WiFi-Based Cross-Domain Gesture Recognition via Modified Prototypical Networks

Xie Zhang, Chengpei Tang, Kang Yin, and Qingqian Ni

***Abstract*—Numerous deep learning studies have achieved remarkable advances in WiFi-based human gesture recognition (HGR) using channel state information (CSI). However, since the CSI patterns of the same gesture change across domains (i.e., users, environments, locations, and orientations), recognition accuracy might degrade significantly when applying the trained model to new domains. To overcome this problem, we propose a WiFi-based cross-domain gesture recognition system (WiGr) which has a domain-transferable mapping to construct an embedding space where the representations of samples from the same class are clustered, and those from different classes are separated. The key insight of WiGr is using the similarity between the query sample representation and the class prototypes in the embedding space to perform the gesture classification, which can avoid the influence of the cross-domain CSI patterns change. Meanwhile, we present a dual-path prototypical network (Dual-Path PN) which consists of a deep feature extractor and a dual-path (i.e., Path-A and Path-B substructures) recognizer. The trained feature extractor can extract the gesture-related domain-independent features from CSI, namely, the domain-transferable mapping. In addition, WiGr implements the cross-domain HGR based on only a pair of WiFi devices without retraining in the new domain. We conduct comprehensive experiments on three data sets, one is built by ourselves and the others are public data sets. The evaluation suggests that WiGr achieves 86.8%–92.7% in-domain recognition accuracy and 83.5%–93% cross-domain accuracy under the four-shot condition.**

***Index Terms*—Channel state information (CSI), cross-domain recognition, gesture recognition, prototypical networks (PNs).**

I. Introduction

HUMAN gesture recognition (HGR) plays an important role in human–computer interaction [1], [2], and can support many emerging Internet-of-Things (IoT) applications, such as smart home [3], [4], user identification [5], [6], and health care [7]. Generally, the methods that enable the HGR rely on cameras [8], [9], wearable devices [10], [11], radars [12], [13], and smartphones [14]. However, these methods may incur extra equipment costs or raise privacy issues. In recent years, WiFi-based HGR methods [15], [16] have received immense attention due to private security and ease of deployment. Particularly, numerous studies [15], [17] based on deep learning have made significant advances in this field.

The rationale behind WiFi-based HGR is that human gestures can bring about signal fluctuations which can be extracted from the physical layer feature of WiFi, namely, channel state information (CSI) [18]. However, WiFi signals are absorbed, diffracted, reflected, or scattered by other objects during propagation, leading to the high coupling between CSI and environmental factors besides human gestures. Fig. 1(a)–(d) shows that the CSI amplitude patterns of the same gesture across domains (i.e., environment, user, location, and orientation) are quite different, called the cross-domain pattern change. Furthermore, such differences are even more obvious than that of different gestures in the same domain shown in Fig. 1(e). Fig. 1(f) also illustrates that high accuracy can be achieved if the ARIL model [19] is trained and tested at the same location. However, the accuracy drops to below 20% when the trained model is applied to the testing data from the new location. In a nutshell, the recognition performance of the general WiFi-based HGR model might degrade significantly when applying the trained model to a new domain (i.e., new users, various environments, and users in different locations and orientations) [20], [21], which is called the cross-domain problem.

To address this problem, many studies have been proposed for WiFi-based cross-domain HGR. In [22] (Widar3.0), the authors introduced a one-fits-all deep learning model for cross-domain HGR based on a domain-independent handcraft feature. Zou *et al.* [16] proposed an adversarial unsupervised domain adaptation scheme JADA to construct a domain-invariant feature space. Similarly, EI [23] adopted adversarial learning to train a robust HGR model. WiAG [24] presented a translation function to automatically generate virtual samples for the target domain, and trained recognition models using virtual samples under all possible domain configurations.

Nevertheless, these methods have obvious limitations. Widar3.0 [22] has shown state-of-the-art performance in cross-domain HGR. However, it needs at least three receivers and one transmitter to gain enough CSI measurements for the feature extraction. JADA [16] and EI [23] are based on adversarial learning. These methods require to collect a large number of unlabeled data from each new domain, which is a labor-intensive and time-consuming process. WiAG [24] needs to generate virtual samples and train specified models for all domain configurations, which is not computing-friendly.

Manuscript received March 26, 2021; revised June 18, 2021, July 1, 2021, and August 3, 2021; accepted September 15, 2021. Date of publication September 21, 2021; date of current version May 23, 2022. This work was supported in part by the Guangdong Provincial Applied Science and Technology Research and Development Program under Grant 2016B090918110 and Grant 2014B090901057, and in part by the Natural Science Foundation of Guangdong Province under Grant 2018A030313797. *(Corresponding author: Chengpei Tang.)*

The authors are with the School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou 510006, China (e-mail: zhangx289@mail2.sysu.edu.cn; tchengp@mail.sysu.edu.cn; yink5@mail2.sysu.edu.cn; niqq5@mail2.sysu.edu.cn).

Digital Object Identifier 10.1109/JIOT.2021.3114309

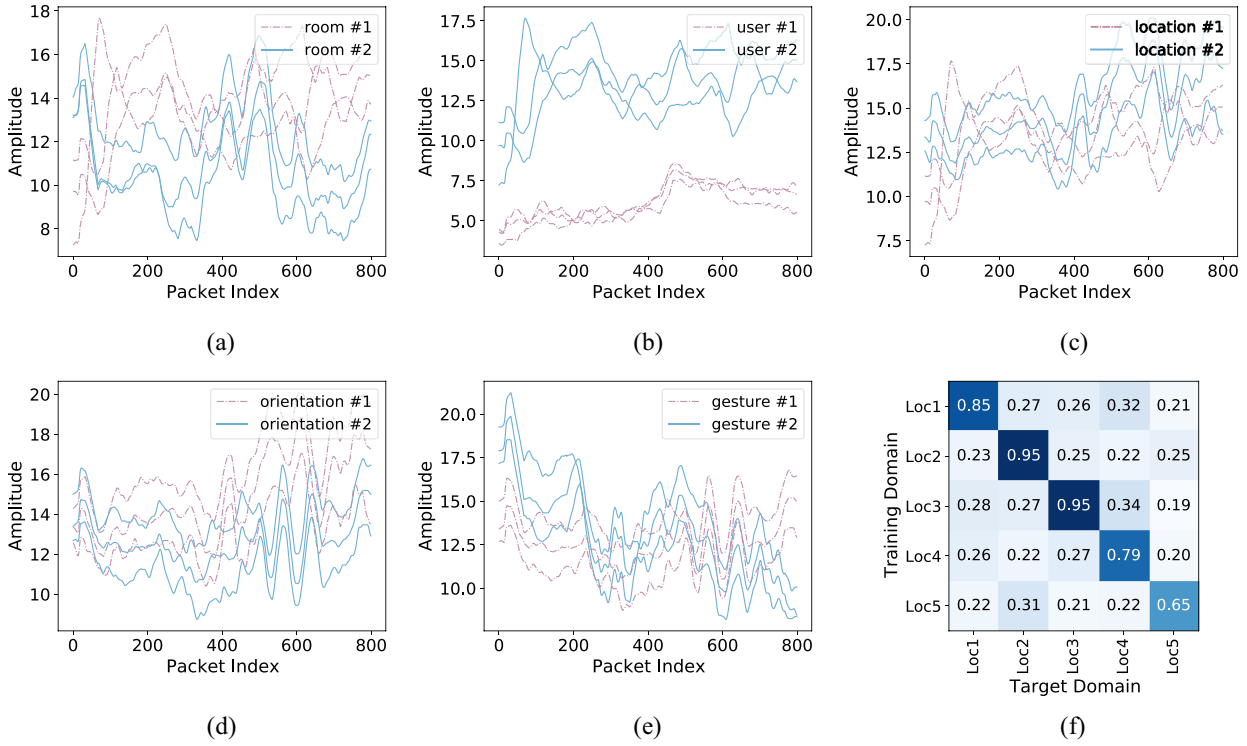

Fig. 1. Examples of cross-domain samples and deep learning model performance. (a), (b), (c), and (d) CSI amplitude patterns for a fixed gesture of cross-environment, cross-user; cross-location, and cross-orientation, respectively. (e) CSI amplitude patterns of different gestures in the same domain. (f) Accuracy of the ARIL method [21] for in-location and cross-location tests.

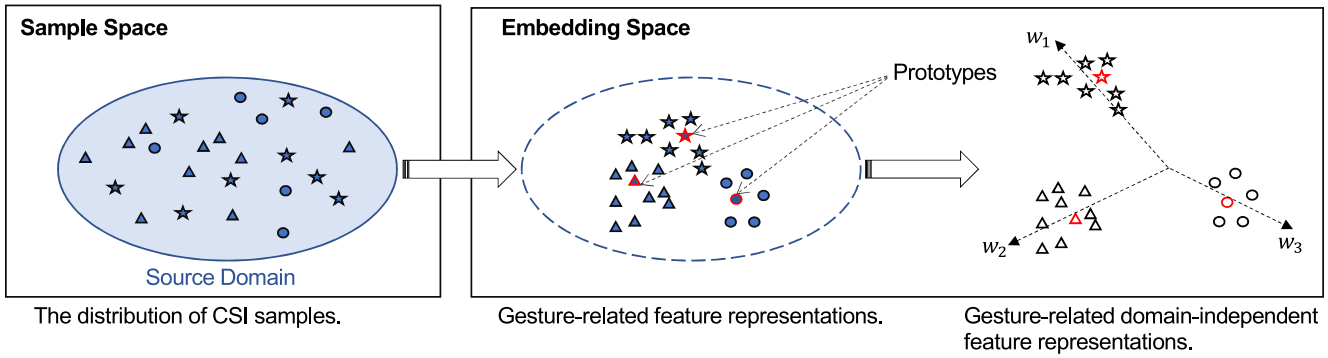

Fig. 2. Intuition of gesture-related domain-independent feature extraction behind Dual-Path PN. Different shapes represent different gestures.

