Imbalanced Semi-Supervised Learning for WiFi Gesture Recognition via Dynamic Threshold-Based Spatio-Temporal Attention Networks

Qihua Feng, Chunhui Duan, Jiawei Xue, Chaozhuo Li, Feiran Huang, Xi Zhang, *Member, IEEE*, Jian Weng, *Senior Member, IEEE*, and Philip S. Yu, *Life Fellow, IEEE*

***Abstract*—WiFi sensing advancements facilitate the capture of human gestures from wireless signals, ensuring both privacy preservation and robustness under low-light conditions. Deep learning-based WiFi Human Gesture Recognition (HGR) demonstrates remarkable performance in handling complex gestures. To reduce labeling efforts, recent years have seen the emergence of semi-supervised WiFi HGR, leveraging massive amounts of unlabeled data. However, existing semi-supervised schemes often assume a balanced class distribution and utilize a fixed threshold for selecting pseudo-labels of unlabeled samples, leading to low performance for minority classes and decreased model generalization on real-world imbalanced datasets. To address this issue, we propose a novel semi-supervised WiFi HGR approach with dynamic pseudo-labeling thresholds to handle imbalanced class distribution, incorporating Spatial-Temporal Attention (STA) networks. Unlike using a fixed threshold for all unlabeled samples, our design implements class-independent thresholds for different classes, dynamically adjusting them by encoding pseudo-label distribution during training. To emphasize critical features in informative areas within the WiFi signals, we incorporate both spatial self-attention and temporal attention mechanisms to dynamically learn salient features and identify pivotal frames, respectively. Moreover, we introduce adaptive WiFi data augmentations that propel the semi-supervised framework and enhance model robustness. Experimental results on the Widar3.0 dataset reveal that our approach outperforms existing semi-supervised methods by large margins in accuracy, effectively mitigating imbalanced bias and enhancing model generalization.**

***Index Terms*—Gesture recognition, WiFi, semi-supervised learning, attention networks, imbalanced classification.**

Received 25 November 2024; revised 13 July 2025; accepted 22 July 2025. Date of publication 25 July 2025; date of current version 3 December 2025. This work was supported in part by the Key Program of the National Natural Science Foundation of China under Grant 62232004, in part by the Beijing Institute of Technology Research Fund Program for Young Scholars, in part by the National Natural Science Foundation of China under Grant 62332007 and U22B2028, in part by the Science and Technology Major Project of Tibetan Autonomous Region of China under Grant XZ202201ZD0006G, in part by the Open Research Fund of Machine Learning and Cyber Security Interdiscipline Research Engineering Center of Jiangsu Province under Grant SDGC2131, in part by the National Joint Engineering Research Center of Network Security Detection and Protection Technology, Guangdong Key Laboratory of Data Security and Privacy Preserving, Guangdong Hong Kong Joint Laboratory for Data Security and Privacy Protection, and Engineering Research Center of Trustworthy AI, Ministry of Education, and in part by the Fundamental Research Funds for the Central Universities under Grant 21624329 and Grant 12625611. Recommended for acceptance by Y. Liu. *(Corresponding author: Chunhui Duan.)*

Qihua Feng, Chunhui Duan, and Jiawei Xue are with the Beijing Institute of Technology, Beijing 100081, China (e-mail: duanch@bit.edu.cn).

Chaozhuo Li and Xi Zhang are with the Beijing University of Posts and Telecommunications, Beijing 100876, China.

Feiran Huang and Jian Weng are with Jinan University, Guangzhou 510632, China.

Philip S. Yu is with the Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607 USA.

The code is publicly at https://github.com/onlinehuazai/Semi-Fi.

Digital Object Identifier 10.1109/TMC.2025.3592965

I. Introduction

HUMAN Gesture Recognition (HGR) encompasses the computing process of comprehending and interpreting hand movement commands, finding extensive practical applications in domains such as human-computer interaction [1], smart homes [2], smart medical [3], virtual reality [4], etc. Conventional camera-based gesture recognition systems [5], [6] rely on analyzing hand motion through pictures or videos, but they are hindered by limitations in low-light and dark environments [7], [8]. Besides, these camera-based approaches fall short in terms of privacy preservation [9], [10] as they involve personal and private human image data. In contrast, the rise of the Internet of Things (IoT) and the widespread adoption of pervasive computing have paved the way for WiFi-based gesture recognition, offering the ability to overcome challenges posed by low-light conditions while safeguarding user privacy [11], [12]. As a result, WiFi gesture recognition has emerged as a compelling research area, attracting a growing number of researchers to explore its potential [13], [14], [15], [16], [17], [18].

