testing data come from different environments, OneSense achieves accuracies above 94%. This is rational, as the gesture features extracted from WiFi CSI are environment-independent DFS, which remain consistent across different environments. Additionally, fine-tuning helps the backbone model accommodate to new environments. Thus, OneSense exhibits excellent cross-environment recognition performance. Once the pre-training is completed by the developer, the system can be deployed in any environment with minimal cost for fine-tuning.

Cross-location evaluation. To investigate the effect of user location, we pre-train the backbone model using data collected at the first location and test it on data from all six locations (Loc 1 to Loc 6). The 1-shot, 3-shot, and 5-shot accuracies are shown in Fig. 16. As expected, accuracy increases with the number of shots. However, the differences between the training and testing locations negatively impact the recognition accuracy. Despite this, the 1-shot accuracy at any location remains above 86%. Fine-tuning allows the backbone model to adapt to new locations, significantly enhancing the usability of OneSense in real-world scenarios.

Cross-orientation evaluation. To understand how the changes in orientation affect the recognition accuracy, we pre-train OneSense with data collected when the user is facing the transmitter (Ori 1) and test it on data from all orientations (Ori 1 to Ori 6). The experimental results are shown in Fig. 17. As observed, the accuracies across all orientations exceed 94%.

Similar to the cross-location evaluation, fine-tuning enables the backbone model to accommodate to new orientations. This flexibility allows users to choose their orientation freely in practical implementations.

D. Case Study

To enable seamless integration of WiFi sensing into off-the-shelf communication systems, we propose OneSense to alleviate the data collection overhead and adapt to fluctuating data traffic. In this part, we conduct a case study to evaluate the effectiveness of OneSense in a communication setting. Specifically, we replicate an ISAC setup introduced in [70] for data collection, where a WiFi router keeps communicating with some terminal devices (e.g., smartphone and computer) and a US RT-AC86U Router measures the CSI. Twelve volunteers (eight males and four females) are asked to perform four activities described in WiFall [28], including sitting down, standing up, walking, and falling. The start and end of each activity are detected by comparing the amplitude variance against a predefined threshold. We collect a total of 4875 CSI samples across three environments (laboratory, living room, and office). Due to irregular network traffics and varying gesture durations, the packet counts in each CSI sample—i.e., the number of sampling points in the temporal domain—range from 156 to 1,737.

1) Overall Performance: To tackle the problem of dimensional variance, traditional approach usually adopt interpolation [71]. Here, we compare OneSense with traditional approach, in which all CSI samples are interpolated to have the same dimension in time (i.e., the largest packet count) without data cropping. The experiment results under three environments are shown in Fig.18. It can be observed that our approach can achieve 97.1%, 98.1%, and 97.7% accuracies in laboratory, living room, and office, respectively, while traditional method only achieves 92.9%, 93.8%, and 93.5% accuracies, respectively. These differences can be attributed to two key factors. On one hand, although the interpolation allows conventional deep learning models to perform gesture classification, the "predictive" elements introduced by interpolation may not accurately represent gesture features. Even worse, this could distort the original features, leading to performance degradation. On the other hand, our data cropping method enhances the prominence of gesture features, and the dynamic size-adaptive recognition model makes full use of these features, resulting in higher accuracy. Additionally, it can be found that the accuracies of different environments are similar to each other. *These results suggest that the design of OneSense, particularly the data cropping and dynamic size-adaptive backbone model, not only enhances compatibility with ISAC but also improves recognition accuracy, while maintaining robust performance across diverse environments.*

2) Impact of Interference: This work aims to design an *ISAC-compatible* gesture recognition system, namely OneSense. The most efficient way to implement OneSense is directly integrating it into existing off-the-shelf communication infrastructures. However, lots of wireless devices may coexist in the vicinity of OneSense. In such cases, signals from surrounding sources can introduce interference into the DFS or even occupy the same frequency band, potentially triggering a frequency switch and disrupting sensing. To assess the impact of such interference, we deliberately introduce five common types of noise (Gaussian, uniform, Rician, Rayleigh distributions, and constant bias) at varying magnitudes (0% to 10%) relative to the maximum energy of the original signals. As presented in Fig. 19, the introduced interference does affect gesture recognition accuracy. However, the degradation remains within an acceptable range. Even when the noise level reaches 10% (corresponding to an SNR of approximately 10 dB), our approach maintains an accuracy of nearly 90%. These results demonstrate the robustness of our method in interference-prone environments. If the energy of the interference noise becomes too severe, OneSense will switch frequency bands along with the communication system to find a cleaner channel, ultimately disrupting sensing. This could be a common limitation of ISAC-oriented WiFi sensing approaches.

