OneFi: One-Shot Recognition for Unseen Gesture via COTS WiFi

Rui Xiao[1], Jianwei Liu[1], Jinsong Han[1,2], Kui Ren[1,2]
[1]Zhejiang University, Zhejiang, China
[2]Key Laboratory of Blockchain and Cyberspace Governance of Zhejiang Province, Zhejiang, China
xiaorui1998@hotmail.com,{jianweiliu,hanjinsong,kuiren}@zju.edu.cn

ABSTRACT

WiFi-based Human Gesture Recognition (HGR) becomes increasingly promising for device-free human-computer interaction. However, existing WiFi-based approaches have not been ready for real-world deployment due to the limited scalability, especially for unseen gestures. The reason behind is that when introducing unseen gestures, prior works have to collect a large number of samples and re-train the model. While the recent advance of few-shot learning has brought new opportunities to solve this problem, the overhead has not been effectively reduced. This is because these methods still require enormous data to learn adequate prior knowledge, and their complicated training process intensifies the regular training cost. In this paper, we propose a WiFi-based HGR system, namely *OneFi*, which can recognize unseen gestures with only one (or few) labeled samples. *OneFi* fundamentally addresses the challenge of high overhead. On the one hand, *OneFi* utilizes a virtual gesture generation mechanism such that the massive efforts in prior works can be significantly alleviated in the data collection process. On the other hand, *OneFi* employs a lightweight one-shot learning framework based on transductive fine-tuning to eliminate model re-training. We additionally design a self-attention based backbone, termed as WiFi Transformer, to minimize the training cost of the proposed framework. We establish a real-world testbed using commodity WiFi devices and perform extensive experiments over it. The evaluation results show that *OneFi* can recognize unseen gestures with the accuracy of 84.2, 94.2, 95.8, and 98.8% when 1, 3, 5, 7 labeled samples are available, respectively, while the overall training process takes less than two minutes.

CCS CONCEPTS

• **Human-centered computing → Ubiquitous and mobile computing systems and tools**.

KEYWORDS

Human Gesture Recognition; WiFi Sensing; Few-Shot Learning

ACM Reference Format:
Rui Xiao[1], Jianwei Liu[1], Jinsong Han[1,2], Kui Ren[1,2]. 2021. OneFi: One-Shot Recognition for Unseen Gesture via COTS WiFi. In *The 19th ACM Conference on Embedded Networked Sensor Systems (SenSys '21), November 15–17, 2021, Coimbra, Portugal.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3485730.3485936

SenSys '21, November 15–17, 2021, Coimbra, Portugal
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-9097-2/21/11…$15.00
https://doi.org/10.1145/3485730.3485936

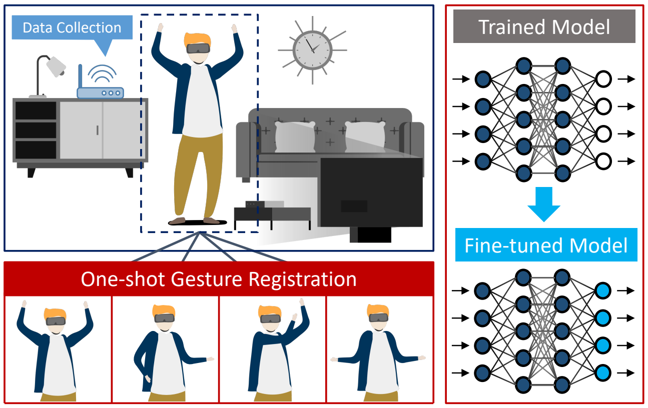


Figure 1: A typical scenario of *OneFi*: a user is playing VR games using WiFi. The user registers several unseen gestures by performing each gesture once. The trained model is then fine-tuned using these one-shot data.

1 INTRODUCTION

Human gesture recognition (HGR) is a key enabler for many human-centered applications, such as virtual reality, smart home, and elderly health care [55]. Traditional HGR solutions are mainly camera-based [12, 24, 43] or device-based [2, 14, 35]. Both of them have their respective shortcomings. Although human gestures can be accurately identified from captured images or videos, cameras would reveal human's sensitive information, e.g., facial information, raising the concern of privacy leakage. The device-based approaches, e.g. wearable sensors, are usually inconvenient for users. It is increasingly important to develop non-intrusive HGR systems that support contactless, lightweight, and ubiquitous monitoring and identification of human gestures. The COVID-19 pandemic again emphasizes the demand for such a contactless interface to help avoid infection [41]. Compared with traditional approaches, WiFi-based solutions are promising to fulfill the above needs [19, 39, 40, 55].

Generally, a WiFi-based HGR system works in three phases, i.e., gesture class definition, training data collection, and recognition model training and testing. In the first phase, users predefine several classes of gestures to recognize (e.g., three classes for label input: '*push*', '*clap*', and '*slide*'). These predefined gesture classes are termed as base classes. In the second phase, a large number of signal samples for each base class are collected to build a training

Jinsong Han is the corresponding author.

set. A recognition model is then trained based on this training set in the third phase. The classifier can provide accurate recognition on these base gestures once well trained.

However, existing WiFi-based HGR approaches face a common drawback. Their scalability is severely limited for *unseen gesture classes*, which are the ones not included in the base dataset. This weakness greatly impedes their real-world deployment because predefined base gestures cannot meet the demand of recognizing increasingly ample gestures. Their limited scalability results in two kinds of overhead when trying to involve unseen gesture classes. ***(1) Data Collection Overhead.*** Users have to collect sufficient signal samples of each unseen class. Considering that collection overhead is linear to the number of unseen classes, it will bring a massive inconvenience to users, consuming numerous manpower and system cost. ***(2) Training Overhead.*** The whole model has to be re-trained using all training data once a single unseen class is introduced. Retraining an HGR model is non-trivial because it usually involves a recurrent structure [19, 20, 29, 55] to perceive the complex and abstract temporal-spatial characteristics in WiFi gesture signals, which is time-consuming due to its inherent sequential-processing characteristic.

Hence, there is an urgent demand for one-shot unseen gesture recognition systems. In such a system, users only need to collect one signal sample (i.e. one shot) for any unseen class, and the model only needs to be fine-tuned using these one-shot samples. These characteristics would significantly increase the system scalability and thus make fast deployment possible. In the literature, this problem falls into the category of *few-shot learning* (FSL) [50].

Recent advances in mobile sensing field [7, 13] come up with *meta-learning algorithms* [17] as their FSL solutions. However, they cannot settle the above two kinds of overhead fundamentally: *(1) Data collection overhead persists.* The principle of meta-learning to recognize unseen classes is to bring adequate prior knowledge from an enlarged base dataset, which could be prohibitively huge to collect. Hence, instead of completely solving the data collection overhead, meta-learning shifts this overhead from unseen gesture collection to base dataset collection. *(2) Training overhead remains.* While meta-learning architecture mitigates the training overhead in unseen gestures recognition, it multiplies the complexity of the regular training process. In this paper, we propose *OneFi*, a novel WiFi-based one-shot HGR system to recognize unseen gestures. We solve these two kinds of overhead at a fundamental level.

To tackle the data collection overhead, we propose a *virtual gesture generation* approach. Instead of collecting a large base dataset like meta-learning, we generate additional, synthetic data by signal modeling. The key property of our method is that the *virtual gestures* are derived from existing gestures yet nearly identical to the real signal samples that would result from real users performing that gesture. Consequently, instead of requiring a large base dataset to learn sufficient prior knowledge, *OneFi* only requires a small base dataset to learn a representative feature of WiFi data, alleviating the data collection overhead.

To overcome the training overhead, we design a lightweight one-shot learning framework based on transductive fine-tuning [37]. The proposed framework is composed of a powerful feature extractor and a simple classifier. The core principle is that once the feature extractor is trained on a base dataset, only the classifier needs to be fine-tuned with the collected one-shot samples, eliminating the effort to re-train the whole network. Besides, to reduce the overhead of training the feature extractor with the base dataset, we design a self-attention [38] based backbone, called WiFi Transformer. Without using sequence-aligned recurrent architecture, e.g. recurrent neural network (RNN), WiFi transformer is entirely built on the self-attention mechanism. In terms of computational complexity, self-attention layers are faster than recurrent layers because they connect all elements in time series with a constant number of sequentially executed operations [38].

We evaluate *OneFi* through extensive real-world experiments by recruiting 10 participants that perform 40 different gestures for a total of more than 2900 times. The results demonstrate that after being trained for two minutes, *OneFi* can recognize new gestures with an average accuracy of 84.2%, while only one training sample of each class is provided. It also outperforms the state-of-the-art few-shot learning solutions by 25.6% and 33.6% in recognition accuracy in the one-shot scenario. In summary, our contributions are as follows:

- We present a novel virtual gesture generation technique that significantly alleviates the data collection overhead.
- We propose a lightweight one-shot learning framework using transductive fine-tuning, eliminating the overhead of re-training the whole model.
- We design WiFi Transformer, a self-attention based backbone, to substitute traditional recurrent architecture. Compared with $O(N)$ sequential operations in RNN, WiFi Transformer only needs $O(1)$ sequential operations because it computes all hidden representations in parallel.
- We evaluate our implementation by conducting extensive real-world experiments to demonstrate its strong ability to recognize unseen gestures.

By means of this work, we also hint at a new avenue of learning-based HGR systems. Instead of trying to bypass the complexity of RF signals, we utilize it and construct additional data, alleviating the data collection overhead in traditional learning-only systems. By combining the power of signal modeling and deep learning, we establish a new state-of-the-art for one-shot gesture recognition.

2 PROBLEM STATEMENT

In this section, we formulate our target problem. We state the motivation and importance to recognize unseen gestures in Section 2.1. Then, we formally define the problem of one-shot unseen gesture recognition in Section 2.2. In Section 2.3, we dig the reason why existing FSL approaches cannot afford solutions for this problem.