A typical workflow of existing WiFi gesture recognition approaches involves the extraction of features from wireless signals, followed by the utilization of deep learning models to learn gestures based on these features [7], [14], [15], [16], [17], [19], [20], [21], [22], [23], [24], [25]. Despite the notable breakthroughs and commendable recognition accuracy attained by these related works in HGR through Deep Neural Networks (DNNs), these learning-based HGR schemes heavily rely on fully supervised learning, which incurs additional labor overhead to label all data. Especially, unlike conventional images, WiFi signals are hard to comprehend visually, thus further exacerbating the complexities in data labeling. For example, Fig. 1(b) shows a gesture sample with 4 frames. Apparently,

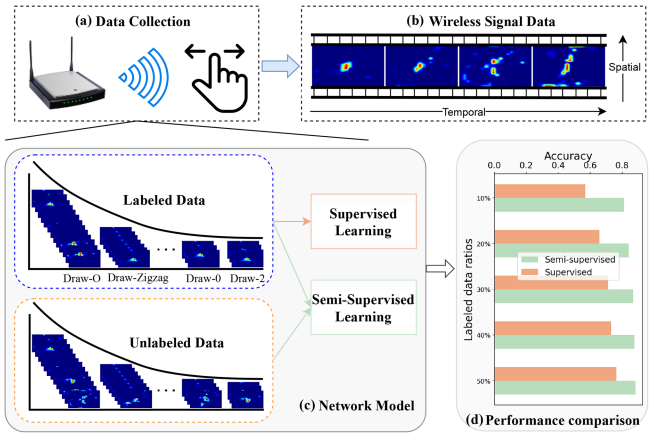


Fig. 1. Illustration of our imbalanced semi-supervised WiFi HGR scenario: (a) WiFi data collection with imbalanced gesture classes. (b) Visualization of the signal data, which is inherently incomprehensible to the human eye. Due to the high labeling costs and the incomprehensibility of WiFi data, only a small portion is labeled to reduce effort. (c) Training of the network model using imbalanced labeled and unlabeled data. The results in (d) demonstrate that SSL significantly outperforms supervised learning in terms of accuracy.

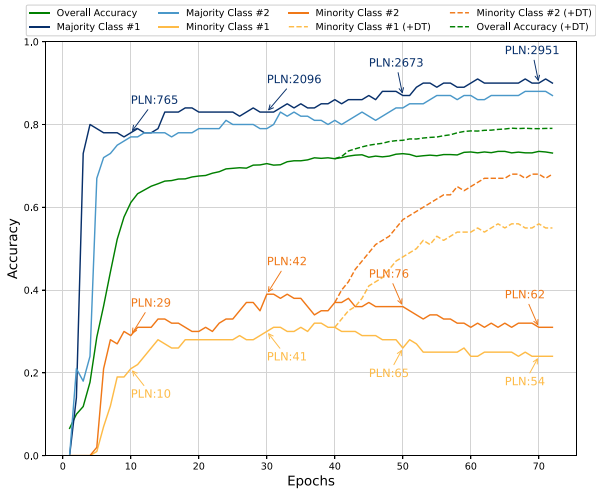


Fig. 2. Accuracy curves of overall, majority classes, and minority classes. The proposed DT mechanism is activated at the pre-defined epoch and improves accuracy by large margins.

it is intractable for human eyes to recognize which gesture it belongs to from the wireless signals. To alleviate the labeling burden, a few pioneer semi-supervised WiFi HGR works [26], [27] propose to only label a portion of the data, leaving a massive amount of unlabeled data. These works utilize Semi-Supervised Learning (SSL) [28], [29] to learn from labeled data and take full advantage of unlabeled data to capture the underlying distribution over the entire data. For example, UDARF [27] assigns pseudo-labels to unlabeled WiFi samples with predicted probabilities exceeding a fixed threshold, such as 0.95.