In this article, we propose a WiFi-based cross-domain gesture recognition system (WiGr) to achieve comparable cross-domain recognition accuracy against the abovementioned methods. WiGr uses a small number of labeled samples from a pair of WiFi devices without retraining in the new domain. Concretely, we first learn a domain-transferable mapping to construct an embedding space where the representations of samples from the same class are clustered, and those from different classes are separated. Then, for a new domain, the prototype representations of each class are computed by using a small number of labeled samples. Finally, the classification can be performed by computing the distances between the query sample representation and the prototypes in the embedding space. In this way, by using the similarity rather than the CSI pattern itself, WiGr can avoid the influence of the cross-domain pattern change to address the cross-domain problem. However, there is a challenge in learning the domain-transferable mapping from the sample space to the embedding space.

To address this challenge, the dual-path prototypical network (Dual-Path PN), which consists of a deep feature extractor and a dual-path recognizer (i.e., Path-A and Path-B substructure), is provided in WiGr. In fact, the learned feature extractor is the abovementioned domain-transferable mapping. In the training phase, given a set of labeled CSI samples from the same domain (i.e., source domain), the feature extractor is trained to extract the gesture-related domain-independent features from the CSI samples. As illustrated in Fig. 2, the training phase consists of two stages. At the first stage, the prototype representation for each class is defined as the mean of randomly chosen sample representations in the embedding space from the corresponding class. To learn extracting gesture-related features, the feature extractor is encouraged to cluster the representations of the remaining samples to

their corresponding prototypes by minimizing the losses of Path-A and Path-B substructures. This is because the similarities among CSI samples are related to gesture types. At the second stage, a regularization, namely, orthogonal regularization (OR), is provided to increase the gaps between different clusters in the embedding space. Since the domain feature is identical in all training samples, this leads to the same representations in the embedding space. Increasing the gaps is helpful in eliminating the domain features while maintaining the gesture-related features.

In the testing phase, for a new domain, a set of labeled samples, namely, support set, is needed in which each class has at least one CSI sample to compute the prototype representation. Classification is then performed by simply finding the nearest class prototype to the query sample representation in the embedding space.

We evaluate WiGr on three data sets: 1) ARIL; 2) Widar3.0; and 3) CSIDA. The experimental results show that, based on the available support set, WiGr can identify common human gestures with high accuracy under domain dynamics. In summary, our contributions are listed as follows.

1) Based on the observation that the CSI pattern changes across domains, we propose a novel framework named WiGr to achieve cross-domain HGR by using the similarities between the query sample representation and the class prototypes rather than the CSI pattern itself for gesture recognition. In addition, WiGr is constructed based on only a pair of WiFi devices and can be directly deployed in any new domain by using a small number of labeled samples without retraining.
2) Dual-Path PN, consisting of a deep feature extractor and a dual-path (i.e., Path-A and Path-B substructures) recognizer, is proposed in this work, which can encourage the clustering under the Euclidean distance or cosine similarity (i.e., a flexible cluster, refer to Section IV).
3) To enhance the availability of the deep feature extractor in any new domain (i.e., a domain-transferable mapping), we propose OR to increase the gaps between different clusters in the embedding space.
4) We conduct experiments on three public data sets. The results show that WiGr achieves on average 89%, 93%, 83.5%, and 84% recognition accuracy for cross-environment, cross-user, cross-location, and cross-orientation, respectively. This achieves comparable performance with state-of-the-art works.

II. Related Works

In this section, we first introduce the cross-domain methods in CSI-based sensing. Then, some studies, dealing with cross-domain issues based on few-shot learning, are briefly presented.

A. CSI-Based Cross-Domain Sensing Techniques

For the CSI-based cross-domain sensing approach, existing methods can be classified into three categories: 1) domain-independent feature-based; 2) transfer learning-based; and 3) few-shot learning-based.

1) Domain-Independent Feature-Based: Widar3.0 [22] proposed a domain-independent feature (i.e., body-coordinate velocity profile), derived from doppler frequency shift and developed a one-fits-all deep learning model for cross-domain gesture recognition. However, Widar3.0 needed at least three receivers to collect enough CSI measurements for the extraction of BVP, which limited the use of the system. Compared with the manually designed domain-independent features, Zou *et al.* [16] adopted adversarial learning to construct a domain-independent feature space. Specifically, there were two deep learning networks as encoders to map source and target domain data into the feature space which was restricted to be domain-independent by adversarial learning. Then, a shared classifier was applied to achieve acceptable gesture recognition performance in both domains. Similarly, some studies [23], [25] adopted both labeled source data and unlabeled target data to train robust domain adaptative models. For extracting domain-independent features, some researchers tried to introduce regularization into the loss function of the deep learning model, e.g., Han *et al.* [17] introduced a Maximum Mean Discrepancy regularization into the loss function to alleviate the feature heterogeneity across domains.

Consequently, these methods need a great amount of data in the new domain or more than a pair of WiFi devices (i.e., a transmitter and a receiver), resulting in the degradation of practicability.

2) Transfer Learning-Based: Zhang *et al.* [26] employed an ANN-based roaming model to generate simulated CSI samples for the target environment and retrained the recognition model by using simulated CSI data to enhance the performance in the new environments. Similarly, Virmani and Shahzad [24] proposed a position and orientation agnostic gesture recognition system, WiAG, which can automatically generate virtual samples in all domain configurations by applying a translation function on the source samples. Subsequently, WiAG trained different gesture classifiers for each configuration by using the corresponding virtual samples. Sheng *et al.* [27] addressed the cross-domain issue by using a trained source domain model as the pretrained model and fine-tuned it with a small amount of labeled data in the new scenario.

As such, these methods are not computing friendly for the extra training in each new domain.

3) Few-Shot Learning-Based: Yang *et al.* [28] proposed a one-shot gesture recognition system based on a Siamese framework and transferable pairwise loss which helped to eliminate the structure noise (i.e., individual heterogeneity, environment differences). Inspired by the relation network, Ma *et al.* [15] presented a device-free gesture recognition system, DFGR, which was robust to new users and environments due to the transferrable similarity evaluation ability. In addition, Shi *et al.* [29] proposed MatNet, a neural network augmented with external memory, to improve the environmental robustness via one-shot learning.

B. Few-Shot Learning-Based Domain Adaptation

Few-shot learning methods are dedicated to enabling machine learning models to quickly adapt to related new

tasks with limited training samples, which have received much attention in recent years. Since the task remains the same across domains and can be regarded as related tasks, the cross-domain problem can be addressed by few-shot learning methods. Zou *et al.* [30] proposed a new few-shot domain adaptation scheme F-CADA. F-CADA specifically adopted adversarial learning to construct an embedding space where the source and target data are confused. Moreover, F-CADA enhanced the performance of the target classifier with a few labeled target data via greedy label propagation. In [31], the basic idea was to initialize the classifier with cross-task parameters which were obtained from multiple specific-task parameters. And the initialed classifier needed to be fine-tuned in the new domain with a few labeled data. Zhao *et al.* [32] proposed a domain-adversarial prototypical network (PN) model to solve the domain-adaptation and few-shot learning in a unified framework. The key point of this method was to align the global domain distribution whilst maintaining source/target per-class distinguishability via adversarial learning and prototype learning.

III. Preliminaries

In this section, we first introduce the CSI of WiFi. Then, the original prototype network (original PN) [33] is briefly introduced, and the problem solved in this article is expounded.

A. Channel State Information

CSI, reflecting the channel response, is the physical layer (PHY) feature in the IEEE 802.11n protocol. In particular, CSI contains information regarding the region that the WiFi signal propagates. Due to the use of orthogonal frequency division multiplexing (OFDM) and multiple-input multiple-output (MIMO), CSI is sufficient to discriminate multipath characteristics. Traditionally, the most widely used wireless signal characteristic is the received signal strength (RSS) [34], [35]. RSS is the feature of the media access control layer, which measures the overall attenuation of signals across all subchannels. Compared with RSS, CSI has higher granularity and more information since it reflects the response of each subchannel [18].

In the time domain, the modeling of the channel response relies on channel impulse response (CIR), which is denoted as [36]

$$h(t) = \sum_{i=0}^{l-1} a_i \delta(t - \tau_i) e^{-j2\pi f_i} \tag{1}$$

where a_i, $2\pi f_i$, and τ_i are the attenuation factor, phase shift, and time delay of the signal on the ith path, respectively. f_i represents the frequency of subcarrier. Letter l denotes the number of signal paths, and $\delta(t)$ is the Dirac delta function.

CSI can reflect CIR of all subchannels, and each CSI element represents the Fourier transform of CIR in the specific subchannel

$$H(f_k) = \|H(f_k)\| e^{j\angle H(f_k)} \tag{2}$$

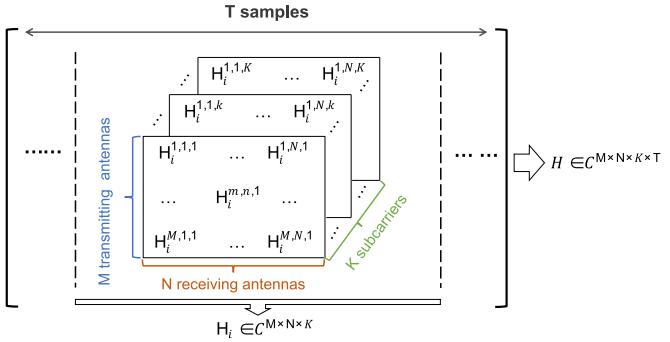

Fig. 3. Data structure of the CSI sample.

where $H(f_k)$ denotes the CSI measurement of a subcarrier with central frequency f_k. $\|H(f_k)\|$ and $\angle H(f_k)$ represent the amplitude and phase, respectively.