2.1 Why do Unseen Gestures Matter?

While there are existing works focusing on adapting WiFi HGR systems to new environments [7, 19, 54], new users [47], different locations and different orientations [40, 55], it is still a missing piece of the puzzle to adapt the system to *unseen gestures.*

However, the unseen gestures matter because it is the performance on unseen gestures that directly determines the *scalability* of a system. It impacts how well a system will keep up with ever-evolving demands from the user and real-world application. Figure 1 shows a typical scenario. On the one hand, from a user-centered perspective, it is crucial to allow the user to adapt the system to their own preference, exercising control over the recognizable gesture set with minimized updating cost. On the other hand, the increasingly expanded spectrum of gestures stresses the point again that it is important to achieve efficient adaption to a large scale of unseen gestures with a small number of samples. In view of this, we need such a system that can adapt to unseen gestures with both the gains of scalability and efficiency.

2.2 Problem Definition

Objective. This paper aims to design a WiFi-based gesture recognition system that can recognize *unseen gestures* using one or few labeled samples of these unseen gestures.
Assumption. We assume that the system has already had a labeled dataset (called base dataset) of a set of gesture classes (called base classes). This base dataset only needs to be collected once to initialize the system.
Requirement 1. We assume the base dataset does not include any unseen gestures for testing because, in that case, the problem becomes a common supervised learning problem, which is not the objective of our paper.
Requirement 2. To reduce the data collection overhead for real-world deployment, the base dataset should not include too many ($\leq$ 20) base classes.
Requirement 3. To further reduce the data collection overhead, we assume that it is not necessary for the base dataset to be collected in the same domain of the incoming unseen gestures. That is, this base dataset can be collected at any environment, position, orientation, and by any person.
Requirement 4. In order to minimize the training overhead, the time for model training should be within minutes.
Summary. In brief, our ultimate goal is to use prior knowledge embedded in a relatively small base dataset to build a model that can be adapted to infinity possible unseen gestures, with minimized data collection and training overhead.

2.3 Prior FSL Solutions and Limitations

Few-shot learning (FSL) is proposed to tackle the problem of adapting models to new tasks and environments when only limited data is available [50]. As a typical FSL problem, few-shot classification aims to learn classifiers when only a few labeled samples of each class are given. With the WiFi HGR scenario in mind, We formulate the problem as an N-way-K-shot classification if the training set contains $J = K \times N$ samples from N classes, each with K samples. Specifically, if there is only one sample for each class (i.e. N-way-1-shot), FSL is also called *one-shot learning*. The training set and test set are called *support set* $\mathcal{D}^{support}$ and *query set* $\mathcal{D}^{query}$ respectively in a few-shot classification context.

Common supervised learning approaches fail to solve the FSL problem because the scarcity of labeled data usually leads to overfitting. The core principle of existing FSL methods is to introduce adequate prior knowledge into the model. For example, towards the HGR task, while we do not have sufficient labeled data on unseen gestures, we have sufficient labeled data for a set of base gestures. FSL algorithms combine the available supervised information of unseen classes with prior knowledge from base classes, which is available before $\mathcal{D}_{support}$ is given, to improve the overall performance of unseen gesture recognition.

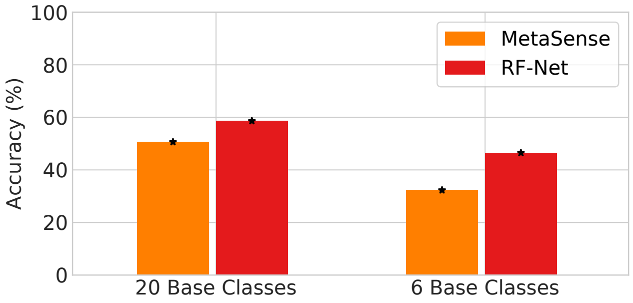


Figure 2: Accuracy of MetaSense and RF-Net to recognize unseen gestures. Neither of them can achieve acceptable accuracy with a small base dataset.

Prior works [7, 13] adopt *meta-learning* [17] as their FSL solution to adapt their sensing system to new users, new devices, and new environments. Known as learning-to-learn [17], meta-learning is a promising paradigm of few-shot classification where a machine learning model gains experience from a collection of related tasks and uses this experience to extract transferable knowledge. A collection of meta-training tasks $\mathcal{T} = \{\mathcal{D}_i^{support}, \mathcal{D}_i^{query}\}_{i=1}^{I}$ is sampled from the base classes $\mathcal{D}_{base}$ to mimic the process of few-shot learning. A meta-learner $\mathcal{A}(\cdot \mid \phi)$ is defined to learn a general-purpose learning algorithm that can generalize across tasks. Then, the optimizing process for a given task set $\{\mathcal{D}_i^{support}, \mathcal{D}_i^{query}\}$ can be formulated as:

$$\begin{aligned}\theta^* &= \mathcal{A}\left(\mathcal{D}_i^{support}; \theta \mid \phi\right)\\ &= \arg\min_{\theta} \mathcal{L}\left(\mathcal{D}_i^{support}; \theta \mid \phi\right),\end{aligned} \tag{1}$$

where $\mathcal{L}$ is a loss function to measure the correctness of the backbone classification model f_θ.

In the meta-training (i.e. 'learning how to learn') process, the optimization objective is to minimize the average test error of the meta-learner $\mathcal{A}(\cdot \mid \phi)$ on the sampled task distribution $\mathcal{T}$:

$$\phi^* = \arg\min_{\phi} \mathbb{E}_{\mathcal{D}^{query} \in \mathcal{T}} \left[\mathcal{L}\left(\mathcal{D}^{query}; \theta \mid \phi\right)\right]. \tag{2}$$

Then, given a new task $\mathcal{D}_{new} = \{\mathcal{D}_{new}^{support}, \mathcal{D}_{new}^{query}\}$, we use the learned meta-knowledge $\mathcal{A}(\cdot \mid \phi^*)$ to train the base model:

$$\theta^*_{NewTask} = \arg\min_{\theta} \mathcal{L}\left(\mathcal{D}_{new}^{support}; \theta \mid \phi^*\right). \tag{3}$$

Based on the above general scheme, meta-learning methods could be further categorized into three types: metric-based, model-based, and optimization-based [22]. Specifically, MetaSense [13] applies an optimization-based approach to deep mobile sensing system by learning a decent model initialization (i.e. the parameters of a network) so that the classifiers can be tuned for a new user/device with a limited number of labeled samples and a small number of gradient update steps. RF-Net [7] employs a metric-based solution and focuses on adapting its RF sensing system to new environments.

Limitations. However, these FSL approaches cannot achieve our goal to recognize unseen gestures. This is because these sophisticated meta-learning approaches do not provide much performance improvement without collecting a huge base dataset [4]. To further illustrate this limitation, we demonstrate the 6-way-1-shot performance of the state-of-the-art FSL approaches, MetaSense and RF-Net, in Figure 2. When there are 20 gesture classes in the base dataset, the accuracy of MetaSense and RF-Net is 50.6% and 58.6%, respectively. If there are only six gesture classes in the base dataset, the accuracy drops to 32.3% and 46.5%, respectively. Furthermore, as aforementioned, meta-learning intensifies the regular training cost. We evaluate their time costs in Section 6.8. In summary, their performance to recognize unseen gestures is unacceptable for real-world deployment. The deficiency of existing few-shot learning solutions asks for a new type of FSL scheme to recognize unseen gestures.

3 SYSTEM OVERVIEW

OneFi is a one-shot recognition system for unseen gestures boosted by virtual gesture generation and transductive fine-tuning. Figure 3 depicts the overview of our system, which contains four major modules: data collection, data pre-processing, virtual gesture generation, and transductive fine-tuning. We release our code[1] to foster reproduction.

(1) Data Collection. To perceive human gestures with WiFi signals, we first collect the *channel state information* (CSI) from WiFi signals. After taking the multipath effect into consideration, each CSI entry [32] with carrier frequency f_c could be formulated as:

$$H(f_c, t) = \left(\sum_{k=1}^{K} \alpha_k(t) e^{-j2\pi f_c \tau_k(t)} \right) e^{j\epsilon(f_c, t)}, \tag{4}$$

where K is the total number of multipath components, and α_k and τ_k are the amplitude attenuation factor and the propagation delay for the k-th path, respectively. $e^{j\epsilon(f_c,t)}$ is the phase offset caused by timing alignment offset, sampling frequency offset and carrier frequency offset.

(2) Pre-processing. In the pre-processing module, we compute Doppler spectrogram of CSI data from each receiver. The concatenated Doppler spectrograms serve as the input of the recognition module. The Doppler frequency shift (DFS) of a signal is the change in the length of the signal propagation path $d(t)$ [33]:

$$f_D(t) = -\frac{1}{\lambda} \frac{d}{dt} d(t), \tag{5}$$

where λ is the wavelength of transmitted signal. Then, the received CSI data can be modeled as:

$$H(f_c, t) = \left(H_s(f_c) + \sum_{k \in P_{dn}} \alpha_k(f_c, t) e^{j2\pi \int_{-\infty}^{t} f_{D_k}(u) du} \right) e^{j\epsilon(f_c, t)}, \tag{6}$$

where $H_s(f_c)$ represents the sum of signals collected from static paths, whose DFS is zero, and P_{dn} is the set of dynamic paths corresponding to the signals reflected by the moving human body.

Figure 4 shows our pre-processing workflow to extract Doppler spectrogram from raw CSI data. We present two subcarriers among all the 90 subcarriers for simplicity. We first remove the DC offset from the CSI of each subcarrier. Then, we remove the phase offset by calculating the conjugate multiplication of CSI readings from another antenna on the same WiFi NIC and remove the static components $H_s(f_c)$ using a high pass filter [25]. We also apply a low pass filter to remove the high frequency noise. After that, we apply a principal component analysis (PCA) on CSI streams to further denoise the signal so that only prominent dynamic components are retained [45]. Finally, we conduct a short-time Fourier transform to extract the Doppler spectrogram.