Existing semi-supervised WiFi HGR methods have achieved decent performance by leveraging the information from unlabeled data, but these methods often assume that the dataset is balanced, neglecting the imbalanced nature of real-world datasets. For example, UDARF [27] evaluates performance on a subset of Widar3.0 dataset [14] by selecting only 6 balanced classes, while the entire Widar3.0 dataset is imbalanced, consisting of 22 imbalanced classes. It is known that models tend to be biased towards majority classes when dealing with imbalanced datasets [30], [31]. Unfortunately, when we evaluate SSL performance of UDARF on the complete imbalanced Widar3.0 dataset, the experimental findings demonstrate that the model bias exacerbates as the training progresses, even though the overall performance improves. Fig. 2 illustrates the accuracy curves for two minority classes (each with approximately 500 samples), two majority classes (each with about 5000 samples), and overall dataset. While the accuracy of overall and majority classes increases during training, the accuracy of the minority classes increases first but then decreases, indicating a worsening of the model bias.

We observe that this aggravation phenomenon stems from the use of a fixed threshold to generate pseudo-labels for unlabeled data. This fixed threshold method in UDARF [27] assigns pseudo-labels to instances based on their predicted probabilities surpassing the threshold. Minority classes gradually lose unlabeled samples with proxy labels because their predicted probabilities do not exceed the fixed threshold. We visualize the Number of Pseudo-Labels (PLN) of majority and minority classes at epoch {10, 30, 50, 70}, shown in Fig. 2. The PLN for minority classes initially increases but then decreases. In contrast, the PLN for majority classes continues to increase dramatically. Therefore, the fixed threshold is unsuitable for minority classes, as it hinders minority classes with more PLN, thus exacerbating model bias and leading to severe performance degradation in minority classes.

Improving spatial-temporal representation ability of DNNs and employing effective data augmentations are vital for semi-supervised WiFi HGR to enhance model generalization. For instance, as illustrated in Fig. 1(b), the blue areas in each spatial frame represent trivial features with little informative content. Similarly, certain frames may contain more non-blue significant regions in the temporal dimension. Hence, learning importance relations among spatial-temporal features enhances the model's fine-grained recognition ability. However, existing schemes often employ basic Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) to learn WiFi signals, limiting the model's capability to capture informative spatial-temporal features. Moreover, unlike image recognition which has powerful augmentation methods, WiFi signal cannot simply employ image transformation (e.g., color augmentation) due to its data attribute [32], [33]. Current WiFi HGR either does not employ augmentation methods [14] or just adds random noise [26] to WiFi signal data. Adding random noise helps, but the effect is limited due to lacking coupling.

In contrast to previous semi-supervised WiFi HGR approaches, our goal is to dynamically assign different thresholds for all classes, alleviating model bias on imbalanced datasets. Additionally, we aim to enhance the model's ability by focusing

on salient spatial-temporal features and employing adaptive data augmentations for WiFi signal, which enables model to learn more discriminative representations of signal data.

To this end, we propose a novel scheme for imbalanced semi-supervised WiFi HGR that utilizes a Dynamic Threshold (DT) mechanism and incorporates spatial-temporal networks. First, during each training epoch, we dynamically adjust the class-dependent thresholds by considering the class-wise pseudo-label distributions. The thresholds of the minority classes are dynamically adjusted by ranking their predicted probability values and aligning with pseudo-labels of majority class proportion distributions. Second, to effectively focus on the informative feature areas, our proposed approach leverages Spatial-Temporal Attention (STA) networks to capture and emphasize the most relevant information. Our STA implementation employs self-attention to highlight salient spatial features within each frame and temporal attention to dynamically learn the importance scores of all frames. Lastly, to enhance model robustness, we develop adaptive data augmentation techniques for WiFi signal data, focusing on adversarial examples and time series augmentations.