As illustrated in Fig. 3, the CSI sample can be regarded as a time series, of which each element is a CSI matrix. In particular, for an MIMO-OFDM channel with M transmitting antennas, N receiving antennas, and K subcarriers, the ith CSI matrix is a 3-D complex matrix $H_i \in C^{M\times N\times K}$ representing amplitude attenuation and phase shift of the ith sampling. In addition, the sampling time T is determined by the sampling rate r of the CSI capture tool and the gesture execution time t (i.e., $T = r \times t$). Hence, the segmented CSI sample used for HGR is a 4D complex tensor $H \in C^{M\times N\times K\times T}$, which characterizes signal variations in different domains (i.e., time, frequency, and spatial).

B. Prototypical Networks

PNs was first proposed in [33], and its basis is that there exists an embedding space in which the embedding representations of each class cluster around a single prototype. In particular, a set with N labeled samples $P = \{(x_i, y_i)\}_{i=1}^{N}$ is given, where $y_i \in \{1, 2, \ldots, C\}$ is the class label of the sample $x_i \in R^D$, belonging to C categories. Support set $S = \{s_1, s_2, \ldots, s_C\}$ is constructed, where $s_c = \{(x_i, y_i)\}_{i=1}^{K}$ is a set of labeled samples of the cth category, where $c \in \{1, 2, \ldots, C\}$. The objective is to learn an embedding function $F_\Theta : R^D \to R^M$ for mapping input samples into a M-dimensional embedding space, where Θ denotes the learnable parameters. The prototype of each category is the mean values of the embedding representations of the corresponding support samples

$$\mathbf{P_c} = \frac{1}{|s_c|} \sum_{(x_i, y_i)\in s_c} F_\Theta(x_i). \tag{3}$$

Given a sample x_i from P, PN will produce a distribution over classes of this sample. The distribution is determined by the distance between the embedding representation of this sample and the prototype. Hence, given a distance function $d : R^M \times R^M \to [0, +\infty)$, the probability of x_i belonging to class c is defined as follows:

$$p\big(y = c \mid x_i\big) = \frac{\exp(-d(F_\Theta\ (x_i), \mathbf{P_c}))}{\sum_{c'=1}^{C} \exp(-d(F_\Theta(x_i), \mathbf{P_{c'}}))}. \tag{4}$$

PN is trained by minimizing the negative log-probability

$$J(\Phi) = -\log p\big(y = k|x_i\big) \tag{5}$$

where k is the true label of the sample x_i, and Φ denotes the learnable parameters of PN.

C. Problem Statement

CSI is used to enable the HGR for it can reflect the channel disturbance caused by gestures. However, the disturbance is highly heterogeneous across users, environments, locations, and orientations. Due to this intrinsic diversity, general CSI-based HGR systems are difficult to generalize to new domains (i.e., new users, various environments, and users with different locations and orientations). This is known as the domain shift problem [37], which significantly degrades the cross-domain performance of existing CSI-based HGR solutions.

In practical applications, it is feasible to collect a small number of labeled samples in the target domain, which can be regarded as a system calibration process. By taking advantage of these labeled samples, a few-shot learning-based method can be used to improve cross-domain recognition performance.

Formally, we have three data sets: 1) a training set P; 2) a support set S; and 3) a testing set T. All of these data sets share the same label space. Specifically, the training set is a collection of labeled data from the source domain, the support set contains few labeled data from the target domain, and a testing set consists of unlabeled data from the target domain. If the number of categories is C, and the support set contains K-labeled samples for each category, then the cross-domain problem is called the C-way K-shot CD problem.

IV. System Design

In this section, we present the WiGr framework of CSI-based HGR. To begin with, we overview the relationships among different parts. We then provide a comprehensive discussion of data processing, feature extractor, Dual-Path PN, and OR in this system. Finally, we introduce the training scheme of WiGr.

A. System Framework

The overall design of WiGr is shown in Fig. 4. At the first stage, the raw CSI samples are collected by two commercial WiFi devices where one is the transmitter and the other is the receiver. As a generalized system, WiGr is compatible with different CSI capture devices (i.e., the 802.11n CSI tool [38], the nexmon CSI Extractor [39], Wi-ESP [40], and Atheros CSI tool [41]). At the next stage, the collected raw CSI samples are preprocessed, including denoise and time length modification. At the final stage, the Dual-Path PN extracts the gesture-related features and predicts the gesture class.

B. Data Processing

1) Denoise: The WiFi signals contain many interferences which are mainly caused by high-frequency noises [27]. We use a finite impulse response filter to obtain the denoised data. In addition, the phase value of CSI is wrapped in the range

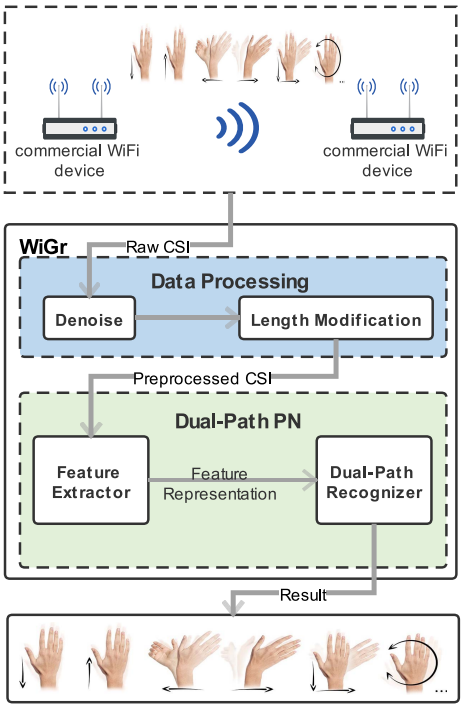

Fig. 4. System architecture of WiGr.

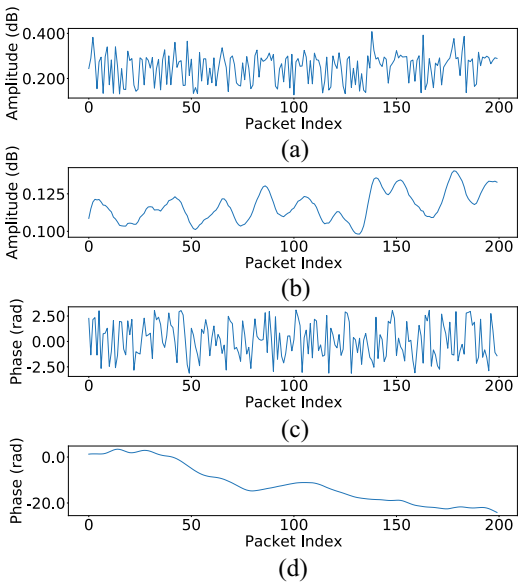

Fig. 5. Example of CSI sample denoise. (a) and (b) Amplitude patterns of CSI samples before and after denoising. (c) Raw phase pattern. (d) Denoised phase pattern.

of $[-\pi, \pi]$, which provides wrong information of the signal variation. Hence, we unwrap the phase values before the filter processing. In Fig. 5, the CSI waveforms of a subcarrier before and after denoising are illustrated.

2) Length Modification: For different gestures, CSI samples might have different time lengths. However, for the parallelization of the Dual-Path PN model, it is crucial to normalize the time length of the samples into a fixed length. To accomplish this, we set the fixed execution time for gestures to be 1.8 s,

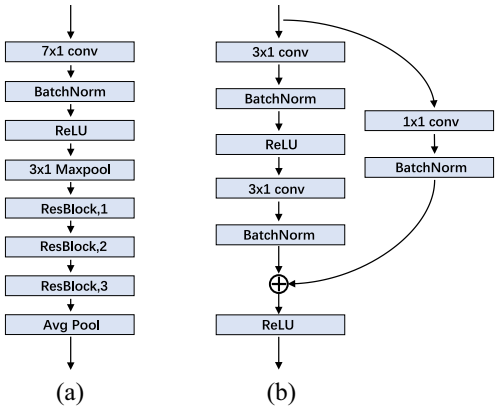

Fig. 6. Structure of the feature extractor. "3 × 1 conv" and "7 × 1 conv" represent the 1-D convolutional operation with 3 × 1 and 7 × 1 kernel sizes, respectively. "BatchNorm" stands for batch normalization [43]. "ReLU" denotes the ReLU. (a) Detailed implementation of the residual block. (b) Structure of the feature extractor. "3 × 1 Maxpool" represents the max pooling operation with 3 × 1 kernel size. "ResBlock, s" stands for the residual block with stride as *s*. "Avg Pool" denotes the average pooling operation.

which is long enough to complete common gestures. For the samples exceeding the fixed time length, the excess part will be truncated directly. For those shorter than the fixed time length, the sample end is padded with zeros.

C. Feature Extractor

The core task of this part is to extract the high-level gesture-related domain-independent features from the preprocessed CSI sample. We use an ResNet-like [42] convolutional architecture as the backbone of the feature extractor. Since the CSI data are time series, 1-D convolution, sweeping along the time axis, is adopted in this model. In fact, limited works are concentrating on what structures of neural networks are suitable for extracting features of wireless signal data. We conducted some basic experiments to select the effective structure.