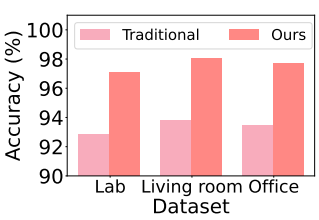


Fig. 18: Accuracies in ISAC context.

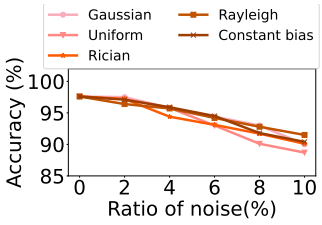


Fig. 19: Impact of interference.

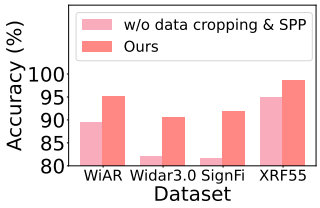


Fig. 20: Accuracies of open-source datastes.

TABLE III: Information of open-source datasets.

Feature / Dataset	No. Class	No. Subcarrier	Transmission Rate
WiAR [72]	5	30	30 p/s
Widar3.0 [5]	6	30	1000 p/s
SignFi [73]	10	30	100 p/s
XRF55 [74]	10	30	1000 p/s

E. Evaluation on Open-source Dataset

To further validate the benefits of data cropping and dynamic size-adaptive backbone model, we conduct comparative experiments on four open-source datasets: WiAR [72], Widar3.0 [5], SignFi [73], and XRF55 [74]. Therein, WiAR is used as an activity recognition dataset, Widar3.0 and SignFi are used as hand gesture recognition datasets, and XRF55 is used as a person identification dataset. We use the portion with highest accuracy for assessment. Previous WGR studies only considered ideal conditions, overlooking the impact of irregular traffic in communication networks. As a result, all CSI

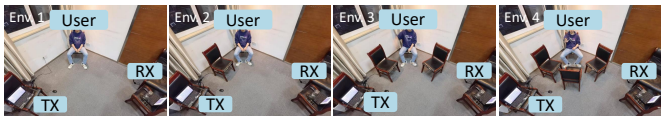


Fig. 21: Experiments in congested conditions.

samples within each dataset share the same dimensionality. The dataset details are provided in Tab. III.

To be specific, we evaluate the recognition accuracy under two data processing settings: (1) "Ours": copping the data to highlight the most representative features in the samples and applying our dynamic size-adaptive model for recognition. (2) "w/o data cropping and SPP": preserving the original dimensionality of all samples and using a backbone model without the SPP component for recognition. For both settings, we train and test the models using standard supervised learning with an [80%, 20%] random split of the dataset. The experiment results in Fig. 20 indicate that the accuracies of 'Ours" are consistently higher than those of "w/o data cropping and SPP" across all datasets. *This demonstrates that our data cropping method effectively enhances the prominence of features in each CSI sample. Our dynamic size-adaptive backbone model can handle the dimensionality changes caused by data cropping. Moreover, the performance improvement in person identification using the XRF55 dataset suggests that our solution is applicable to a wide range of sensing tasks.*

F. Impact of Congested Environment

This part investigates the impact of congested sensing environments. As shown in Fig. 21, we first collect data under four different conditions, using data from Env. 1 for pre-training. Then, we measure the 1-shot accuracy difference between Env. 1 and the other three conditions. The results indicate accuracy drops of 1.8%, 2.6%, and 5.0%, respectively, suggesting that our approach maintains robust performance under varying levels of congestion. This resilience can be attributed to the fine-tuning process, which enables the pre-trained model to adapt to new conditions with just a few samples, even when occlusions alter the distribution of sensing information in DFS. However, a higher degree of congestion leads to a more significant accuracy decrease, as occlusions can partially obscure sensing information. Thus, while congestion does impact OneSense's performance, fine-tuning helps the recognition model accommodate these new conditions effectively.