We evaluate our approach on the Widar3.0 dataset, and the results demonstrate its exceptional performance with limited labeled data. It outperforms only supervised learning by approximately $20\% \sim 25\%$ in terms of accuracy and achieves $0.5 \sim 1.5\%$ higher accuracy compared to current semi-supervised approaches. The proposed DT mechanism improves accuracy of overall and minority classes by large margins, as shown in Fig. 2. Furthermore, our findings highlight the significant improvement in model accuracy resulting from the proposed STA structure and adaptive data augmentations. In summary, our work presents the following core contributions:

1) We propose a novel semi-supervised WiFi HGR scheme that addresses the challenge of working on imbalanced dataset, alleviating model bias by employing dynamic class-independent thresholds instead of fixed ones. To our knowledge, we are the first to explore imbalanced semi-supervised learning in WiFi HGR.
2) To emphasize informative areas in WiFi signal, our proposed scheme incorporates STA structure that dynamically concentrates on spatial salient values and assigns higher scores to more important temporal frames. Additionally, we introduce adaptive WiFi data augmentations to effectively enhance model robustness.
3) Extensive experimental evaluations on the Widar3.0 dataset demonstrate that our approach outperforms existing methods by a wide margin, making it a viable solution for imbalanced semi-supervised WiFi HGR.

The rest of this paper is organized as follows. Section II introduces the related work. We present background and preliminary analysis in Section III. We elaborate on the technical details of our study in Section IV. In Section V, we describe the implementation and evaluation of our system. Section VI highlights potential future research directions. Finally, Section VII summarizes the conclusion.

II. Related Work

Existing WiFi HGR approaches are broadly classified into two categories: model-based and learning-based methods [14].

A. Model-Based WiFi HGR

Model-based HGR techniques establish physical connections between WiFi signals and gestures without the need for training. These methods extract various features, such as Channel State Information (CSI), Doppler Frequency Shift (DFS), and Angle of Arrival/Departure (AoA/AoD), and utilize pattern matching to recognize gestures. For instance, WiGest [34] employs Received Signal Strength Indicator (RSSI) and performs similarity matching with a predefined gesture feature database. WiSee [35] extracts DFS and employs k-Nearest Neighbor (kNN) to recognize hand gestures. Similarly, [36], [37] utilize CSI and the Dynamic Time Warping (DTW) algorithm to match gestures. WiTraj [38] employs the Fresnel zone model for recognition, while WiDraw [39] utilizes the AoA parameter. However, when dealing with complex gestures, traditional model-based HGR approaches become inadequate and intractable [14], [18]. To overcome such limitation, learning-based HGR methods leverage DNNs to effectively learn fine-grained complex gestures.

B. Learning-Based WiFi HGR

Supervised learning-based WiFi HGR: These schemes utilize DNNs to train models for WiFi gesture recognition, with all training sets labeled. For example, the works [7], [15], [16], [19], [21] have focused on extracting CSI feature and leveraging CNNs or RNNs to learn gestures. WiHi [22] employs CNN-RNN structure, and the work [23] utilizes Transformer [40] to learn gesture. Tong et al. [24] utilize a gesture truncation algorithm to remove redundant information and a deep attention model to learn CSI. The works [16], [25] propose to learn cross-domain gestures by using metric learning and adversarial learning techniques, respectively. Since CSI is highly relevant to the surrounding environment, resulting in significant target-independent noise within these features [14], [20], Widar3.0 [14] proposes the domain-independent Body-coordinate Velocity Profile (BVP) feature. Moreover, Widar3.0 recently opens by far the largest publicly available WiFi gesture dataset. Unlike other DNNs-based schemes that require retraining in new environments [14], the model of Widar3.0 does not due to its advanced BVP feature. The open dataset in Widar3.0 not only provides fair comparison conditions but also facilitates further research in the field of WiFi gesture recognition [17], [18], [20], [21], [22], [24], [25], [41]. Additionally, some works, such as [17], [42], propose few-shot learning for WiFi HGR. It is worth noting that although [17], [42] are also supervised learning, their setups differ significantly from the aforementioned supervised learning schemes. Specifically, few-shot learning in [17], [42] requires training the model on an extra-large labeled base dataset (e.g., SignFi dataset [15]) first and then fine-tuning it with the few target source dataset (e.g, Widar3.0 dataset [14]), where the base and target datasets are from different sources but with complete

labeling. In contrast, the above supervised schemes [7], [14], [15], [16], [21], [22], [23], [24], [25] only use a target dataset source, such as Widar3.0 dataset, for train and test. For a fair comparison, our study also focuses on training and testing with a single dataset source. Although existing supervised WiFi HGR methods facilitate gesture recognition accuracy with DNNs, the labeling challenges and associated overheads have led to investigations of SSL-based WiFi HGR methods in recent years.