(3) Virtual Gesture Generation (Section 4). Motivated by data augmentation in computer vision research, we design a data construction mechanism for the WiFi HGR task to strengthen the generalization ability of our deep model and mitigate the data collection overhead. The intuition is that we can create a '*push left*' gesture by transforming a '*push forward*' gesture. Similar to data augmentation, we generate additional, synthetic data, called *virtual gestures*, from the base samples we have.

While data augmentation methods are popular in computer vision field, there is no existing counterpart on WiFi data. The reason is that the data augmentation methods, such as cropping, zooming, rotation, and the like, cannot be directly applied to WiFi data because they are inherently semantically different from vision data.

We achieve this through rigorous signal modeling. First, we recover body movement velocity information from multiple Doppler spectrograms (i.e. the output of the pre-processing module) of a given gesture using non-linear optimization. Then, we generate Doppler spectrogram of a transformed gesture, i.e. the virtual gesture, by mapping each velocity component to the corresponding Doppler frequency component. We can generate multiple virtual gestures from each gesture sample in the base dataset.

(4) Transductive Fine-tuning (Section 5). Recent works propose that the learned representation, instead of the meta-learning algorithm, is responsible for the fast adaption to test time tasks [37]. To alleviate training overhead, we approach this problem with a lightweight two-stage transductive fine-tuning solution instead of designing another sophisticated meta-learning framework. In the first stage, we train a feature extractor on base dataset to learn a representation of gesture data by feeding the labeled augmented base classes, i.e. virtual gestures, into the model. When doing one-shot classification in the second stage, we fine-tune a classifier to compare the cosine distance between the extracted features of the support and query samples to get the prediction result. To further reduce training overhead, we innovatively tailor a self-attention model, called WiFi Transformer (Section 5.2), as the backbone of our feature extractor to perceive the temporal information in the Doppler spectrogram.

4 VIRTUAL GESTURE GENERATION

In this section, we detail our virtual gesture generation process. In Section 4.1, an intuitive explanation of this process is given. In Section 4.2, we introduce velocity distribution, a feature that describes the body movement information of a gesture. We compute velocity distribution by non-linear optimization. In Section 4.3, we generate virtual gestures by rotating the computed velocity

[1] https://github.com/ruixiao24/onefi

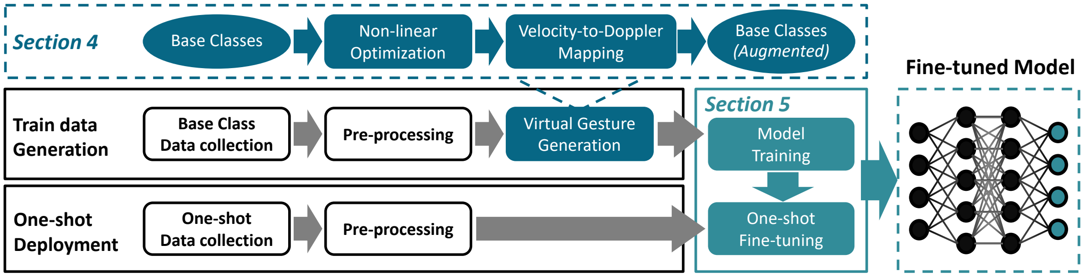


Figure 3: Figure depicts the overview of *OneFi*. For train data generation, we collect a set of base classes and leverage virtual gestures to further augment the dataset. By training the model with this augmented base dataset and fine-tuning it with one-shot data of unseen gestures, we obtain an fine-tuned model which can recognize these unseen gestures.

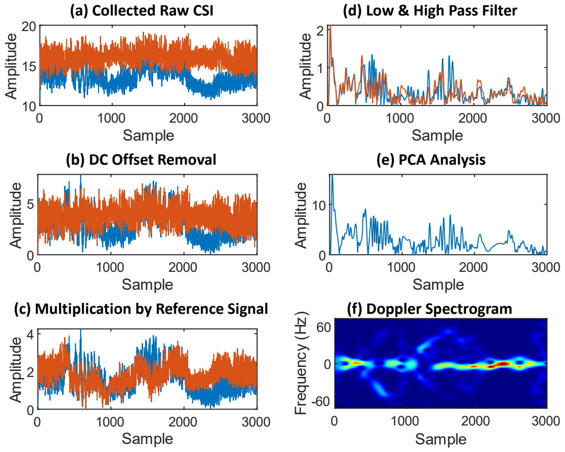


Figure 4: Pre-processing workflow of *OneFi*.

distribution and projecting that back to the Doppler frequency domain.

4.1 Intuitive Explanation

Virtual gesture generation is essentially simulating Doppler spectrograms of the gestures that do not exist in the base dataset by transforming existing gestures. The intuition here is that we can create a *'push left'* gesture by *rotating* the *'push forward'* gesture. To achieve this, we need to know the body movement information of the gesture.

Multiple Doppler spectrums are required to reconstruct the body movement information [55]. To intuitively explain this, an analogy would be photography. When a picture is taken by a camera, the 3D world is projected into a 2D image, with the loss of depth information. Similarly, when the Doppler spectrum is captured by a WiFi receiver, the velocity information is projected into a lower dimension, with the loss of some spatial information. Therefore, we need multiple Doppler spectrums to reconstruct the body movement information.

To formally describe the body movement information, we propose *velocity distribution* as the bridge between the source gesture (e.g. real gesture) and rotated gesture (e.g. virtual gesture). By reconstructing the *velocity distribution* from multiple Doppler spectrums, we could generate virtual gestures by projecting the rotated velocity distributions back to Doppler frequency domain.

4.2 Velocity Distribution

Definition of Velocity Distribution: Assuming a human body is composed of infinite moving segments and the velocity of each segment can be described by a two-dimensional random variable $v = (v_x, v_y)$. Then, the velocity distribution $P(v, t)$ is the probability distribution of body segment velocity v at the given time point t.

Velocity distribution is similar to a feature called *body-coordinate velocity profile* (BVP) proposed in the previous work [55], which describes the signal power distribution over different velocity components. The velocity distribution in our paper is different from BVP in that velocity distribution is a continuous model while BVP offers a discrete model. Specifically, BVP discretizes the velocity space while the velocity distribution is instead modeled on continuous velocity space. Therefore, velocity distribution provides a better physical picture and is more intuitive and explainable.

(1) Velocity-Doppler Mapping. In order to compute velocity distribution from multiple Doppler readings, we first compute the mapping f_D between velocity v and Doppler frequency shift. With a fixed WiFi transmitter, the mapping f_D [36] can be given by:

$$f_D(v) = \frac{|v|}{c} f_c cos\theta, \tag{7}$$

where c is the speed of light in the medium, v is the velocity of the receiver relative to the medium, f_c is the central carrier frequency, and θ is the angle between the propagation direction of RF signal and movement direction of the receiver.

In the HGR setting, the transmitter and receiver do not move. Instead, it is the body movement, acting as a moving reflector, that leads to the Doppler frequency shift. The Doppler frequency shift with a moving reflector is:

$$f_D(v) = \frac{|v|}{c} f_c (cos\theta_r + cos\theta_t), \tag{8}$$

where θ_r and θ_t are the angles between the RF signal propagation direction and the body movement direction relative to the receiver and the transmitter, respectively. Suppose that the relative locations of the transmitter and the receiver with respect to the user body are $l_t = (x_t, y_t)$, $l_r = (x_r, y_r)$. We define:

$$a_x = \frac{x_t}{\|l_t\|^2} + \frac{x_r}{\|l_r\|^2}, \quad a_y = \frac{y_t}{\|l_t\|^2} + \frac{y_r}{\|l_r\|^2}. \tag{9}$$

Then, the Doppler frequency shift can be written as:

$$f_D(v) = \frac{1}{\lambda}(a_x v_x + a_y v_y), \tag{10}$$

where λ is the wavelength of transmitted WiFi signal.

(2) Velocity Distribution Computation. We now give the relationship between Doppler spectrum D and velocity distribution $P(v,t)$. When the positions of transmitter, receiver, and user are known, every velocity v maps to a Doppler frequency shift $f_D(v)$ following Eq. 10. Then, the Doppler spectrum D at the time t can be formulated as:

$$D(f,t) = \sum_{f_D(v)=f} P(v,t), \tag{11}$$

where f represents different frequency components.

Since multiple receivers are used to capture WiFi signals in our setting, we obtain multiple Doppler spectrum data D^m_{real} from these receivers where $m = 1, \dots, N_R$ and N_R is the number of receivers. To get the best approximate solution of $P(v,t)$, we track it as a non-linear optimization problem [55]. The optimization objective is:

$$\min_{P(v,t)} Q(P) \quad \text{s.t. } P(v,t) \geq 0, \tag{12}$$

where

$$Q\left(P(v,t)\right) = \sum_{m=1}^{N_R} \text{EMD}\left(\sum_{f_D^m(v)=f} P(v,t), D^m_{real}\right). \tag{13}$$

In the above equation, EMD$(\cdot,\cdot)$ is the Earth Mover's Distance [34] between two distributions.

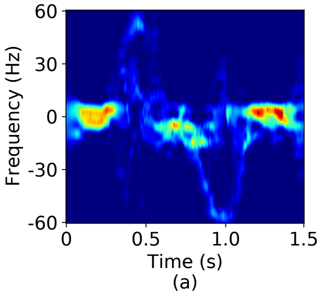
To solve this optimization problem, we first specify a discrete grid of samples that provides local averages of $P(v,t)$ over neighborhoods of size proportional to m. Thus, $P(v,t)$ is discretized as a k-by-k matrix $P_d(t)$ where $k = \frac{V_m}{m}$ and V_m is the maximum velocity bound which is an empirical parameter. Then, the problem becomes $min_{P_d(t)} Q(P_d(t))$ where $Q : \mathbb{R}^{k\times k} \mapsto \mathbb{R}$. We leverage *interior-point methods* [31] to solve this optimization problem and get optimized $P_d(t)$, which is a discrete approximation of velocity distribution $P(v,t)$.