G. Time Cost and Complexity

1) Time Cost: The time costs of AML mainly come from two components: two-stage pre-training and fine-tuning. In this subsection, we assess the time costs of these two components based on an NVIDIA RTX 3080 GPU.

Pre-training. The experimental results show that, OneSense requires only 41.8 seconds for pre-training (including aug-training and meta-training stages), to achieve a one-shot recognition accuracy exceeding 90%. In comparison, a traditional meta-learning approach, MAML [15], demands more than 300 seconds of pre-training to achieve an accuracy of convergence (88.3%). It can be found that the delicate design of our AML framework not only reduces the pre-training latency (86.1%+ reduction), but also improves the recognition performance.

Fine-tuning. In the use of OneSense, pre-training is done by the developer, and the user only needs to perform fine-tuning. As only a classifier with a few parameters needs to be updated in fine-tuning, the tuning takes only 2.52 seconds. This allows the backbone model to quickly adapt to new gestures.

2) Complexity: To further evaluate the real-time performance of OneSense, we analyze its floating point operations per second (FLOPs), parameter count, and inference time for a single unseen sample. Experimental results show that OneSense requires only ∼13M FLOPs, significantly fewer than BPCloak [75] (∼57M), a real-time deep learning-based behavior privacy preserving method for WiFi sensing. Additionally, OneSense has just ∼0.486M parameters, which is substantially smaller than MobileNetV2 [76] (∼3.47M), a lightweight deep neural network that can run smoothly on mobile devices like iPhone 6s. Furthermore, the backbone model achieves an inference time of only ∼0.02s per sample. These results highlight the outstanding real-time efficiency of OneSense.

TABLE IV: Comparing OneSense with related works.

System	OneFi	WiGr	CrossFi	**OneSense**
1-shot accuracy	84.2%	81.7%	48.4%	94.8%
Interpretability	✓	×	×	✓
ISAC-compatibility	×	×	×	✓

H. Comparison to Related Works

In this part, we compare OneSense with three related works (OneFi, WiGr, and CrossFi) in terms of accuracy, interpretability, and ISAC-compatibility. As shown in Tab. IV, only OneSense achieves over 90% accuracy in the 1-shot setting. Both OneFi and OneSense offer strong interpretability due to their theoretical analysis of signal propagation and learning frameworks. Regarding ISAC-compatibility, only OneSense can be seamlessly integrated into communication systems, as it can effectively adapt to uneven WiFi traffic. Overall, OneSense outperforms existing methods across these key aspects.

IX. Conclusion

To enhance the scalability of WGR to new gestures and improve its compatibility with ISAC, this paper introduces a novel solution called OneSense. The design of OneSense begins with a virtual gesture generation method based on a signal propagation model to enrich the training data. Next, an AML framework is proposed, which enables scalable one-shot gesture recognition while significantly reducing model training overhead. Additionally, OneSense incorporates a data cropping technique to emphasize key features and a dynamic size-adaptive backbone model to enhance compatibility with communication systems. Extensive real-world experiments demonstrate that OneSense achieves over 94% accuracy in one-shot gesture recognition. Furthermore, the performance of OneSense remains stable despite changes in the environment,

user location, or orientation. A case study highlights the feasibility of OneSense for ISAC applications, while evaluations on four open-source datasets confirm the advantages of the data cropping method and dynamic size-adaptive model.

APPENDIX A
THEORETICAL ANALYSIS ON THE EFFECTIVENESS OF AML FRAMEWORK

In this part, we theoretically analyze the superiority of our AML framework compared to traditional meta-learning (MAML), in terms of recognition accuracy and time complexity.