There are two adjustments deployed for the preprocessed CSI sample before input to the feature extractor. The preprocessed CSI sample $H \in C^{M\times N\times K\times T}$ is a 4D complexed tensor that is not suitable for the input of the feature extractor. As illustrated in Fig. 6(a), we deploy the group convolution in the first convolutional layer of the extractor, and the number of groups is $M \times N$. The input of each group is a 2-D complexed tensor $H_g \in C^{K\times T}$, where $g = 1, 2, \ldots, M \times N$. Next, the CSI sample is decomposed into two parts: 1) amplitude $A_g \in R^{K\times T}$ and 2) phase $P_g \in R^{K\times T}$. Hence, the first convolutional layer adopts K-channel convolution when utilizing amplitude or phase as input. If both amplitude and phase are used as input, we directly concatenate these two tensors as $AP_g \in R^{2K\times T}$, and the first convolutional layer deploys a $2K$-channel convolution. The outputs of the first layer are processed by batch normalization [43] and max pooling. Then, the features are input to three cascade residual blocks [refer to Fig. 6(b)] which have 3 × 1 filters and batch normalization [43]. The activate function is the rectified linear unit (ReLU), and a shortcut is constructed in the residual block. Finally, the last layer of the extractor is an average pooling layer to output the feature representation.

Let F_Θ represents the feature extraction network, where Θ denotes the learnable parameters. Given the input data x, we can obtain the feature representation as follows:

$$z = F_\Theta(x). \tag{6}$$

D. Dual-Path Prototypical Network

The Dual-Path PN, as illustrated in Fig. 7, consists of two parts, a deep feature extractor and a dual-path recognizer. Since the details of the feature extractor have been provided in the previous section, we only focus on the dual-path recognizer in this section. Specifically, the dual-path recognizer contains two substructures: 1) Path-A and 2) Path-B. Path-A consists of a fully connected (FC) layer and a softmax function. Path-B contains a similarity evaluation function and a softmax function.

A sample x_j^T from the testing set T and a support set $S = \{(x_i^S, y_i^S)\}_{i=1}^M$ are given, where $y_i^S \in \{1, 2, \ldots, C\}$ is the class label of sample $x_i^S \in R^D$. We obtain the feature representations of the above samples via the feature extractor as follows:

$$z_j^T = F_\Theta\left(x_j^T\right) \text{ and } Z^S = \left\{z_i^S\right\}_{i=1}^M = \left\{F_\Theta\left(x_i^S\right)\right\}_{i=1}^M \tag{7}$$

where z_j^T and Z^S are the feature representations of x_j^T and samples in S, respectively. Based on the outputs of the feature extractor, a dual-path recognizer is used to predict the gesture classes.

In *Path-A*, an FC layer (without bias) followed by a softmax function is used to obtain the probability distribution over the gesture classes. Based on z_j^T, the predicted probability distribution is calculated as follows:

$$\widehat{y}_{jk}^T = \frac{\exp\left(H_{jk}^T\right)}{\sum_{c=1}^C \exp\left(H_{jc}^T\right)} \text{ and } H_j^T = W z_j^T \tag{8}$$

where $\widehat{y}_{jk}^T$ presents the predicted probability of x_j^T belonging to class k. W denotes the learnable parameters of the FC layer.

Similarly, based on Z^S, the probability distribution over the gesture classes of the samples in the support set can be obtained as follows:

$$\widehat{y}_{ik}^S = \frac{\exp\left(H_{ik}^S\right)}{\sum_{c=1}^C \exp\left(H_{ic}^S\right)} \text{ and } H_i^S = W z_i^S \tag{9}$$

where $\widehat{y}_{ik}^S$ denotes the predicted probability of x_i^S belonging to class k. W denotes the learnable parameters of the FC layer.

For labeled data (i.e., replacing the unlabeled testing set with a labeled set, namely, a query set $Q = \{(x_j^Q, y_j^Q)\}_{j=1}^O$), cross-entropy function is adopted to calculate the loss between predictions and the ground truth as follows:

$$L_f = \left(-\frac{1}{|S|}\sum_{y_i\in S}\sum_{c=1}^{C}(1|y_i = c\rangle)\log\widehat{y}_{ic}\right) + \left(-\frac{1}{|Q|}\sum_{y_j\in Q}\sum_{c=1}^{C}\left(1\middle|y_j = c\right\rangle)\log\widehat{y}_{jc}\right) \tag{10}$$

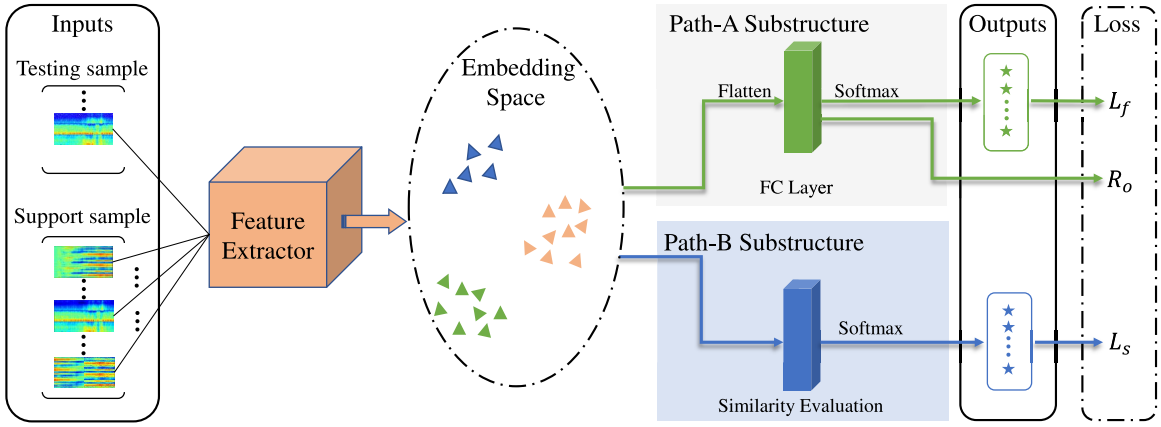

Fig. 7. Architecture of Dual-Path PN.

where $|S|$ and $|Q|$ represent the number of samples in the support set and the query set. y_i and y_j denote the ground truths of samples x_i^S and x_j^Q, respectively. $\widehat{y}_{ic}$ and $\widehat{y}_{jc}$ are the predicted probability of x_i^S and x_j^Q belonging to class c, respectively. In addition, $(1|y_j = c\rangle)$ means that if $y_j = c$ is true, return 1; otherwise, return 0.

In *Path-B*, a function Φ is used to evaluate the similarity between the test sample x_j^T and the prototype. In particular, the function Φ, in this work, is defined as the negative Euclidean distance or cosine similarity. In addition, the prototype is obtained from the feature representations Z^S by

$$\mathbf{P_c} = \frac{1}{|Z_c^S|} \sum_{z \in Z_c^S} z \tag{11}$$

where $\mathbf{P_c}$ denotes the prototype of gesture class c. Z_c^S is a subset of the elements labeled c in Z^S. Given the above, the predicted probability distribution over classes of x_j^T is calculated as follows:

$$\widehat{y}_{jk}^T = \frac{\exp\left(O_{jk}^T\right)}{\sum_{c=1}^{C} \exp\left(O_{jc}^T\right)} \text{ and } O_{ji}^T = \Phi\left(z_j^T, P_i\right) \tag{12}$$

where $\widehat{y}_{jk}^T$ presents the predicted probability of x_j^T belonging to class k. z_j^T is the feature representation of x_j^T.

In the training phase, we adopt a query set $Q = \{(x_j^Q, y_j^Q)\}_{j=1}^{O}$ to replace the testing set T. We can obtain the loss via cross-entropy function as follows:

$$L_s = \left(-\frac{1}{|Q|} \sum_{y_j \in Q} \sum_{c=1}^{C} \left(1|y_j = c\right) \log \widehat{y}_{jc} \right) \tag{13}$$

where $|Q|$ represents the number of samples in the query set. y_j denotes the ground truth of y_j^Q. $\widehat{y}_{jc}$ represents the predicted probability of x_j^Q belonging to class c by using the Dual-Path PN. In addition, $(1|y_j = c\rangle)$ means that if $y_j = c$ is true, return 1; otherwise, return 0.

Note that the combination of Path-B and the feature extractor is the original PN. Path-A is an assistant to construct an appropriate embedding space, and the details will be discussed in the next section. Hence, we adopt the outputs of Path-B as the final results of this model in the testing phase.

E. Orthogonal Regularization

The key to improving the original PN is the introduced Path-A substructure which can be regarded as a flexible cluster maker.

In Path-A, given a sample x whose feature representation is z and label is c, the predicted probability distribution over the gesture classes can be obtained by

$$\widehat{y}_k = \frac{\exp(H_k)}{\sum_{c=1}^{C} \exp(H_c)} \text{ and } H = Wz \tag{14}$$

where $W = [w_1, w_2, \ldots, w_C]^\top$ is the parameter of the FC layer. H_k is the kth value of vector H. Then, the predicted label of x is defined as follows:

$$\widehat{y} = \arg\max_k \widehat{y}_k, \text{ s.t. } k = 1, 2, \ldots, C. \tag{15}$$

In the training phase, the objective is to make $\widehat{y}$ identified with the ground truth c, which means $\widehat{y}_c$ is the largest probability value. Consequently, H_c is bigger than any H_k with $k = 1, 2, \ldots, c-1, c+1, \ldots, C$. Since $H = [H_1, H_2, \ldots, H_C]^\top = [w_1 \cdot z, w_2 \cdot z, \ldots, w_C \cdot z]^\top$, $w_c \cdot z$ is bigger than the others, i.e.,

$$|w_c||z|\cos\theta_c > |w_k||z|\cos\theta_k \tag{16}$$

where θ_i is the angle between z and w_i, $k = 1, 2, \ldots, c-1$, $c+1, \ldots, C$. In the training phase, $\cos\theta_c$ will increase so that (16) is satisfied. Namely, the feature vectors of all samples from class c will get closer to w_c based on the cosine distance.