4.3 Augmented Dataset Generation

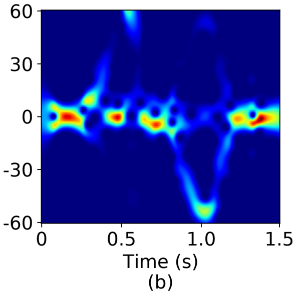
WiFi gesture data is essentially time series data collected at T consecutive timestamps. We call the data at each timestamp as a time patch. To obtain the body movement information of a given gesture, we have to compute the velocity distributions of all time patches. The computed velocity distribution set is denoted as $\mathcal{P} = \{P(t_i)\}_{i=1}^T$, where $P(t_i)$ is the velocity distribution at the time t_i. We omit the variable v in $P(v,t)$ for simplicity.

After computing $\mathcal{P}$, we can create virtual gestures by rotating $\mathcal{P}$ around its center point. We use nearest neighbor interpolation when doing rotation. The rotated velocity distribution set is denoted as $\mathcal{P}' = \{P'(t_i)\}_{i=1}^T$, where $P'(t_i)$ is the rotated velocity distribution at the time t_i. Then, the Doppler spectrogram of the virtual gesture is $\{D'(f,t_i)\}_{i=1}^T$ where $D'(f,t_i) = \sum_{f_D(v)=f} P'(t_i)$ (following Eq. 11). As a result, we can generate virtual gestures by transforming the available data.

Figure 5 (a) shows the Doppler spectrogram of a virtual gesture, '*push left*'. We generate this virtual gesture by rotating the gesture '*push forward*' counterclockwise by 45 degrees. Figure 5 (b) shows the Doppler spectrogram of '*push left*' collected in real experiment. Figure 5 (a) and (b) look similar, supporting our statement that the created data is of high quality from the training perspective.



(a)



(b)

Figure 5: Doppler spectrogram of the gesture '*push left*'. (a) is a generated *virtual gesture*. (b) is a real data collected by a WiFi receiver.

Suppose that our base dataset $\mathcal{D}_{base}$ is composed of N_g different gesture classes so that $\mathcal{D}_{base} = \{\mathbf{x}_i, y_i\}$, where $y_i \in Y$ and $|Y| = N_g$. We now expand the base dataset by applying the generation process on each gesture by K times (i.e. rotating each gesture by K different degrees). In this way, we can generate an augmented base dataset $D'_{base} = \{\mathbf{x}_i, y_i\}$ where $y_i \in Y_{new}$ and $|Y_{new}| = N_g \times K$. In our experiment, we expand the base dataset by 12 times and improve the accuracy greatly.

5 FEW-SHOT RECOGNITION MECHANISM

Our transductive fine-tuning based few-shot recognition mechanism can be divided into two parts: training a feature extractor using the augmented base dataset (i.e. virtual gestures) and fine-tuning a feature classifier using one-shot data. In this section, we first explain this mechanism in Section 5.1. Then, we detail the design of our novel feature extractor, i.e. WiFi Transformer, in Section 5.2.

5.1 Transductive Fine-tuning

As shown in Figure 6, we adopt a two-stage transductive fine-tuning approach as our one-shot gesture recognition framework. Let $(\mathbf{x}, y)$ denote an input gesture vector and its ground-truth label. Then, the base dataset $\mathcal{D}_{base}$ can be denoted as $\mathcal{D}_{base} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$. In the first stage, we train a powerful feature extractor using the data in $\mathcal{D}_{base}$. In the second stage, we fine-tune a classifier based on cosine similarity to predict the label of query samples.

In the first stage, our goal is to train a feature extractor, which can give a high-level representation of input signal samples, using data in $\mathcal{D}_{base}$. Instead of splitting the base dataset $\mathcal{D}_{base}$ into multiple sub-tasks like meta-learning, we train the feature extractor using all the available data in $\mathcal{D}_{base}$ as a single task. The trained feature extractor can map the input into an embedding space and thus provides a much meaningful cosine similarity. Our approach is to train the feature extractor f_θ together with a classifier $C(\cdot|W,b)$ on $\mathcal{D}_{base}$ as a multi-classification task. The classifier C consists of a weight term W and a bias term b. We design WiFi Transformer as

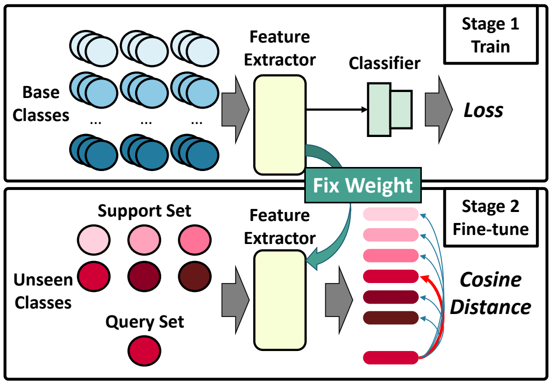


Figure 6: Figure depicts our few-shot recognition mechanism. In the first stage, we train a feature extractor along with a classifier using available base classes. In the second stage, we compare the cosine similarity of extracted features between query sample and support samples by fine-tuning a new classifier to predict the label of query sample.

our feature extractor (see Section 5.2). We use cross-entropy as our loss function $\mathcal{L}$. Then, the optimized parameters are:

$$\begin{aligned}\theta^*, W^*, b^* &= \underset{\theta,W,b}{\arg\min}\, \mathcal{L}(\mathcal{D}; \theta, W, b) \\ &= \underset{\theta,W,b}{\arg\min} \sum_{(\mathbf{x},y)\in\mathcal{D}} -\log\left(W^T f_\theta(y|\mathbf{x}) + b\right).\end{aligned} \tag{14}$$

In this way, we obtain a trained feature extractor f_{θ^*}.

In the second stage, a new similarity-based classifier is fine-tuned. This classifier can estimate the class label of query sample based on the cosine similarity of the extracted features between the query sample and the labeled support samples. We implement this by replacing the classifier C with a weight matrix $W \in \mathbb{R}^{d\times c}$. The weight matrix W can be written as $[w_1, w_2, \ldots, w_c]$, where each class has a d-dimensional weight vector. We fine-tune this weight matrix W using support samples while keeping the entire feature extractor f_θ fixed. To classify query samples, we compute its cosine similarity to each weight vector and obtain the similarity scores $[s_1, s_2, \ldots, s_c]$ for all classes, where

$$s_j = \frac{f_\theta(\mathbf{x}) \cdot w_j}{\|f_\theta(\mathbf{x})\|\|w_j\|}. \tag{15}$$

Note that we only need to run the first stage once. After we finish training our feature extractor f_θ at one place, we can ship our system to any other place and register new gestures by running the second stage.

5.2 WiFi Transformer

In *OneFi*, we design *WiFi Transformer*, a self-attention based model, to come up high-level representation of the Doppler spectrogram. Doppler spectrogram is essentially a sequence data because it represents the Doppler spectrum varying with time. We use self-attention, a recent advance to capture long-range interactions in sequential data [38], to serve as the primary primitive of our model.

Figure 7 illustrates the overall structure of the WiFi Transformer. We apply a linear projection and position embedding on the input sequence. Then, we stack alternating layers of multi-head self-attention blocks and fully-connected feed-forward blocks sequentially as the Transformer encoder to model long-term dependencies among all the time patches. Each block has a residual connection [16], followed by layer normalization [1]. After processing with the Transformer encoder, we obtain a high-level representation of the input sequence.

We use the following naming conventions: F and C refer to the number of frequency bins [9] and the number of input channels. Note that every single receiver maps to one channel. T refers to the number of timestamps.

5.2.1 Model inputs. The input of WiFi Transformer is the concatenated Doppler spectrogram $\mathbf{x} = (x_1, \ldots, x_T)$, where $x_i \in \mathbb{R}^{F\times C}$, $i = 1, \ldots, T$ is the received Doppler spectrum from C receivers at each timestamp. Note that each x_i is a time patch of $\mathbf{x}$. We first flatten each x_i and apply a linear projection with a parameter matrix $W_l \in \mathbb{R}^{(F\cdot C)\times d}$ on each time patch to get an embedding of each x_i with a fixed length d. Now the input $\mathbf{x} \in \mathbb{R}^{T\times d}$.

A special classification token [CLS] of shape $\mathbb{R}^d$ is attached to the beginning of embedded time patches to represent the meaning of the entire sequence [6]. The final hidden state (i.e. the output of the Transformer encoder) of the [CLS] token serves as the fixed-dimensional feature of the input sequence.

5.2.2 Multi-head self-attention block. Self-attention blocks aim to model long-distance interactions of features received at different time patches [38]. We apply multi-head attention with h attention heads, in which the self-attention function is calculated for h times. The multi-head attention increases the model's ability to focus on different positions in the sequence, and it also gives the attention layer multiple different representation sub-spaces [38].

The multi-head mechanism splits the inputs into smaller chunks and then computes the scaled dot-product attention over each subspace in parallel. Given the input $\mathbf{x} = (x_1, \ldots, x_T)$, each attention head outputs a new sequence $\mathbf{z} = (z_1, \ldots, z_T)$ where $z_i \in \mathbb{R}^{d/h}$. Concatenating all the output sequence $\mathbf{z}$ of each head, the final output of a multi-head self-attention block is $\mathbf{x^o} = (x_1^o, \ldots, x_T^o)$ where $x_i^o \in \mathbb{R}^d$.

When doing computation for each head, the output element, z_i, is a weighted sum of the input value vector, which can be written as:

$$z_i = \sum_{j=1}^{n} \alpha_{ij}(x_j W^V), \tag{16}$$

where each weight coefficient α_{ij} is determined by a compatibility function that compares two input elements:

$$\begin{aligned}\alpha_{ij} &= \text{softmax}\left(\frac{(x_i W^Q)(x_j W^K)^\intercal}{\sqrt{d/h}}\right) \\ &= \frac{\exp\left(\frac{(x_i W^Q)(x_j W^K)^\intercal}{\sqrt{d/h}}\right)}{\sum_{k=1}^{T} \exp\left(\frac{(x_i W^Q)(x_k W^K)^\intercal}{\sqrt{d/h}}\right)}.\end{aligned} \tag{17}$$

Note that $W^Q, W^K, W^V \in \mathbb{R}^{d\times\frac{d}{h}}$ in Eq. (17) are the trainable query matrix, key matrix, and value matrix, respectively. They are unique across different self-attention blocks and attention heads.