A. Recognition Performance

Firstly, by pretraining with virtual samples, the backbone model learns better feature representations, reducing the variance of gradient updates. Let $\mathcal{T}$ represent the task distribution, and $\mathcal{L}_{\mathcal{T}}$ be the loss function for a given task. The variance of gradients during training satisfies:

$$\mathrm{Var}(\nabla_\theta \mathcal{L}_{\mathcal{T}}^{\text{aug-train}}) < \mathrm{Var}(\nabla_\theta \mathcal{L}_{\mathcal{T}}^{\text{MAML}}). \tag{18}$$

Since virtual samples enrich the training set, this stabilizes the gradient updates and leads to faster convergence.

Moreover, the generalization ability is improved by incorporating additional virtual samples. If we assume the generalization error follows a bound dependent on the number of training samples m (real) and M (virtual), then:

$$O\left(\sqrt{\frac{d\log(m+M)}{m+M}}\right) < O\left(\sqrt{\frac{d\log m}{m}}\right), \tag{19}$$

where d is the model complexity. Since $M > 0$, the error bound tightens, leading to better performance on unseen gestures.

B. Time Complexity

Traditional mate-learning consists of two stages: meta-training and fine-tuning. The total complexity is:

$$O(KNT) + O(KT), \tag{20}$$

where K is the number of gradient steps per task, N is the number of tasks in meta-training, and T is the time per gradient update.

Our approach adds a supervised aug-training step but reduces required meta-training iterations by starting from a better initialization. Its time complexity can be represented as:

$$O(MT) + O(KN'T'), \tag{21}$$

where M is the number of aug-training samples (fewer than KN). As fewer meta-training iterations are needed, we have $N' < N$. Meanwhile, due to faster convergence, we can also get $T' < T$.

Since $M \ll KN$ and $T' < T$, AML framework reduces the training time while achieving better recognition accuracy.

REFERENCES

[1] L. Zhao, R. Xiao, J. Liu, and J. Han, "One is enough: Enabling one-shot device-free gesture recognition with COTS WiFi," in *Proc. IEEE INFOCOM*, May 2024, pp. 1231–1240.

[2] W. Jiang, C. Miao, F. Ma, S. Yao, Y. Wang, Y. Yuan, H. Xue, C. Song, X. Ma, D. Koutsonikolas, W. Xu, and L. Su, "Towards environment independent device free human activity recognition," in *Proc. Annu. Int. Conf. Mobile Comput. Networking*, Oct. 2018, pp. 289–304.

[3] R. H. Venkatnarayan, G. Page, and M. Shahzad, "Multi-user gesture recognition using WiFi," in *Proc. Annu. Int. Conf. Mob. Syst., Appl. Serv.*, Jun. 2018, pp. 401–413.

[4] A. Virmani and M. Shahzad, "Position and orientation agnostic gesture recognition using WiFi," in *Proc. Annu. Int. Conf. Mob. Syst., Appl. Serv.*, Jun. 2017, pp. 252–264.

[5] 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 *Proc. Annu. Int. Conf. Mob. Syst., Appl. Serv.*, Jun. 2019, pp. 313–325.

[6] F. Meneghello, C. Chen, C. Cordeiro, and F. Restuccia, "Toward integrated sensing and communications in IEEE 802.11bf Wi-Fi networks," *IEEE Commun. Mag.*, vol. 61, no. 7, pp. 128–133, Jul 2023.

[7] K. Ma, C. Feng, G. Geraci, and H. H. Yang, "The meta distribution of the SIR in joint communication and sensing networks," Apr. 2024.

[8] J. Pegoraro, J. O. Lacruz, T. Azzino, M. Mezzavilla, M. Rossi, J. Widmer, and S. Rangan, "JUMP: joint communication and sensing with unsynchronized transceivers made practical," *IEEE Trans. Wirel. Commun.*, vol. 23, no. 8, pp. 9759–9775, Sep 2024.

[9] Y. He, J. Liu, M. Li, G. Yu, and J. Han, "Forward-compatible integrated sensing and communication for WiFi," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 9, pp. 2440–2456, Sep. 2024.

[10] Y. He, G. Yu, Y. Cai, and H. Luo, "Integrated sensing, computation, and communication: System framework and performance optimization," *IEEE Trans. Wireless Commun.*, vol. 23, no. 2, pp. 1114–1128, Feb. 2024.