Note that, it is unacceptable to increase $|w_c|$ or $|z|$ only during training. If only $|z|$ increases, there is no help for the establishment of (16). If only $|w_c|$ increases, it is suitable for the input samples from class c, but will be a disaster for the samples from the other classes. To the extreme, if $|w_C| \to \infty$, all the samples, whose feature representations are acute angles to w_C, will be predicted to belong to class c. In other words, there exists a subspace E which is half of the embedding space, and the label of any feature representations in it will be estimated as c.

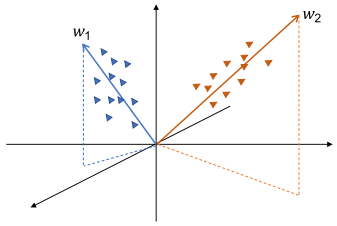

Fig. 8. Example of the relationship between the feature representations and the corresponding FC layer parameter vectors w_1 and w_2 in the embedding space, where different colors represent different classes.

Given the above, the feature representations Z_c of all the samples from class c will shift closer to the parameter vector w_c based on the cosine distance during the training phase, where $c = 1, 2, \ldots, C$. Hence, w_c can be regarded as the prototype of class c. As shown in Fig. 8, the feature representations from two classes distribute around the corresponding parameter vector w_1 or w_2 in the embedding space. Since the feature extractor F_Θ and w_C are learnable, Path-A substructure can be seen as a flexible cluster maker in the embedding space.

Furthermore, for the Dual-Path PN, an appropriate embedding space should have the following properties: 1) the feature representation clustering of the same class is compact and 2) the margin between different clusters should be as large as possible. Therefore, we construct an OR on the parameter $W = [w_1, w_2, \ldots, w_C]^\top$ of the FC layer to enhance the second property. In particular, we define the regularization as follows:

$$R_o = \frac{\lambda}{2}\left(\sum_{i=1}^{C}\sum_{j=1}^{C}\frac{w_i \cdot w_j}{|w_i||w_j|} + \sum_{k=1}^{C}|w_k|^2\right) \tag{17}$$

where λ is a hyperparameter. We use 1 for λ in this work. Additionally, R_o is composed of two parts: one is to make the angle θ_{ij} between w_i and w_j tend to 90°, the other is the L2 regularization.

F. Training Strategy

The episode-based strategy is adopted to train the Dual-Path PN, which mimics the process of the testing phase. Specifically, we have three data sets: 1) a training set P; 2) a testing set T; and 3) a support set S. P is a collection of labeled samples from the source domain. T consists of unlabeled samples and S is a set of labeled samples, which are both collected from the target domain. The label spaces of all three data sets are the same.

For a C-way K-shot CD model, the training details are illustrated in Algorithm 1. First, a subset S^P is randomly selected from P, which contains K samples for each class to mimic the support set S. Then, a query set $Q = \{(x_j^Q, y_j^Q)\}_{j=1}^{O}$ is randomly sampled from remainder samples in P to mimic the testing set. Afterwards, we propagate all the samples in the Dual-Path PN. We finally calculate the loss by

$$\text{Loss} = L_f + L_s + R_o. \tag{18}$$

In addition, Adam [44] optimization technique is used to optimize this model, and each episode consists of S^T and Q.

Algorithm 1 Episode-Based Training for C-Way K-Shot CD Problem

Input: Training set $\mathrm{P} = \{(\mathrm{x_i}, \mathrm{y_i})\}_{\mathrm{i}=1}^{\mathrm{N}}$, where $\mathrm{y_i} \in \{1, 2, \ldots, \mathrm{C}\}$ is the label of sample $\mathrm{x_i}$. P_k denotes a subset of P, containing all elements $(\mathrm{x_i}, \mathrm{y_i})$ with $\mathrm{y_i} = \mathrm{k}$.
Output: The loss J of a randomly generated episode.
for c **in** $\{1, 2, \ldots, \mathrm{C}\}$ **do**
  $\mathrm{S_c^P} \longleftarrow \mathrm{RandomSample}(P_\mathrm{c}, \mathrm{K})$ *// Select support samples*
  $\mathrm{Q_c} \longleftarrow \mathrm{RandomSample}(P_\mathrm{c} \setminus \mathrm{S_c^P},\ \mathrm{M/C})$ *// Select query samples*
  $\mathrm{Z_c^S} = \{\mathrm{F_\Theta}(\mathrm{x^S})\}$ for all $\mathrm{x^S} \in \mathrm{S_c^P}$ *// calculate feature representation*
  $\mathbf{P_c} = \frac{1}{|\mathrm{Z_c^S}|}\sum_{\mathrm{z} \in \mathrm{Z_c^S}} \mathrm{z}$ *// calculate prototype*
End for
$\mathrm{J} \longleftarrow 0$ *// Initialize loss*
for $\left(x_j^Q, y_j^Q\right)$ **in** $\mathrm{Q} = \mathrm{Q_1} \cup \mathrm{Q_2} \cup \ldots \cup \mathrm{Q_C}$ **do**
  $\mathrm{z_j^Q} = \mathrm{F_\Theta}(\mathrm{x_j^Q})$ *// Calculate feature representations*
  Calculate the $\mathrm{L_f}$ by using (8), (9), and (10).
  Calculate $\mathrm{L_s}$ by using (12) and (13)
  Calculate the Orthogonal Regularization by using (17).
  $\mathrm{J} \longleftarrow \mathrm{J} + \mathrm{L_f} + \mathrm{L_s} + \mathrm{R_o}$ *// Update loss*
End for

We repeat the abovementioned procedure until the network parameters change insignificantly.

V. Experimental Evaluation

In this section, we evaluate the effectiveness of WiGr on three data sets under both in-domain and cross-domain scenarios. Also, the performance of the different hyperparameter settings is discussed. The data set and code are available online.[1]

A. Experimental Setup

We conduct the experiments on three CSI-based HGR data sets. The details of these data sets are as follows.

1) Widar3.0 Data Set: Widar3.0 [22], a public CSI-based gesture data set consisting of two subdata sets, is contributed by researchers from Tsinghua University. One subdata set contains a total of 12000 CSI gesture samples. 16 users perform six gestures (push & pull, sweep, clap, slide, draw a circle, and draw zigzag) in five different positions and five orientations with each gesture repeated five times. Moreover, Widar3.0 includes three different environments (i.e., a classroom, an office, and a hall). The other holds 5000 CSI samples of two volunteers (one male and one female) drawing numbers 0–9 in a horizontal plane. Each number was drawn ten times and in five different orientations. The data is collected by six receivers and one transmitter equipped with three antennas. Here, the 802.11n CSI tool is adopted to provide 30 subcarriers for each link and send 1000 packages/s. The CSI sample of each gesture is $H \in C^{3\times3\times30\times T}$, where $T = 1000 \times t$ and t is the execution time of the gesture. Note that in this work, we only adopt the first part of this data set and one receiver's data are enough for our method.

2) ARIL Data Set: Wang *et al.* [19] proposed a CSI-based gesture data set for the joint task of activity recognition and indoor localization. One volunteer repeats each of six gestures (i.e., up, down, left, right, circle, and cross) 15 times

[1]https://github.com/Zhang-xie/WiGr

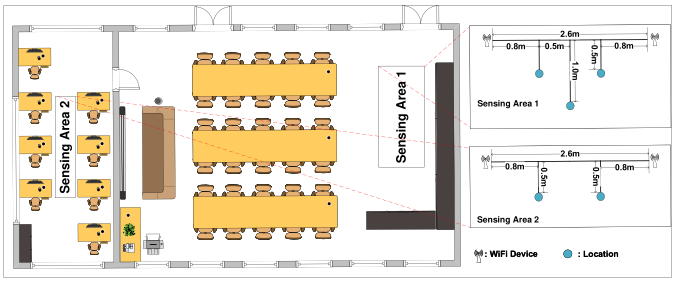

Fig. 9. CSI data acquisition scenarios.

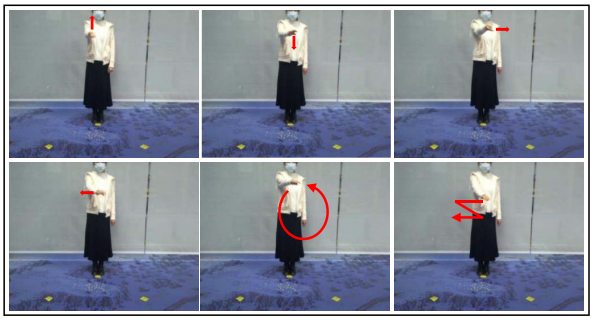

Fig. 10. Gestures in CSIDA.

in 16 locations in one room. The considered gestures are regarded as relevant for human–computer interaction applications in a smart home. The CSI samples in the ARIL data set were collected using the universal software radio peripherals (USRPs). For each gesture, USRPs collect 192 packets over 52 subcarriers and the shape of the corresponding CSI sample is $1 \times 52 \times 192$. We adopt this data set to test the cross-location recognition performance of the proposed system.

3) CSIDA Data Set: CSIDA is constructed by ourselves using one transmitter and one receiver equipped with the Atheros CSI tool [41]. Devices are set to work at monitor mode at 5 GHz. Furthermore, the channel bandwidth is 40 MHz with 114 subcarriers captured. The transmitter activates one antenna to transmit 1000 packets/s. The receiver is equipped with three antennas arranged in a line and separated by 1.6 cm. Meanwhile, the height of all antennas from the ground is 1.3 m. As shown in Fig. 9, we collect CSI measurements at different locations in two in-door Environments (i.e., a small office and a classroom) and perform six gestures (i.e., pull left, pull right, lift up, press down, draw a circle, and draw zigzag, illustrated in Fig. 10) that are suitable for human–computer interaction. In addition, each gesture is performed by five users (three males and two females) repeating ten times with a fixed execution time of 1.8 s.