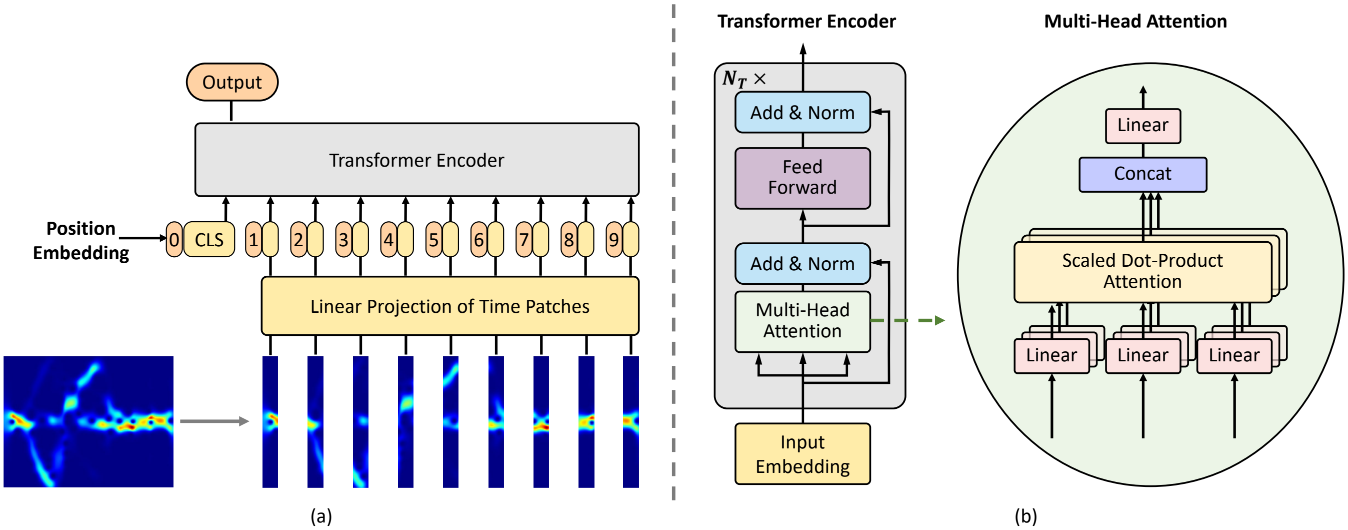


Figure 7: Figure depicts our feature extractor, WiFi Transformer. (a) shows the overview of WiFi Transformer. After applying a linear projection and position embedding on the input time patches and then feeding them into the transformer encoder, we take the final hidden state of the [CLS] token as the representation of the whole sequence. (b) shows components of the transformer encoder, which stacks alternating layers of multi-head self-attention blocks and fully-connected feed-forward blocks sequentially.

By grouping the queries ($x_i W^Q$), keys ($x_i W^K$) and values ($x_i W^V$) in Q, K, V matrices, the self-attention computation can be done for the entire input sequence in parallel:

$$\mathbf{z} = Attention(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d/h}})V. \quad (18)$$

5.2.3 Position embedding. WiFi gesture data are basically time series data. Therefore, the position and order of time patches are the essential components of the HGR task. Because the self-attention operation is permutation invariant, the Transformer architecture itself does not have any sense of the position for each patch. Therefore, we add a learnable position embedding $\mathbf{p} = (p_1, \ldots, p_T)$, where $p_j \in \mathbb{R}^d$, to each patch embeddings to retain the absolute position information. After the position embedding module, we obtain the position-encoded input element representation $\mathbf{x}' = (x + p_1, \ldots, x_T + p_T)$.

5.2.4 Model outputs. After the sequence of the Transformer encoder, a set of high-level features $\mathbf{x}^{\mathrm{f}} \in R^{T \times d}$ can be inferred. Remember that we attach a [CLS] token at the beginning of input sequence as the representation of the whole input sequence (see Section 5.2.1). Now, we take the first element in output feature sequence $\mathbf{x}^{\mathrm{f}}$ as the output of the WiFi Transformer. Note that in the fine-tuning stage, we use the intermediate hidden state of [CLS] token as the output feature rather than the final hidden state because the final hidden state might be overfitted on the base dataset.

6 EVALUATION

In this section, we present our real-world implementation and detail the performance of *OneFi*.

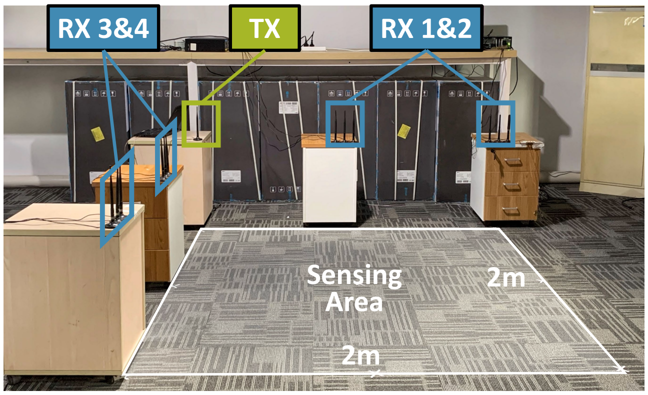


Figure 8: Experiment Setup

6.1 Experiment Methodology

Experiment setup: We use commercial off-the-shelf (COTS) devices to build the prototype of *OneFi*. As shown in Figure 8, we conduct basic experiments in a laboratory environment. The sensing area is a 2m × 2m square, which is a typical setting to perform interactive gestures for recognition and response [55]. The prototype consists of one transmitter and four receivers, all of which are equipped with Intel 5300 network interface cards. We use *Linux 802.11n CSI Tool* [15] on Ubuntu 14.04 to extract CSI values from WiFi packets on channel 108 at 5.54 GHz, where there is little interference from other devices. The transmitter sends WiFi packet every one millisecond, and the receivers working in the monitor mode collect the CSI values. We activate one antenna of the transmitter and all the three antennas of the receivers, and thus we obtain a

CSI measurement of $1 \times 3 \times 30 = 90$ subcarriers from each WiFi packet.

Dataset: We recruit 10 participants (6 males and 4 females) to participate in our experiments. We conduct the experiments by adhering to the approval of our university's Institutional Review Board (IRB). The participants are asked to perform the gestures in the sensing area. We totally collect more than 2900 signal samples of 40 different gesture classes, in which six gestures (*'push and pull'*, *'sweep'*, *'slide'*, *'clap'*, *'draw zig-zag'*, and *'draw triangle'*) introduced by [55] are regarded as the default unseen gestures. Each gesture is performed at least 15 times to ensure that we have plenty number of signal samples for testing. In default settings, the base dataset involves 20 gesture classes[1]. In the unseen dataset, the support set and query set of the same gesture class are collected from the same domain (e.g. same orientation and location), implying that the location and orientation information of query sets is pre-known. To validate the *requirement 3* in Section 2.2, we perform cross-domain experiments in Section 6.7 by varying the location and orientation of the user in the sensing area and also by varying the environments through changing the room furniture layout. We pre-process each gesture sample as stated in Section 3 and further downsample the input Doppler spectrogram by taking the mean value of every 40 time patches without overlapping.

Metric: We opt to use *accuracy* [55] to quantify the performance of *OneFi*. It represents the probability that an unseen signal sample can be correctly recognized, which can be calculated by:

$$Accuracy = \frac{N_{cor}}{N_{all}}, \tag{19}$$

where N_{cor} is the number of correctly recognized signal samples and N_{all} is the number of all testing signal samples. The higher the accuracy is, the better *OneFi* performs.

6.2 Overall Accuracy

To show the superiority of *OneFi*, we compare the accuracy of *OneFi* with that of the state-of-the-art few-shot learning works, MetaSense [13] and RF-Net [7]. MetaSense [13] employs MAML [8] as its meta-learning framework. Meanwhile, RF-Net [7] adopts a metric-based meta-learning framework, and it also involves an innovative dual-path base network specifically tailored for RF signals (including WiFi signals).

We calculate the accuracy to recognize six unseen gestures in the one-, three-, five-, and seven-shot settings and show the results in Figure 9. We can observe that in the one-, three, five-, and seven-shot settings, the recognition accuracy of MetaSense is 50.6%, 62.5%, 70.8%, and 73.0% respectively, and the recognition accuracy of RF-Net is 58.6%, 62.1%, 70.0%, and 78.0% respectively. Meanwhile, in these settings, the accuracy of *OneFi* is 84.2%, 94.2%, 95.8%, and 98.8%, respectively. Apparently, no matter in the one-, three-, five-, or seven-shot setting, our system outperforms both MetaSense and RF-Net. The reason is that apart from *OneFi*'s specially-designed one-shot learning framework and WiFi Transformer backbone, the proposed virtual gesture generation technique improves the overall performance, especially when the base dataset is small. Besides, *OneFi* uses a higher packet rate than RF-Net and then performs downsampling on CSI data, which smooths out the noise brought by the hardware. Note that the one-shot performance of *OneFi* is even better than the seven-shot performance of MetaSense and RF-Net, demonstrating that our system has significant scalability for unseen gestures. Such outstanding scalability would greatly promote the performance of WiFi-based HGR in real-world deployment.