[11] K. Cui, Q. Yang, L. Shen, Y. Zheng, F. Xiao, and J. Han, "Towards isac-empowered mmwave radars by capturing modulated vibrations," *IEEE. Trans. Mobile Comput.*, vol. 23, no. 12, pp. 13 787–13 803, Aug. 2024.

[12] K. Cui, L. Shen, Y. Zheng, F. Xiao, and J. Han, "Talk2radar: Talking to mmwave radars via smartphone speaker," in *Proc. IEEE INFOCOM*, May 2024, pp. 2358–2367.

[13] R. Xiao, J. Liu, J. Han, and K. Ren, "Onefi: One-shot recognition for unseen gesture via COTS WiFi," in *Proc. ACM Conf. Embed. Networked Sens. Syst.*, Nov. 2021, pp. 206–219.

[14] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proc. NeurIPS*, May 2016, pp. 3630–3638.

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

[16] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in *Proc. NeurIPS*, Jan. 2017, pp. 4077–4087.

[17] K. Cui, Y. Wang, Y. Zheng, and J. Han, "ShakeReader: 'read' UHF RFID using smartphone," *IEEE. Trans. Mobile Comput.*, vol. 22, no. 3, pp. 1793–1809, Mar. 2023.

[18] T. Li, Q. Liu, and X. Zhou, "Practical human sensing in the light," in *Proc. Annu. Int. Conf. Mob. Syst., Appl. Serv.*, 2016.

[19] M. Wang, B. Ni, and X. Yang, "Recurrent modeling of interaction context for collective activity recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, Jul. 2017, pp. 7408–7416.

[20] A. I. Withana, R. L. Peiris, N. Samarasekara, and S. Nanayakkara, "zSense: Enabling shallow depth gesture recognition for greater input expressivity on smart wearables," in *Proc. Conf. Hum. Fact. Comput. Syst.*, Apr. 2015, pp. 3661–3670.

[21] T. Zhao, J. Liu, Y. Wang, H. Liu, and Y. Chen, "PPG-based finger-level gesture recognition leveraging wearables," in *Proc. IEEE INFOCOM*, Apr. 2018, pp. 1457–1465.

[22] K. Kalgaonkar and B. Raj, "One-handed gesture recognition using ultrasonic doppler sonar," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process*, Apr. 2009, pp. 1889–1892.

[23] K. Yatani and K. N. Truong, "Bodyscope: a wearable acoustic sensor for activity recognition," in *Proc. ACM Int. Jt. Conf. Pervasive Ubiquitous Comput.*, Sep. 2012, pp. 341–350.

[24] P. Melgarejo, X. Zhang, P. Ramanathan, and D. Chu, "Leveraging directional antenna capabilities for fine-grained gesture recognition," in *Proc. ACM Int. Jt. Conf. Pervasive Ubiquitous Comput.*, Mar. 2014, pp. 541–551.

[25] H. Abdelnasser, M. Youssef, and K. A. Harras, "WiGest: A ubiquitous WiFi-based gesture recognition system," in *Proc. IEEE INFOCOM*, 2015.

[26] S. Ji, Y. Xie, and M. Li, "SiFall: Practical online fall detection with RF sensing," in *Proc. ACM Conf. Embed. Networked Sens. Syst.*, Jan. 2022, pp. 563–577.

[27] F. Adib and D. Katabi, "See through walls with WiFi!" in *Proc. Annu. Conf. ACM Spec. Interest Group Data Commun. Appl., Technol., Archit., Protoc. Comput. Commun.*, Aug. 2013, pp. 75–86.

[28] Y. Wang, K. Wu, and L. M. Ni, "WiFall: Device-free fall detection by wireless networks," *IEEE. Trans. Mobile Comput.*, vol. 16, no. 2, pp. 581–594, Apr. 2017.

[29] S. Palipana, D. Rojas, P. Agrawal, and D. Pesch, "FallDeFi: Ubiquitous fall detection using commodity Wi-Fi devices," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 4, pp. 1–25, Jan. 2017.