For all the following experiments, we adopt the average accuracy as the metric of evaluation. The models are optimized by Adam [44] with a learning rate of 0.0005 and multistep decay scheduler, and the mini-batch size is 60. The hyperparameter λ is empirically set to 1, respectively. We implement the proposed system on PyTorch-1.8.0, framework on an Intel Xeon CPU E5-2630 v4, with an NVIDIA Titan X Pascal GPU and 32.0-GB RAM.

B. In-Domain Evaluation

We first evaluate the proposed method in the traditional way that all CSI sample sets (i.e., training set P, testing set T, and support set S) are collected from the same domain. Fig. 11 shows the confusion matrices of the in-domain evaluations on Widar3.0, ARIL, and CSIDA data sets. The proposed system WiGr achieves an accuracy of 92.7%, 86.8%, and 91.2% on Widar3.0, ARIL, and CSIDA data sets, respectively. The support set consists of only one sample for each gesture category, namely, the one-shot condition. We use 80% of the remaining data as training data and 20% as the testing set. Cosine similarity is employed, and both amplitude and phase values are used in this experiment.

C. Cross-Domain Evaluation

We further evaluate WiGr on the cross-domain experiments with the domain factors of environment, user, location, and orientation. For each cross-domain experiment, only one domain factor altered. We use holdout cross-validation on three data sets (i.e., Widar3.0, ARIL, and CSIDA).

Environment Variety: In this experiment, we select one room as the source domain and the other as the target domain. We conduct these experiments on both Widar3.0 and CSIDA data sets. In addition, Widar3.0 contains three different environments, i.e., a classroom, an office, and a hall. CSIDA data set is collected from two different environments (i.e., a classroom and an office). For each data set, we randomly select a pair of environments as the source domain and the target domain. Then, we repeat each experiment ten times to obtain the objective evaluation results. As shown in Fig. 12(a), the average accuracies are above 85% on the CSIDA data set and achieve the best performance of 95% in the four-shot testing. The evaluation on the Widar3.0 data set shows average accuracies of 60%, 65%, 75%, and 83% from one-shot to four-shot testing, respectively. The decline in accuracy from the CSIDA data set to the Widar3.0 data set is due to significant difference of the environmental layouts of the two data sets. Furthermore, since the prototype is the average of the support feature vectors, the accuracies are higher with the number of support samples increasing in both the Widar3.0 and the CSIDA data sets.

User Independent: The recognition accuracy across users is another criterion for cross-domain evaluation. As a human sensing technique, the ability to adapt to different users is critical for practicality. The challenges of cross-user recognition toward two aspects: 1) users who have different body features may influence diverse fluctuations in WiFi signals and 2) individuals exhibit different behavior patterns with respect to the same gesture. To evaluate the cross-user performance of WiGr, we train the model with the labeled CSI samples of one user and test with the CSI samples from other users. As Fig. 12(b) depicts, the average accuracy remains over 90% on the CSIDA data set. WiGr achieves the highest accuracy of 89% under the four-shot condition on the Widar3.0 data set. The difference in performance between testing on CSIDA and Widar3.0 can be attributed to the disparity in number of users; Widar3.0 has 16 users for testing while CSIDA only has five users. Overall, WiGr is robust to different users.

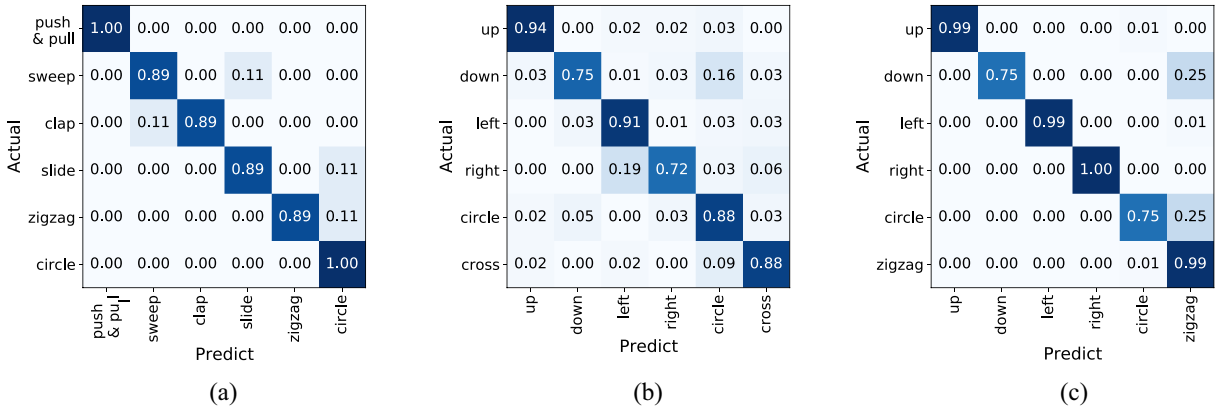

Fig. 11. Confusion matrices of in-domain evaluations on three gesture data sets. (a) Widar3.0 (92.7%). (b) ARIL (86.8%). (c) CSIDA (91.2%).

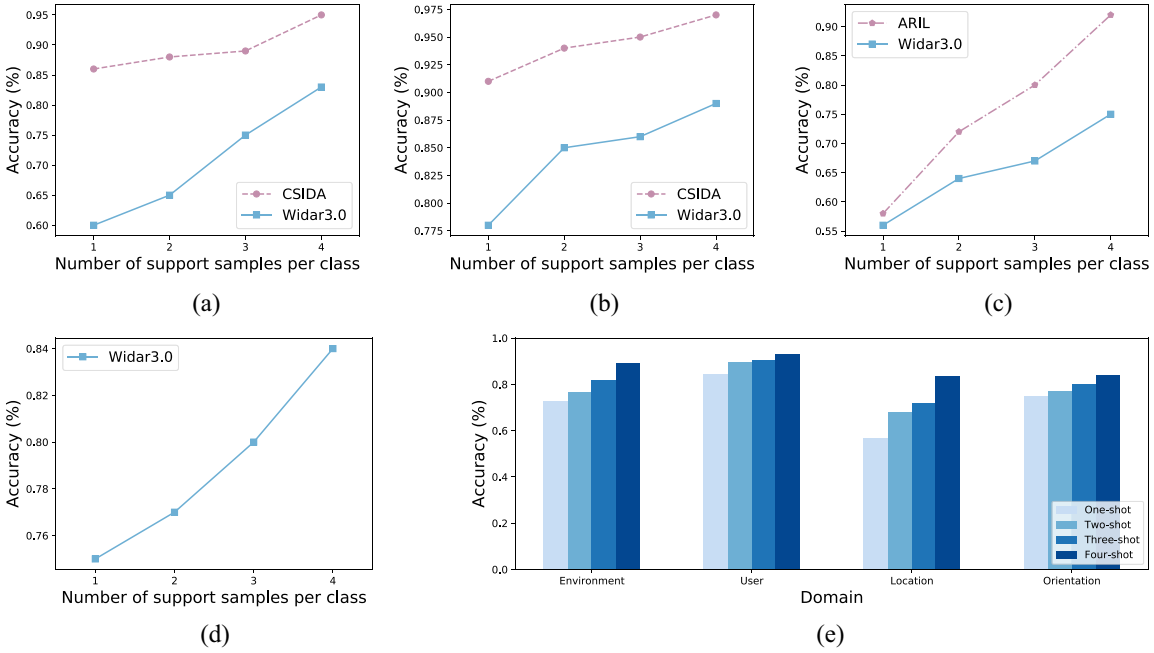

Fig. 12. Accuracies of cross-domain evaluations. (a) Cross-environment. (b) Cross-user. (c) Cross-location. (d) Cross-orientation. (e) Overall cross-domain accuracy.

Location Diversity: In this experiment, we adopt two data sets, ARIL and Widar3.0. In particular, the ARIL data set collected the CSI samples from 16 locations in one room. Meanwhile, Widar3.0 contains samples from five different locations in each environment (i.e., a classroom, an office, and a hall). We train the model with CSI samples collected at one location and then test with the remaining samples from the other locations in the same room. As shown in Fig. 12(c), WiGr reaches average accuracies between 55% and 60% on both ARIL and Widar3.0 under the one-shot condition. With the increase of the support samples, average accuracies achieve 92% and 75% on ARIL and Widar3.0 data sets, respectively. Due to the influence of different users and orientations, cross-location recognition on the Widar3.0 data set is more difficult than on the ARIL data set.

Orientation Sensitivity: For commonly gesture recognition applications (e.g., computer control and motion-sensing game), the orientations of the user may shift. Consequently, it is important to enhance the orientation-robust ability of the recognition system. We conduct the cross-orientation experiment on the Widar3.0 data set to evaluate the orientation sensitivity of WiGr. Specifically, we use CSI samples of one orientation to train the model and test it with the samples collected from the other four orientations performed by the same user at the same location. As illustrated in Fig. 12(d), the accuracy under across-orientation grows from 75% to 84% in accordance with the increase of the support samples. This shows that the performance is stable under cross-orientation situations.

Overall Cross-Domain Results: As shown in Fig. 12(e), the performance on cross-domain and cross-user evaluations is better than those in the cross-location and cross-orientation testing. Notably, the accuracy also grows with increases in the number of support samples. In particular, the average accuracies across domains are over 85% under the four-shot setting. We only need 24 labeled samples (i.e., four labeled samples for each gesture) from the new target domain to maintain a consistently high recognition accuracy without retraining the model.