6.3 Effect of Virtual Gestures

In *OneFi*, we generate virtual gestures to enrich the base dataset while mitigating the data collection overhead. The introduction of virtual gestures is indeed a kind of data augmentation. In this part, we demonstrate the effectiveness of virtual gesture generation by comparing with two baselines: 1) without using data augmentation; 2) a naive augmentation: adding Gaussian noise on signals to augment data [18]. In the experiment, we calculate the accuracy while varying the number of gesture classes (from two to 20) in the base dataset for training. Theoretically, a smaller number of base classes means a harder problem, leading to lower accuracy. The experiment results for six unseen gesture recognition are shown in Figure 10. We can find that the naive augmentation does not improve accuracy remarkably compared with no augmentation. Meanwhile, the curve of virtual gesture generation is on top of the curves of the two baselines. When there are nine base classes, the accuracy of our approach is higher than 85%. Hence, while the effect of adding Gaussian noise is limited, virtual gesture generation is effective in improving recognition accuracy without collecting a huge base dataset.

6.4 Effect of Proposed Backbone

The backbone plays an important role in a learning framework. In *OneFi*, we design WiFi Transformer as our backbone of the feature extractor. To show the superiority of the WiFi Transformer, we compare its performance with two baselines: LSTM (long short-term memory) and LSTM+CNN (convolutional neural network). We choose these baselines because CNN and LSTM are the most popular and explainable backbones in existing WiFi sensing researches [19, 20, 29, 55]. Specifically, we implement a two-layer LSTM with a hidden size of 128 as our LSTM model. And we add an additional one-dimensional convolutional layer with a kernel size of 3 upon that as our CNN+LSTM model. The comparison results are displayed in Figure 11. CNN+LSTM gets a low accuracy because the coupling of these two structures makes the model hard to converge. Meanwhile, we can easily observe that WiFi Transformer in our system outperforms LSTM and CNN+LSTM in all settings. Thus, the designed backbone, WiFi Transformer, is reasonable and effective.

6.5 Impact of Number of Unseen Gestures

For scalability in real-world deployment, it is important for the HGR system to perform still well when the number of gesture classes becomes larger. While existing works usually consider only 6-8 gesture classes [39, 40, 52], we want to explore the impact of the increasing number of unseen gesture classes on our system.

We vary the number of unseen gesture classes N_u from six to 20. The results are shown in Figure 12. In the one-shot setting, the accuracy is higher than 80% when $N_u \leq 8$. However, one-shot accuracy decreases with the increasing N_u. When $N_u = 20$, the accuracy is only 55%. This is because the recognition difficulty increases when

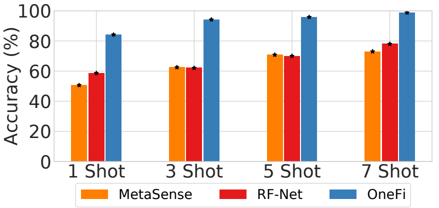


Figure 9: Overall accuracy of *OneFi*, comparing with *MetaSense* and *RF-Net*.

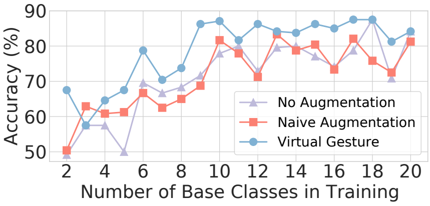


Figure 10: Effect of virtual gestures.

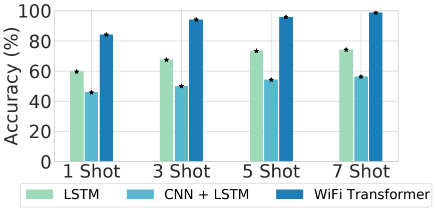


Figure 11: Comparison of different backbone.

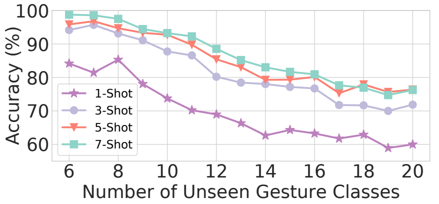


Figure 12: Figure depicts the recognition accuracy when the number of unseen gesture classes varies.

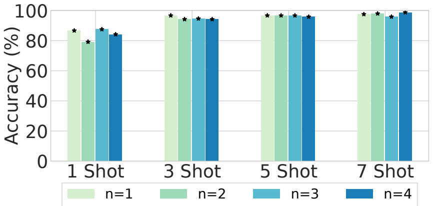


Figure 13: Recognition accuracy with different numbers of receivers.

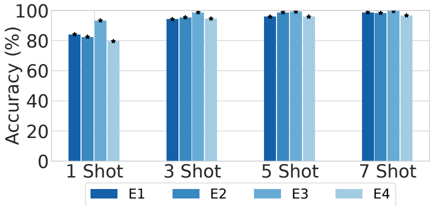


Figure 14: Recognition accuracy on unseen gestures in four different environments.

N_u becomes larger. Few-shot recognition on WiFi gesture data is an inherently difficult task. Even the same gesture performed by the same person would be quite different in terms of movement range and speed. As N_u increases, one solution to counter the accuracy decrease is to add more shots. We can see that with three/five/seven shots from each class, we can achieve an accuracy higher than 70% even when N_u reaches 20, while the accuracy of 'random guess' is only 5%. Therefore, *OneFi* can recognize a large number of unseen gestures with high accuracy.

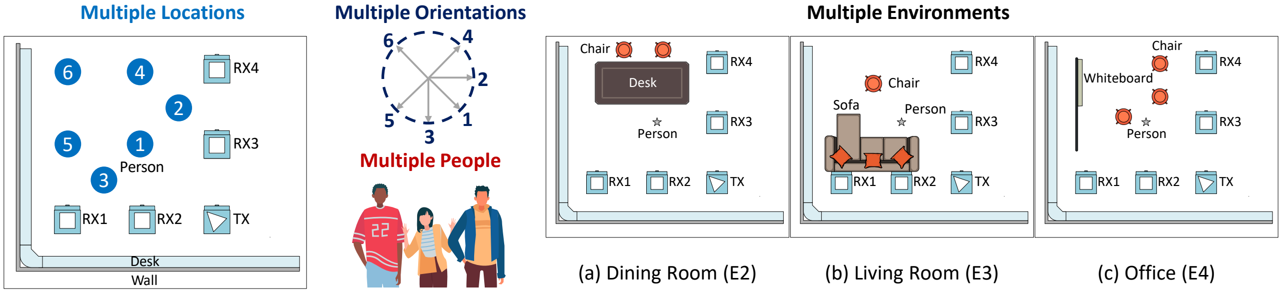
6.6 Impact of Number of Receivers

In our default setting, four receivers are utilized to collect the samples of unseen gestures. Considering the real-world deployment, the impact of the number of receivers is worth exploring because fewer receivers mean easier implementation in a real environment. Specifically, we vary the number of the receivers from one to four and show the accuracy in Figure 13. We can observe that the recognition accuracy does not decrease remarkably with the decrease of the number of receivers, no matter in the one-, three, five, or seven-shot settings. Thus, using one receiver can also achieve comparable accuracy to that using four receivers. This is because the accuracy with one receiver is already high, which almost saturates the performance limit of our framework. In this circumstance, the room for improvement is marginal. Therefore, the performance improvement would be trivial if we increase the number of receivers. It is worth mentioning that multiple receivers are still necessary to collect training data because we require the data of multiple receivers to generate virtual gestures. Once the learning model is trained, only one receiver is required for unseen sample recognition. Thus, *OneFi* is user-friendly and can be easily deployed in real-world environments.

6.7 Cross-Domain Performance

To validate *requirement 3* in Section 2.2, we perform cross-domain experiments. In practice, WiFi signals are sensitive to the variation of the domain, including environment, person, orientation, and location. To suppress the impacts of domain variations, we take three countermeasures: 1) Extracting environment-resistant Doppler spectrogram as the input feature of the learning model, reducing the impacts of the environment. 2) Generating massive virtual gestures to reduce the gap between different orientation domains. 3) Leveraging our few-shot learning framework to fine-tune the deep model so that the recognition model can learn the transferable knowledge from the original location/person domain to the target location/person domain. To validate the effectiveness of our countermeasures, we conduct the following experiments, as shown in Figure 15. We train the feature extractor with the base dataset. Then, we test the unseen gesture recognition accuracy when the location, orientation, user, and environment are different from the base dataset. Note that it is possible to achieve higher accuracy in the unseen domain than in the seen domain because the one-shot recognition accuracy depends on the intra-class similarity due to its inherent similarity comparison nature. For a given gesture class, its support set and query set may have higher similarity in the unseen domain than those in seen domain. In this case, the unseen domain would have even higher accuracy than seen domain.

Cross-environment evaluation. We test the accuracy when unseen gesture classes and base classes are performed in different environments. As shown in Figure 15, we do this experiment in three different environments. The base dataset is collected in 'E1'. The accuracy in the one-, three, five-, and the seven-shot setting is shown in Figure 14. When tested in other environments, we can observed that the accuracy on unseen gestures is higher than 78% in all environments. In the three-, five, and seven-shot settings, most accuracy is higher than 90%. These results indicate that our system



(a) Dining Room (E2) (b) Living Room (E3) (c) Office (E4)

Figure 15: Figure depicts our cross-domain experiment design. We vary multiple parameters to comprehensively investigate the cross-domain behavior of *OneFi*.

has outstanding cross-environment capability, which would greatly facilitate real-world deployment.

Cross-person evaluation. To evaluate the cross-person performance of *OneFi*, we conduct a cross-person experiment with ten people. Specifically, the base dataset is collected from 'P1'. Then, we test *OneFi*'s performance on unseen gestures performed by ten people (from 'P1' to 'P10'). The accuracy in the one- to seven-shot settings is shown in Figure 16. We can find that the recognition accuracy becomes higher with the increase of shot number. In the two-shot setting, the accuracy on most people is higher than 75%. In the three-shot setting, the accuracy on all people excluding 'P4', 'P8', 'P9' is larger than 80%. When the number of shots increases to six, the accuracy on nine people is higher than 84%. Meanwhile, it is worth noting that the mean accuracy on ten people in the one-, three-, five-, and seven-shot settings is 73.3%, 85.8%, 89.8%, and 91.3%, respectively. Apparently, *OneFi* performs well on cross-person unseen gesture recognition.