[30] D. Zhang, H. Wang, Y. Wang, and J. Ma, "Anti-fall: A non-intrusive and real-time fall detector leveraging CSI from commodity WiFi devices," in *Proc. ICOST*, Jun. 2015, pp. 181–193.

[31] S. Naribole, S. Chen, E. Heng, and E. W. Knightly, "LiRa: A WLAN architecture for visible light communication with a Wi-Fi uplink," in *Proc. IEEE SECON*, Jun. 2017, pp. 1–9.

[32] D. Cavalcanti, C. Cordeiro, M. Smith, and A. Regev, "WiFi TSN: enabling deterministic wireless connectivity over 802.11," *IEEE Commun. Stand. Mag.*, vol. 6, no. 4, pp. 22–29, Dec. 2022.

[33] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Device-free human activity recognition using commercial WiFi devices," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1118–1131, Mar. 2017.

[34] L. Shen, Q. Yang, K. Cui, Y. Zheng, X.-Y. Wei, J. Liu, and J. Han, "Fedconv: A learning-on-model paradigm for heterogeneous federated clients," in *Proc. Annu. Int. Conf. Mob. Syst., Appl. Serv.*, Jun. 2024, pp. 398–411.

[35] L. Shen, Q. Yang, X. Huang, Z. Ma, and Y. Zheng, "Gpiot: Tailoring small language models for iot program synthesis and development," in *Proc. ACM Conf. Embed. Networked Sens. Syst.*, 2025.

[36] Y. Xu, W. Yang, J. Wang, X. Zhou, H. Li, and L. Huang, "WiStep: Device-free step counting with WiFi signals," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 4, pp. 1–23, Jan. 2017.

[37] F. Wang, J. Han, F. Lin, and K. Ren, "WiPIN: Operation-free passive person identification using Wi-Fi signals," in *Proc. IEEE Global Commun. Conf.*, Dec. 2019, pp. 1–6.

[38] X. Liu, J. Cao, S. Tang, J. Wen, and P. Guo, "Contactless respiration monitoring via off-the-shelf WiFi devices," *IEEE. Trans. Mobile Comput.*, vol. 15, no. 10, pp. 2466–2479, Dec. 2016.

[39] L. Gong, W. Yang, D. Man, G. Dong, M. Yu, and J. Lv, "WiFi-based real-time calibration-free passive human motion detection," *Sensors*, vol. 15, no. 12, pp. 32 213–32 229, Dec. 2015.

[40] Y. Gu, J. Zhan, Y. Ji, J. Li, F. Ren, and S. Gao, "MoSense: An RF-based motion detection system via off-the-shelf WiFi devices," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 2326–2341, Sep. 2017.

[41] K. Yan, F. Wang, B. Qian, H. Ding, J. Han, and X. Wei, "Person-in-wifi 3d: End-to-end multi-person 3d pose estimation with wi-fi," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2024, pp. 969–978.

[42] F. Wang, S. Zhou, S. Panev, J. Han, and D. Huang, "Person-in-wifi: Fine-grained person perception using wifi," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 5452–5461.

[43] B. Lan, P. Li, J. Yin, Y. Song, G. Wang, H. Ding, J. Han, and F. Wang, "Xrf v2: A dataset for action summarization with wi-fi signals, and imus in phones, watches, earbuds, and glasses," *arXiv Prepr. arXiv:2501.19034*, Jan. 2025.

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

[45] X. Ding, T. Jiang, Y. Zhong, Y. Huang, and Z. Li, "Wi-Fi-based location-independent human activity recognition via meta learning," *Sensors*, vol. 21, no. 8, p. 2654, Feb. 2021.

[46] X. Zhang, C. Tang, K. Yin, and Q. Ni, "WiFi-based cross-domain gesture recognition via modified prototypical networks," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8584–8596, Jun. 2022.

[47] 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 *Proc. ACM Conf. Embed. Networked Sens. Syst.*, Nov. 2020, pp. 517–530.