TABLE I
METHODS COMPARATION

Method	Methodology	No. Parameters	Training Time of One Sample (s)	Testing Time of One Sample (s)	Mean Accuracy ±1.96*SE (%)
Widar3.0 [22]	domain-independent feature (handcraft)	531.238K	0.027355	0.010495	0.890±0.076
JADA [16]	domain-independent feature (Adversarial learning)	10453.976K	0.092482	0.006600	0.870±0.064
EI [23]	domain-independent feature (Adversarial learning)	103.943K	0.007594	0.000878	0.832±0.054
WiAG [24]	Transfer Learning	0	0.114120	0.016041	0.833±0.144
WiGr (ours)	Meta-learning/few-shot learning	3922.944K	0.051404	0.049227	0.901±0.103

'No. Parameters' denotes the number of parameters of the recognition model in each method. 'Mean Accuracy' is the average recognition accuracy of in-domain and cross-domain tests. SE represents the standard error of the mean.

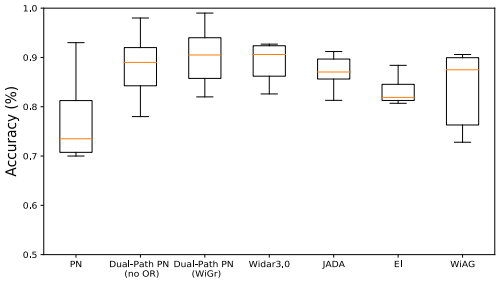

Fig. 13. Comparison of methods.

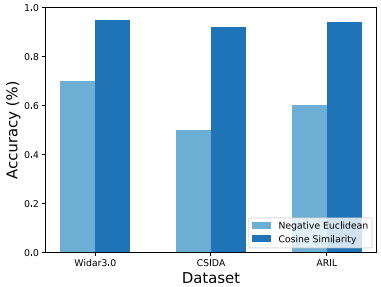

Fig. 14. Comparison of similarity measurements.

D. Discussion

In this section, we compare the performance of original PN, Dual-Path PN (without OR), Dual-Path PN (ours), and several state-of-the-art methods in cross-domain HGR. Then, we evaluate the impact of the similarity measurements and the input types of CSI data.

1) Comparison of Methods: We first compare WiGr against several state-of-the-art methods, Widar3.0 [22], JADA [16], EI [23], and WiAG [24] that have been introduced in Section II. Table I illustrates the differences between these works. Specifically, the methodology (first column) is the type of CSI-based cross-domain sensing techniques discussed in Section II. The second column is the number of parameters. Our method WiGr has the second largest number of parameters. JADA has the most parameters since it needs three neural networks to form the proposed framework. However, WiAG has no parameters in the recognition model because it uses the K-Nearest-Neighbor algorithm to perform the classification. The third and fourth columns are the training and testing time of one CSI sample, respectively. The training time of WiGr is acceptable compared with the others. Since WiAG needs to generate virtual samples for all possible domains, it is the most time-consuming method. The testing time of all the methods, except WiGr, is significantly reduced compared to the training time. This is because they can reduce part of the process in the testing phase. The last column proposes the mean accuracy (with the standard error of the mean) of the in-domain and the cross-domain evaluations. WiGr achieves state-of-the-art recognition accuracy. Nevertheless, the recognition accuracy of WiGr has a high standard error that is not as stable as other methods.

The crucial part of the proposed HGR system WiGr is the Dual-Path PN which is a modified version based on the original PN. To test the effectiveness of the improvements, we compare the performances of the original PN, Dual-Path PN (without OR), and Dual-Path PN under the cross-domain configuration. In addition, as illustrated in Fig. 13, the average accuracy of Dual-Path PN (without OR) is 14% higher than that of PN. Hence, the Path-A substructure is effective for constructing an appropriate embedding space. Further, the Dual-Path PN has better performance than that without OR, which proves that OR helps expand the gaps among different clusters in the embedding space. WiGr achieves better performance than JADA [16], EI [23], and WiAG [24] and comparable average accuracy with Widar3.0 [22]. However, the performance of WiGr is not as stable as Widar3.0. For the utility of the WiGr system, we only use the CSI samples from one receiver to realize the recognition task rather than at least three receivers as needed in Widar3.0.

2) Comparison of Similarity Measurements: In Path-B, the similarity measurement is an important factor. To compare the impact of two different metric methods (i.e., the negative Euclidean distance and cosine similarity). We conduct in-domain one-shot experiments on the three data sets. Further, the input type is the amplitude of CSI. As demonstrated in Fig. 14, the average accuracy of WiGr based on cosine similarity is better than that based on the negative Euclidean distance. As aforementioned, the Path-A substructure and OR are dedicated to constructing an appropriate embedding space for cosine similarity. Hence, adopting cosine similarity in Path-B is more suitable than the negative Euclidean distance.

3) Impact of Input Types: Since CSI sample is a complex tensor, we decompose it into amplitude and phase. To evaluate the impact of different input types (i.e., amplitude only, phase only, and amplitude-phase concatenated), we conduct cross-environment experiments on the CSIDA data set and Widar3.0 data set. As shown in Fig. 15, the average accuracy of WiGr with phase as the input type is better than that

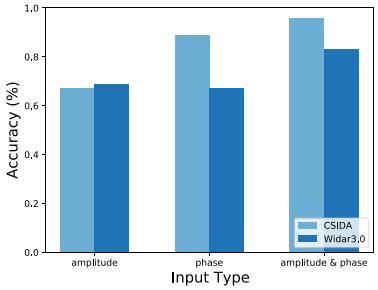

Fig. 15. Comparison of input types.

with amplitude, and the best performance is that with both amplitude and phase. In addition, the difference in accuracy with phase or amplitude as input, in CSIDA, are more significant than in Widar3.0 data set. The reason for the above phenomenon is that the change of phase is caused by the motion of subjects in the propagation path. Meanwhile, the variation of amplitude is related to the obstruction in signal by subjects in different signal paths. For gesture recognition, the motion of hands directly connected to the gesture type. Hence, the change of phase contains more information than variation of amplitude. Consequently, this observation suggests that phase may serve as a better input than amplitude. The discrepancy in capture tools adopted in CSIDA and Widar3.0 may cause the above phenomenon.

4) Limitations and Future Works: There are several limitations with our proposed model, which can serve as fruitful directions for further investigation.

First, the performance of WiGr is not stable as it is sensitive to the representative ability of prototypes. To expand, the representative ability may degrade for two reasons: 1) the number of labeled samples for each class is too small to compute the prototypes and 2) the labeled sample in the support set is far away from its ground-truth center. To address this problem, we can try to introduce prior knowledge to enhance the prototype estimation.

Second, WiGr is only suitable for one-domain-cross scenarios (i.e., cross-room only, cross-location only, cross-user only, and cross-orientation only). We are interested in expanding WiGr to the multidomain-cross scenario. For example, training the model on the CSI samples of user A performed in room 1 and deploying the model in room 2 for user B.

Third, the learning process of WiGr is computationally intensive. As mentioned above, the WiGr is sensitive to the prototypes. As a result, we need to train the Dual-Path PN multiple times with different sampling for the support set. In addition, the CSI is time-series data. As such, we need to expand the Dual-Path PN to the sequence data. Moreover, in the testing procession, WiGr can access unlabeled data which is helpful to estimate the data distribution. Therefore, more studies are needed to enable online learning with streaming data on lightweight devices.

VI. Conclusion

In this article, we propose a WiFi-based cross-domain gesture recognition system, WiGr. To begin with, we propose a novel Dual-Path PN to identify common human gestures with consistently high accuracy under domain dynamics. We then provide a regularization, namely, OR, to increase the gaps between different clusters in the embedding space. Next, we construct a WiFi-based HGR data set, namely, CSIDA, for testing the performance of WiGr. Finally, we conduct comprehensive experiments on the CSIDA data set and the other two public data sets (i.e., Widar3.0 and ARIL). The evaluation suggests that WiGr achieves 86.8%, 91.2%, and 92.7% in-domain recognition accuracy on ARIL, CSIDA, and Widar3.0 data sets, respectively. Further, WiGr achieves high accuracy in cross-domain experiments without retraining under the four-shot setting. Specifically, WiGr obtains 89%, 93%, 83.5%, and 84% average accuracies in cross-environment, cross-user, cross-location, and cross-orientation experiments, respectively.

Acknowledgment

The authors would like to thank Junyan Li, Pengli Hu, and Yasong An from Sun Yat-sen University for their cooperation in collecting experimental data.

References

[1] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: A survey," *Artif. Intell. Rev.*, vol. 43, no. 1, pp. 1–54, 2015.

[2] J. Qi, G. Jiang, G. Li, Y. Sun, and B. Tao, "Intelligent human-computer interaction based on surface EMG gesture recognition," *IEEE Access*, vol. 7, pp. 61378–61387, 2019.

[3] H.-C. Shih, "Hand gesture recognition using color-depth association for smart home," in *Proc. 1st Int. Cogn. Cities Conf. (IC3)*, Okinawa, Japan, 2018, pp. 195–197.

[4] H. Zou, Y. Zhou, J. Yang, H. Jiang, L. Xie, and C. J. Spanos, "WiFi-enabled device-free gesture recognition for smart home automation," in *Proc. IEEE 14th Int. Conf. Control Autom. (ICCA)*, Anchorage, AK, USA, 2018, pp. 476–481.

[5] M. Shahzad and S. Zhang, "Augmenting user identification with WiFi based gesture recognition," in *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 2, no. 3, pp. 1–27, 2018.

[6] C.-Y. Hsu, R. Hristov, G.-H. Lee, M. Zhao, and D. Katabi, "Enabling identification and behavioral sensing in homes using radio reflections," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2019, pp. 1–13.