Cross-orientation evaluation. This experiment tests the accuracy when unseen gesture classes and base classes are performed towards different orientations. As shown in Figure 15, we collect samples of six orientations. The base dataset is collected facing 'O1'. The angle between 'O1' and other five orientations are $\frac{\pi}{4}$, $-\frac{\pi}{4}$, $\frac{\pi}{2}$, $-\frac{\pi}{2}$, and π, labeled as 'O2', 'O3', 'O4', 'O5', and 'O6', respectively. The recognition accuracy is shown in Figure 17. We can find that most of the accuracy is higher than 80% in the one-shot setting. In the three-, five, and seven-shot settings, almost all the accuracy is larger than 85%, and some of the accuracy is even higher than 95%. Meanwhile, the accuracy at 'O6' is relatively low. This is because the reflected signals are not strong enough when the person is performing gestures with his back towards the transmitter. Nevertheless, the accuracy at 'O6' is around 80% in the three-, five-, and seven-shot settings. Hence, our system shows excellent cross-orientation performance.

Cross-location evaluation. We evaluate *OneFi* when unseen gesture classes and base classes are performed at different locations. The base dataset is collected at 'L1', and we collect samples of unseen gestures at six different locations ('L1' to 'L6'), respectively. The results shown in Figure 18 indicate that the recognition accuracy is larger than 70% at 'L2', 'L3', 'L4' and 'L5' in the one-shot setting, which demonstrates the decent cross-location ability of *OneFi*. In the three-, five, and seven-shot settings, the accuracy at all locations excluding 'L6' is higher than 80%. The accuracy at 'L6' is relatively low because 'L6' is far from 'L1', leading to a weak received signal. However, the accuracy at 'L6' reaches 70% when we fine-tune the classifier with more than three shots. Thus, the cross-location performance of *OneFi* is also acceptable.

6.8 Training Overhead

One of the goals in our paper is to reduce training overhead. As aforementioned, our system requires no re-training for unseen gestures. In this experiment, we investigate how much time is required for our system to converge to its best performance (with respect to validation) on the base dataset with a single NVIDIA RTX2080 Ti GPU.

Figure 19 (a) plots the accuracy changes for MetaSense, RF-Net, and *OneFi* as training proceeds. Training *OneFi* for eight seconds gives an accuracy higher than training MetaSense and RF-Net for 2000 seconds, showing *OneFi* entails significantly less training overhead. To investigate whether WiFi Transformer would further relieve training overhead, we do this experiment again, varying the backbone of *OneFi*. Figure 19 (b) shows WiFi Transformer converges within two minutes, minimizing the training overhead.

The time cost of the recognition process is directly related to the real-time performance of *OneFi*. Thus, we measure the latency of our system. It takes our model 7.49 milliseconds to recognize one gesture sample, which demonstrates that our system is not only high in accuracy but also efficient and lightweight in terms of resource consumption.

7 DISCUSSION

Sensing Distance. Even though Section 6.7 shows that *OneFi* can operate in cross-domain settings, the result in Figure 18 shows a drop in accuracy when the performer is far away from the WiFi APs. This is because *OneFi* only leverages non-line-of-sight signals, which are reflected off the human body and hence carrying the body movement information. This signal not only has a longer propagation distance compared with line-of-sight signals but also suffers from attenuation after being reflected from the human body. To get high accuracy in long-distance scenarios, extra efforts are required, e.g., increasing signal strength or the sensitivity of receivers. Another way to improve the accuracy in long-distance scenarios is to enhance our signal propagation model by taking path loss into account.

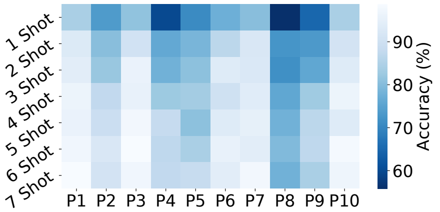


Figure 16: Recognition accuracy on unseen gestures performed by different people.

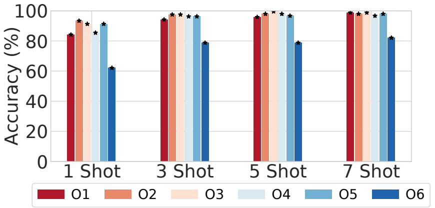


Figure 17: Recognition accuracy on unseen gestures of different orientations.

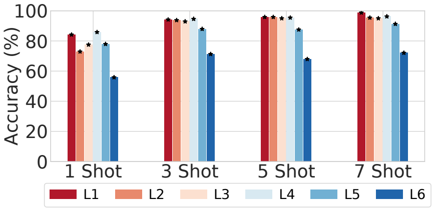


Figure 18: Recognition accuracy on unseen gestures at different different locations.

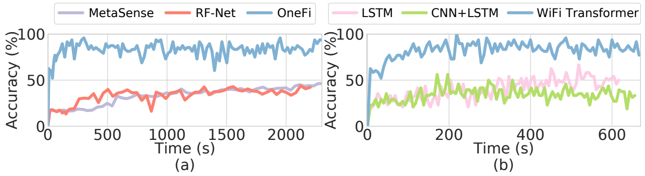


Figure 19: Accuracy changes over training time. (a) shows results of different few-shot learning frameworks. (b) shows results of *OneFi* with different backbones.

Potential extensions of virtual gesture generation. *OneFi* leverages rotation to generate virtual gestures because different orientations of the same gesture can be interpreted as new gestures. To further increase the difference between virtual gestures and base gestures, we can extend our virtual gesture generation method. One potential way of extending *OneFi* is to synthesize two gestures. For example, we may create a *'clap'* gesture by combining a *'sweep left'* gesture and a *'sweep right'* gesture. Another potential extension is to substitute the rotation, a two-dimensional operation, with a three-dimensional transformation. This will further enrich the diversity of the base dataset.

8 RELATED WORK

This work is mainly related to two techniques, i.e., WiFi-based gesture recognition technique and few-shot learning techniques.

WiFi-based gesture recognition: The development of WiFi-based gesture recognition techniques [5, 7, 10, 19, 27, 28, 39, 40, 46, 48, 49, 51, 53–55] can be divided into two phases. In the first phase, researchers aim to model WiFi signal propagation and achieve accurate recognition. For example, Wang *et al.* [48] first build WiFi signal profiles for activities. They accurately identify in-place activity and walking by comparing signal profiles. Zhang *et al.* [53] analyze the sensing feasibility of activity recognition by WiFi signals theoretically and propose a Fresnel zone model to achieve decimeter-scale activity recognition. In the second phase, researchers attempt to improve systems' adaption ability to domain variations (including environment, person, location, and orientation). For instance, Jiang *et al.* [19] leverage adversarial networks to extract environment/subject-independent features of human activities. Zheng *et al.* [55] define a gesture-specific feature, body-velocity profile, to achieve cross-environment, -person, -orientation, and -position gesture recognition. Aditya *et al.* [40] propose WiAG, a cross-position and cross-orientation gesture recognition system. However, these systems still face data collection overhead and training overhead to recognize unseen gestures. In this paper, we solve this problem with virtual gesture generation and a few-shot learning framework.

Few-shot learning techniques: Few-shot learning [3, 7, 11, 13, 21, 23, 26, 30, 42, 44, 47] is proposed to reduce the manpower invested for training data collection. Due to its superiority on unseen class training, it has been applied to many learning-based fields, such as visual data classification, acoustic signal recognition, and mobile computing. For example, Koch *et al.* [21] design a unique structure, called siamese network, to rank similarity, making one-shot image recognition possible. Wang *et al.* [47] achieve one-shot gesture recognition for different users by adopting a dynamic speed warping algorithm. Gong *et al.* [13] employ a few-shot learning framework to enable a deep mobile sensing system to adapt to new users and new devices rapidly. In [7], Ding *et al.* leverage the meta-learning technique to achieve one-shot cross-environment activity recognition with RF signals. To our best knowledge, we are the first to achieve one-shot unseen gesture recognition using COTS WiFi infrastructures.

9 CONCLUSION

In this paper, we propose *OneFi*, a one-shot HGR system to recognize *unseen gestures* using COTS WiFi. Specifically, we overcome the shortcomings of the traditional learning-only approaches by generating virtual gestures via signal modeling to considerably enrich the base dataset and mitigate extra effort in data collection. Besides, we propose a lightweight few-shot learning framework using transductive fine-tuning, along with a novel backbone, WiFi Transformer, to reduce training overhead to a great extent. Extensive experimental results show that *OneFi* achieves a high recognition accuracy in various settings. Combining the power of signal modeling and deep learning to complement each other, *OneFi* is envisioned as a promising step towards practical wireless human-computer interface.

ACKNOWLEDGMENTS

We sincerely thank our anonymous reviewers and shepherd for their valuable feedback. This work is supported in part by National Natural Science Foundation of China under grant 62032021, 61872285, 61772236, and 61972348, Research Institute of Cyberspace Governance in Zhejiang University, Leading Innovative and Entrepreneur Team Introduction Program of Zhejiang (Grant No. 2018R01005), Zhejiang Key R&D Plan (Grant No. 2019C03133), National Key R&D Program of China (2020AAA0107700) and Alibaba-Zhejiang University Joint Institute of Frontier Technologies.

REFERENCES

[1] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. *CoRR* abs/1607.06450 (2016). http://arxiv.org/abs/1607.06450

[2] Andreas Bulling, Ulf Blanke, and Bernt Schiele. 2014. A tutorial on human activity recognition using body-worn inertial sensors. *Comput. Surveys* 46, 3 (2014), 33:1–33:33.