[48] X. Zhang, Q. Hu, Z. Xiao, T. Sun, J. Zhang, J. Zhang, and Z. Li, "Few-shot adaptation to unseen conditions for wireless-based human activity recognition without fine-tuning," *IEEE. Trans. Mobile Comput.*, pp. 1–15, Sep. 2024.

[49] Y. Song, T. Wang, P. Cai, S. K. Mondal, and J. P. Sahoo, "A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities," *ACM Comput. Surv.*, vol. 55, no. 13, pp. 1–40, Jul. 2023.

[50] R. Kwitt, S. Hegenbart, and M. Niethammer, "One-shot learning of scene locations via feature trajectory transfer," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, Jun. 2016, pp. 78–86.

[51] J. Li, Z. Wang, and X. Hu, "Learning intact features by erasing-inpainting for few-shot classification," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2021, pp. 8401–8409.

[52] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021.

[53] K. I. Wang, X. Zhou, W. Liang, Z. Yan, and J. She, "Federated transfer learning based cross-domain prediction for smart manufacturing," *IEEE Trans. Ind. Informatics*, vol. 18, no. 6, pp. 4088–4096, Jun. 2022.

[54] S. Wang, J. Yue, J. Liu, Q. Tian, and M. Wang, "Large-scale few-shot learning via multi-modal knowledge discovery," in *Proc. ECCV*, Aug. 2020, pp. 718–734.

[55] A. Li, T. Luo, Z. Lu, T. Xiang, and L. Wang, "Large-scale few-shot learning: Knowledge transfer with class hierarchy," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, Jun. 2019, pp. 7212–7220.

[56] M. A. Jamal and G. Qi, "Task agnostic meta-learning for few-shot learning," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, Jun. 2019, pp. 11 719–11 727.

[57] J. Rajasegaran, S. H. Khan, M. Hayat, F. S. Khan, and M. Shah, "itaml: An incremental task-agnostic meta-learning approach," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, Jun. 2020, pp. 13 585–13 594.

[58] Y. Ma, G. Zhou, and S. Wang, "WiFi sensing with channel state information: A survey," *ACM Comput. Surv.*, vol. 52, no. 3, pp. 1–36, Mar. 2019.

[59] V. Jones and G. Raleigh, "Channel estimation for wireless OFDM systems," in *Proc. IEEE Global Commun. Conf.*, Jul. 1998, pp. 980–985.

[60] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[61] R. Vilalta and Y. Drissi, "A perspective view and survey of meta-learning," *Artif. Intell. Rev.*, vol. 18, no. 2, pp. 77–95, Jun. 2002.

[62] K. Qian, C. Wu, Z. Zhou, Y. Zheng, Z. Yang, and Y. Liu, "Inferring motion direction using commodity Wi-Fi for interactive exergames," in *Proc. Conf. Hum. Fact. Comput. Syst.*, May 2017, pp. 1961–1972.

[63] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of WiFi signal based human activity recognition," in *Proc. Annu. Int. Conf. Mobile Comput. Networking*, Sep. 2015, pp. 65–76.

[64] K. Niu, X. Wang, F. Zhang, R. Zheng, Z. Yao, and D. Zhang, "Rethinking doppler effect for accurate velocity estimation with commodity WiFi devices," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 7, pp. 2164–2178, Mar. 2022.

[65] J. Wang and J. R. Jang, "Training a singing transcription model using connectionist temporal classification loss and cross-entropy loss," *IEEE Trans. Audio Speech Lang. Process.*, vol. 31, pp. 383–396, Nov. 2023.

[66] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: gathering 802.11n traces with channel state information," *Comput. Commun. Rev.*, vol. 41, no. 1, p. 53, Jan. 2011.

[67] Z. Zhao, T. Chen, Z. Cai, X. Li, H. Li, Q. Chen, and G. Zhu, "Crossfi: A cross domain wi-fi sensing framework based on siamese network," *IEEE Internet Things J.*, pp. 1–18, Feb. 2025.

[68] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, Jun. 2018, pp. 1199–1208.

[69] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020.

[70] Y. He, J. Liu, M. Li, G. Yu, J. Han, and K. Ren, "SenCom: Integrated sensing and communication with practical WiFi," in *Proc. Annu. Int. Conf. Mobile Comput. Networking*, Oct. 2023, pp. 1–16.