[7] H. Zhao, S. Wang, G. Zhou, and D. Zhang, "Ultigesture: A wristband-based platform for continuous gesture control in healthcare," *Smart Health*, vol. 11, pp. 45–65, Jan. 2019.

[8] K. Geng and G. Yin, "Using deep learning in infrared images to enable human gesture recognition for autonomous vehicles," *IEEE Access*, vol. 8, pp. 88227–88240, 2020.

[9] Z. Ren, J. Meng, and J. Yuan, "Depth camera based hand gesture recognition and its applications in human-computer-interaction," in *Proc. 8th Int. Conf. Inf. Commun. Signal Process.*, Singapore, 2011, pp. 1–5.

[10] B. Fang, F. Sun, H. Liu, and C. Liu, "3D human gesture capturing and recognition by the IMMU-based data glove," *Neurocomputing*, vol. 277, pp. 198–207, Feb. 2018.

[11] M. Kim, J. Cho, S. Lee, and Y. Jung, "IMU sensor-based hand gesture recognition for human-machine interfaces," *Sensors*, vol. 19, no. 18, p. 3827, 2019.

[12] Z. Peng, C. Li, J.-M. Muñoz-Ferreras, and R. Gómez-García, "An FMCW radar sensor for human gesture recognition in the presence of multiple targets," in *Proc. 1st IEEE MTT-S Int. Microw. Bio Conf. (IMBIOC)*, Gothenburg, Sweden, 2017, pp. 1–3.

[13] Z. Wang, Z. Yu, X. Lou, B. Guo, and L. Chen, "Gesture-radar: A dual doppler radar based system for robust recognition and quantitative profiling of human gestures," *IEEE Trans. Human-Mach. Syst.*, vol. 51, no. 1, pp. 32–43, Feb. 2021, doi: 10.1109/THMS.2020.3036637.

[14] M. A. A. Haseeb and R. Parasuraman, "Wisture: RNN-based learning of wireless signals for gesture recognition in unmodified smartphones," 2017. [Online]. Available: arXiv:1707.08569.

[15] X. Ma, Y. Zhao, L. Zhang, Q. Gao, M. Pan, and J. Wang, "Practical device-free gesture recognition using WiFi signals based on metalearning," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 228–237, Jan. 2020, doi: 10.1109/TII.2019.2909877.

[16] H. Zou, J. Yang, Y. Zhou, and C. J. Spanos, "Joint adversarial domain adaptation for resilient WiFi-enabled device-free gesture recognition," in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Orlando, FL, USA, Dec. 2018, pp. 202–207, doi: 10.1109/ICMLA.2018.00037.
[17] Z. Han, L. Guo, Z. Lu, X. Wen, and W. Zheng, "Deep adaptation networks based gesture recognition using commodity WiFi," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Seoul, South Korea, May 2020, pp. 1–7, doi: 10.1109/WCNC45663.2020.9120726.
[18] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: Indoor localization via channel response," *ACM Comput. Surveys*, vol. 46, no. 2, pp. 1–32, Dec. 2013, doi: 10.1145/2543581.2543592.
[19] F. Wang, J. Feng, Y. Zhao, X. Zhang, S. Zhang, and J. Han, "Joint activity recognition and indoor localization with WiFi fingerprints," *IEEE Access*, vol. 7, pp. 80058–80068, 2019, doi: 10.1109/ACCESS.2019.2923743.
[20] Y. Ma, G. Zhou, S. Wang, H. Zhao, and W. Jung, "SignFi: Sign language recognition using WiFi," in *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 2, no. 1, pp. 1–21, Mar. 2018, doi: 10.1145/3191755.
[21] S. Palipana, D. Rojas, P. Agrawal, and D. Pesch, "FallDeFi: Ubiquitous fall detection using commodity Wi-Fi devices,"in *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 1, no. 4, pp. 1–25, Jan. 2018, doi: 10.1145/3161183.
[22] Y. Zheng *et al.*, "Zero-effort cross-domain gesture recognition with Wi-Fi," in *Proc. 17th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, New York, NY, USA, Jun. 2019, pp. 313–325, doi: 10.1145/3307334.3326081.
[23] W. Jiang *et al.*, "Towards environment independent device free human activity recognition," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, New York, NY, USA, Oct. 2018, pp. 289–304, doi: 10.1145/3241539.3241548.
[24] A. Virmani and M. Shahzad, "Position and orientation agnostic gesture recognition using WiFi," in *Proc. 15th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, New York, NY, USA, Jun. 2017, pp. 252–264, doi: 10.1145/3081333.3081340.
[25] C. Xiao, D. Han, Y. Ma, and Z. Qin, "CsiGAN: Robust channel state information-based activity recognition with GANs," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10191–10204, Dec. 2019, doi: 10.1109/JIOT.2019.2936580.
[26] J. Zhang, Z. Tang, M. Li, D. Fang, P. Nurmi, and Z. Wang, "CrossSense: Towards cross-site and large-scale WiFi sensing," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, New York, NY, USA, Oct. 2018, pp. 305–320, doi: 10.1145/3241539.3241570.
[27] B. Sheng, F. Xiao, L. Sha, and L. Sun, "Deep spatial–temporal model based cross-scene action recognition using commodity WiFi," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3592–3601, Apr. 2020, doi: 10.1109/JIOT.2020.2973272.
[28] 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. 10763–10772, Dec. 2019, doi: 10.1109/JIOT.2019.2941527.
[29] Z. Shi, J. A. Zhang, Y. D. R. Xu, and Q. Cheng, "Environment-robust device-free human activity recognition with channel-state-information enhancement and one-shot learning," *IEEE Trans. Mobile Comput.*, early access, Jul. 28, 2020, doi: 10.1109/TMC.2020.3012433.
[30] H. Zou, Y. Zhou, J. Yang, H. Liu, H. P. Das, and C. J. Spanos, "Consensus adversarial domain adaptation," in *Proc. AAAI*, vol. 33, Jul. 2019, pp. 5997–6004, doi: 10.1609/aaai.v33i01.33015997.
[31] G. Lan, B. Heit, T. Scargill, and M. Gorlatova, "GazeGraph: Graph-based few-shot cognitive context sensing from human visual behavior," in *Proc. 18th Conf. Embedded Netw. Sens. Syst.*, New York, NY, USA, Nov. 2020, pp. 422–435, doi: 10.1145/3384419.3430774.
[32] A. Zhao *et al.*, "Domain-adaptive few-shot learning," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2021, pp. 1390–1399.
[33] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," Jun. 2017. [Online]. Available: http://arxiv.org/abs/1703.05175.
[34] W. K. Zegeye, S. B. Amsalu, Y. Astatke, and F. Moazzami, "WiFi RSS fingerprinting indoor localization for mobile devices," in *Proc. IEEE 7th Annu. Ubiquitous Comput. Electron. Mobile Commun. Conf. (UEMCON)*, New York, NY, USA, 2016, pp. 1–6.
[35] M. Ibrahim, M. Torki, and M. ElNainay, "CNN based indoor localization using RSS time-series," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Natal, Brazil, 2018, pp. 01044–01049.
[36] Y. Ma, G. Zhou, and S. Wang, "WiFi sensing with channel state information: A survey," *ACM Comput. Surveys*, vol. 52, no. 3, pp 1–36, Jun. 2019, doi: 10.1145/3310194.
[37] A. J. Storkey and M. Sugiyama, "Mixture regression for covariate shift," in *Advances in Neural Information Processing Systems*, vol. 19. Red Hook, NY, USA: Curran, 2007, p. 1337.
[38] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 1, p. 53, Jan. 2011, doi: 10.1145/1925861.1925870.
[39] F. Gringoli, M. Schulz, J. Link, and M. Hollick, "Free your CSI: A channel state information extraction platform for modern Wi-Fi chipsets," in *Proc. 13th Int. Workshop Wireless Netw. Testbeds Exp. Eval. Characterization*, New York, NY, USA, Oct. 2019, pp. 21–28, doi: 10.1145/3349623.3355477.
[40] M. Atif, S. Muralidharan, H. Ko, and B. Yoo, "Wi-ESP—A tool for CSI-based device-free Wi-Fi sensing (DFWS)," *J. Comput Design Eng.*, vol. 7, no. 5, pp. 644–656, 2020, doi: 10.1093/jcde/qwaa048.
[41] Y. Xie, Z. Li, and M. Li, "Precise power delay profiling with commodity Wi-Fi," *IEEE Trans. Mobile Comput.*, vol. 18, no. 6, pp. 1342–1355, Jun. 2019, doi: 10.1109/TMC.2018.2860991.
[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778. Accessed: Oct. 23, 2020. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html
[43] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2015, pp. 448–456. Accessed: Mar. 9, 2021. [Online]. Available: http://proceedings.mlr.press/v37/ioffe15.html
[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: arXiv:1412.6980, 2014.

Xie Zhang received the B.S. degree from Sun Yat-sen University, Guangzhou, China, in 2019, where he is currently pursuing the M.S. degree in pattern recognition and intelligence system.

His research interests include human activity recognition, wireless sensing, and artificial intelligence.

Chengpei Tang received the Ph.D. degree in radio physics from the Department of Electronics and Communication Engineering, Sun Yat-sen University, Guangzhou, China, in 2007.

His research fields are wireless sensor networks, artificial intelligence, and indoor-positioning systems.

Kang Yin received the bachelor's degree from Sun Yat-sen University, Guangzhou, China, in 2019, where he is currently pursuing the master's degree.

His current interests include artificial intelligence and wireless sensing.

Qingqian Ni received the bachelor's degree in engineer from Sun Yat-sen University, Guangzhou, China. He is currently pursuing the master's degree in data science with the University of Hong Kong, Hong Kong.

Her current interests include artificial intelligence and data science.