[3] Kaidi Cao, Jingwei Ji, Zhangjie Cao, Chien-Yi Chang, and Juan Carlos Niebles. 2020. Few-Shot Video Classification via Temporal Alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR.*

[4] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. 2019. A Closer Look at Few-shot Classification. In *Proceedings of the International Conference on Learning Representations, ICLR.*

[5] Zhenghua Chen, Le Zhang, Chaoyang Jiang, Zhiguang Cao, and Wei Cui. 2019. WiFi CSI Based Passive Human Activity Recognition Using Attention Based BLSTM. *IEEE Transactions on Mobile Computing, TMC* 18, 11 (2019), 2714–2724.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* (2018).

[7] Shuya Ding, Zhe Chen, Tianyue Zheng, and Jun Luo. 2020. RF-net: a unified meta-learning framework for RF-enabled one-shot human activity recognition. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems, SenSys.*

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the International Conference on Machine Learning, ICML.*

[9] James L Flanagan. 2013. *Speech analysis synthesis and perception.* Vol. 3. Springer Science & Business Media.

[10] Qinhua Gao, Jie Wang, Xiaorui Ma, Xueyan Feng, and Hongyu Wang. 2017. CSI-Based Device-Free Wireless Localization and Activity Recognition Using Radio Image Features. *IEEE Transactions on Vehicular Technology* 66, 11 (2017), 10346–10356.

[11] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. 2019. Boosting Few-Shot Visual Learning With Self-Supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV.*

[12] Georgia Gkioxari, Ross B. Girshick, Piotr Dollár, and Kaiming He. 2018. Detecting and Recognizing Human-Object Interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR.*

[13] Taesik Gong, Yeonsu Kim, Jinwoo Shin, and Sung-Ju Lee. 2019. MetaSense: few-shot adaptation to untrained conditions in deep mobile sensing. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems, SenSys.*

[14] Yu Guan and Thomas Plötz. 2017. Ensembles of Deep LSTM Learners for Activity Recognition using Wearables. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, IMWUT* 1, 2 (2017), 11:1–11:28.

[15] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. 2011. Tool release: gathering 802.11n traces with channel state information. *Computer Communication Review* 41, 1 (2011), 53.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR.*

[17] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. 2020. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439* (2020).

[18] Md Tamzeed Islam and Shahriar Nirjon. 2020. Wi-Fringe: Leveraging Text Semantics in WiFi CSI-Based Device-Free Named Gesture Recognition. In *Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems, DCOSS.*

[19] Wenjun Jiang, Chenglin Miao, Fenglong Ma, Shuochao Yao, Yaqing Wang, Ye Yuan, Hongfei Xue, Chen Song, Xin Ma, Dimitrios Koutsonikolas, Wenyao Xu, and Lu Su. 2018. Towards Environment Independent Device Free Human Activity Recognition. In *Proceedings of the ACM International Conference on Mobile Computing and Networking, MobiCom.*

[20] Wenjun Jiang, Hongfei Xue, Chenglin Miao, Shiyang Wang, Sen Lin, Chong Tian, Srinivasan Murali, Haochen Hu, Zhi Sun, and Lu Su. 2020. Towards 3D human pose construction using wifi. In *Proceedings of the ACM International Conference on Mobile Computing and Networking, MobiCom.*

[21] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *Proceedings of the International Conference on Machine Learning, ICML.*

[22] Yoonho Lee and Seungjin Choi. 2018. Gradient-Based Meta-Learning with Learned Layerwise Metric and Subspace. In *Proceedings of the International Conference on Machine Learning, ICML.*

[23] Kai Li, Yulun Zhang, Kunpeng Li, and Yun Fu. 2020. Adversarial Feature Hallucination Networks for Few-Shot Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR.*

[24] Tianxing Li, Qiang Liu, and Xia Zhou. 2016. Practical Human Sensing in the Light. In *Proceedings of the ACM International Conference on Mobile Systems, Applications, and Services, MobiSys.*

[25] Xiang Li, Daqing Zhang, Qin Lv, Jie Xiong, Shengjie Li, Yue Zhang, and Hong Mei. 2017. IndoTrack: Device-Free Indoor Human Tracking with Commodity Wi-Fi. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, IMWUT* 1, 3 (2017), 72:1–72:22.

[26] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. 2019. Few-Shot Unsupervised Image-to-Image Translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV.*

[27] Yong Lu, Shaohe Lv, and Xiaodong Wang. 2019. Towards Location Independent Gesture Recognition with Commodity WiFi Devices. *MDPI AG Electronics* 8, 10 (Sep 2019), 1069.

[28] Junyi Ma, Hao Wang, Daqing Zhang, Yasha Wang, and Yuxiang Wang. 2016. A Survey on Wi-Fi Based Contactless Activity Recognition. In *IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress.*

[29] Yongsen Ma, Gang Zhou, Shuangquan Wang, Hongyang Zhao, and Woosub Jung. 2018. SignFi: Sign Language Recognition Using WiFi. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, IMWUT* (2018).

[30] Seonwook Park, Shalini De Mello, Pavlo Molchanov, Umar Iqbal, Otmar Hilliges, and Jan Kautz. 2019. Few-Shot Adaptive Gaze Estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV.*

[31] Florian A. Potra and Stephen J. Wright. 2000. Interior-point methods. *J. Comput. Appl. Math.* 124, 1 (2000), 281–302.

[32] John G Proakis. 1995. *Digital communications.* McGraw-Hill.

[33] Kun Qian, Chenshu Wu, Zheng Yang, Yunhao Liu, and Kyle Jamieson. 2017. Widar: Decimeter-Level Passive Tracking via Velocity Monitoring with Commodity Wi-Fi. In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc.*

[34] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. 2000. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision* 40, 2 (2000), 99–121.

[35] Sheng Shen, He Wang, and Romit Roy Choudhury. 2016. I am a Smartwatch and I can Track my User's Arm. In *Proceedings of the ACM International Conference on Mobile Systems, Applications, and Services, MobiSys.*

[36] Mehrdad Soumekh. 1999. *Synthetic aperture radar signal processing.* Vol. 7. New York: Wiley.

[37] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B. Tenenbaum, and Phillip Isola. 2020. Rethinking Few-Shot Image Classification: A Good Embedding is All You Need?. In *Proceedings of the European Conference on Computer Vision, ECCV.*

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Proceedings of the Conference on Neural Information Processing Systems, NeurIPS.*

[39] Raghav H. Venkatnarayan, Griffin Page, and Muhammad Shahzad. 2018. Multi-User Gesture Recognition Using WiFi. In *Proceedings of the ACM International Conference on Mobile Systems, Applications, and Services, MobiSys.*

[40] Aditya Virmani and Muhammad Shahzad. 2017. Position and Orientation Agnostic Gesture Recognition Using WiFi. In *Proceedings of the ACM International Conference on Mobile Systems, Applications, and Services, MobiSys.*

[41] Juan Pablo Wachs, Mathias Kölsch, Helman Stern, and Yael Edan. 2011. Vision-based hand-gesture applications. *Commun. ACM* 54, 2 (2011), 60–71.

[42] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. 2019. PANet: Few-Shot Image Semantic Segmentation With Prototype Alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV.*

[43] Minsi Wang, Bingbing Ni, and Xiaokang Yang. 2017. Recurrent Modeling of Interaction Context for Collective Activity Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR.*

[44] Tao Wang, Jianhua Tao, Ruibo Fu, Jiangyan Yi, Zhengqi Wen, and Rongxiu Zhong. 2020. Spoken Content and Voice Factorization for Few-Shot Speaker Adaptation. In *Proceedings of the ISCA Conference of the International Speech Communication Association, Interspeech.*

[45] Wei Wang, Alex X. Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. 2015. Understanding and Modeling of WiFi Signal Based Human Activity Recognition. In *Proceedings of the ACM International Conference on Mobile Computing and Networking, MobiCom.*

[46] Wei Wang, Alex X. Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. 2017. Device-Free Human Activity Recognition Using Commercial WiFi Devices. *IEEE Journal on Selected Areas in Communications* 35, 5 (2017), 1118–1131.

[47] Xun Wang, Ke Sun, Ting Zhao, Wei Wang, and Qing Gu. 2020. Dynamic Speed Warping: Similarity-Based One-shot Learning for Device-free Gesture Signals. In *Proceedings of the IEEE Conference on Computer Communications, INFOCOM.*

[48] Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu. 2014. E-eyes: device-free location-oriented activity identification using fine-grained WiFi signatures. In *Proceedings of the ACM International Conference on Mobile Computing and Networking, MobiCom.*

[49] Yuxi Wang, Kaishun Wu, and Lionel M. Ni. 2017. WiFall: Device-Free Fall Detection by Wireless Networks. *IEEE Transactions on Mobile Computing, TMC* 16, 2 (2017), 581–594.

[50] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. 2020. Generalizing from a Few Examples: A Survey on Few-shot Learning. *Comput. Surveys* 53, 3 (2020), 63:1–63:34.

[51] Siamak Yousefi, Hirokazu Narui, Sankalp Dayal, Stefano Ermon, and Shahrokh Valaee. 2017. A Survey on Behavior Recognition Using WiFi Channel State Information. *IEEE Communications Magazine* 55, 10 (2017), 98–104.

[52] Yinggang Yu, Dong Wang, Run Zhao, and Qian Zhang. 2019. RFID based real-time recognition of ongoing gesture with adversarial learning. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems, SenSys*.

[53] Daqing Zhang, Hao Wang, and Dan Wu. 2017. Toward Centimeter-Scale Human Activity Sensing with Wi-Fi Signals. *Computer* 50, 1 (2017), 48–57.

[54] Jie Zhang, Zhanyong Tang, Meng Li, Dingyi Fang, Petteri Nurmi, and Zheng Wang. 2018. CrossSense: Towards Cross-Site and Large-Scale WiFi Sensing. In *Proceedings of the ACM International Conference on Mobile Computing and Networking, MobiCom*.

[55] Yue Zheng, Yi Zhang, Kun Qian, Guidong Zhang, Yunhao Liu, Chenshu Wu, and Zheng Yang. 2019. Zero-Effort Cross-Domain Gesture Recognition with Wi-Fi. In *Proceedings of the ACM International Conference on Mobile Systems, Applications, and Services, MobiSys*.