[71] J. Liu, Y. He, C. Xiao, J. Han, L. Cheng, and K. Ren, "Physical-world attack towards WiFi-based behavior recognition," in *Proc. IEEE INFOCOM*, May 2022, pp. 400–409.

[72] L. Guo, S. Guo, L. Wang, C. Lin, J. Liu, B. Lu, J. Fang, Z. Liu, Z. Shan, and J. Yang, "Wiar: A public dataset for WiFi-based activity recognition," *IEEE Access*, vol. 7, pp. 154 935–154 945, Oct. 2019.

[73] Y. Ma, G. Zhou, S. Wang, H. Zhao, and W. Jung, "SignFi: Sign language recognition using WiFi," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 1, pp. 1–21, Mar. 2018.
[74] F. Wang, Y. Lv, M. Zhu, H. Ding, and J. Han, "XRF55: A radio frequency dataset for human indoor action analysis," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 8, Mar. 2024.
[75] J. Liu, C. Xiao, K. Cui, J. Han, X. Xu, and K. Ren, "Behavior privacy preserving in RF sensing," *IEEE Trans. Dependable Secur. Comput.*, vol. 20, no. 1, pp. 784–796, Jan. 2023.
[76] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

Jianwei Liu received his Ph.D. degree from the College of Computer Science and Technology, Zhejiang University, in 2024. He currently holds a Postdoctoral position with Zhejiang University and Hangzhou City University. His research interests include smart sensing, IoT, and mobile computing.

Jiantao Yuan received the B.E. degree in electronic and information engineering from Dalian University, Dalian, China, in 2009, the M.S. degree in signal and information processing from The First Research Institute of Telecommunications Technology, Shanghai, China, in 2012, and the Ph.D. degree from the College of Information Science and Electrical Engineer, Zhejiang University, Hangzhou, China.

He was with Datang mobile communication equipment co. LTD, Shanghai, China, from 2012 to 2013, where he was involved in LTE network planning and optimization. He used to work as a post-doctoral fellow at the Institute of Ocean Sensing and Networking of the Ocean College of Zhejiang University, Hangzhou, China, from 2019 to 2021. He is now working at School of Information and Electrical Engineering, Hangzhou City University, Hangzhou, China. His research interests include cross-layer protocol design, digital twin network, and unlicensed ultra-reliable low latency communications (uRLLC).

Guanding Yu (Senior Member, IEEE) received the B.E. and Ph.D. degrees in communication engineering from Zhejiang University, Hangzhou, China, in 2001 and 2006, respectively. He joined Zhejiang University in 2006, and is now a Professor with the College of Information and Electronic Engineering. From 2013 to 2015, he was also a Visiting Professor at the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. His research interests include integrated sensing and communications (ISAC), mobile edge computing/learning, and machine learning for wireless networks.

Dr. Yu has served as a guest editor of IEEE Communications Magazine special issue on Full-Duplex Communications, an Editor of IEEE Journal on Selected Areas in Communications Series on Green Communications and Networking, and Series on Machine Learning in Communications and Networks, an Editor of IEEE Wireless Communications Letters, a lead Guest Editor of IEEE Wireless Communications Magazine special issue on LTE in Unlicensed Spectrum, an Editor of IEEE Transactions on Green Communications and Networking, and an Editor of IEEE Access. He is now serving as an editor of *IEEE Transactions on Machine Learning in Communications and Networking*. He received the 2016 IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award. He regularly sits on the technical program committee (TPC) boards of prominent IEEE conferences such as ICC, GLOBECOM, and VTC. He also serves as a Symposium Co-Chair for IEEE GLOBECOM 2019 and a Track Chair for IEEE VTC 2019'Fall.

Jinsong Han (Senior Member, IEEE) received his Ph.D. degree in computer science from Hong Kong University of Science and Technology in 2007. He is now a professor at the School of Cyber Science and Technology, Zhejiang University. He is a senior member of the ACM and IEEE. His research interests focus on IoT security, smart sensing, wireless and mobile computing.