Semi-Supervised learning-based WiFi HGR: This type of WiFi HGR only needs a few labeled data in training set, while the remaining training set is unlabeled. SSL has been widely used in deep learning tasks (e.g., image and text classification) and has demonstrated excellent performance [43], [44], [45], [46], [47], [48], [49], even surpassing certain supervised learning approaches [50], [51]. However, SSL-based WiFi HGR schemes have not been extensively explored, with only a few pioneer works [26], [27], [33]. For example, UDARF [27] combines consistency regularization (i.e., the same sample, when subjected to different data augmentations, produces similar embeddings) and pseudo-label technique (i.e., the model generates proxy labels for unlabeled data with predicted probabilities above a fixed threshold) to learn more information from the abundant unlabeled data and enhance model performance. RF-URL [33] and AutoFi [26] learn representations of unlabeled data through self-supervised learning and then fine-tune models with the labeled samples. Existing semi-supervised WiFi HGR methods greatly improve the issue of insufficient annotated data. However, they are all based on the assumption of a balanced dataset. Real-world datasets are often imbalanced [52], [53], with Widar3.0 dataset having 22 imbalanced classes. Ignoring the imbalanced nature of the data, such as using a fixed threshold during the training, leads to serious model bias and insufficient generalization performance, especially for minority classes. Therefore, our study explores imbalanced semi-supervised WiFi HGR, aiming to alleviate model bias and enhance model generalization on imbalanced data.

III. Background and Preliminary Analysis

In this section, we provide a brief introduction to the background of the WiFi BVP feature in Section III-A. Following that, we present a preliminary experiment analysis of the dynamic threshold in Section III-B.

A. WiFi Signal

Prevalent WiFi signal features, such as CSI and DFS, are susceptible to environmental influences and lack domain independence [14], [20]. Widar3.0 innovatively introduces domain-independent BVP that shows significant improvements over traditional CSI and DFS in diverse environments. A given gesture sample X comprising T BVP frames, is represented as $X = (x_1, \ldots, x_t, \ldots, x_T)$, where $1 \leq t \leq T$. The dimension of a BVP frame is 20×20 [14]. As shown in Fig. 3, the physical velocity components, V_x and V_y, are linearly mapped into the range of [1,20]. For instance, when $V_x = 0$ and $V_y = 0$, the corresponding coordinates in the BVP frame are $m = 10$ and $n = 10$. Consequently, the non-zero value of the coordinate

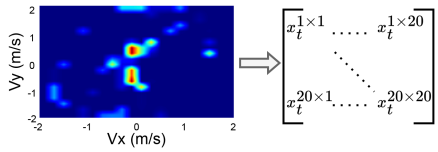


Fig. 3. An example of the frame x_t and its matrix representation. V_x and V_y are physical velocity components [14].

(m, n) in the BVP frame represents the power value (p_v) when the post-mapped physical velocity components V_x and V_y are m and n, respectively. The power values are small $(p_v \in [0, 1])$ and are associated with positional relationships within the BVP frames.

Widar3.0 not only introduces the BVP feature but also provides the largest open WiFi gesture dataset, which significantly advances research in WiFi HGR. The open dataset [1] contains 22 types of gestures, such as "Draw-0", "Draw-2", "Draw-5", "Draw-6", "Draw-7", "Slide", "Draw-O (Horizontal)", "Draw-O (Vertical)", "Draw-Zigzag (Horizontal)", "Push&Pull", and so on. Many WiFi HGR works utilize this open dataset, but they often only select a subset of classes (e.g., 6 out of 22 classes) and overlook the imbalanced nature of the dataset. Specifically, the majority classes (e.g., "Draw-O (Horizontal)", "Slide", "Draw-Zigzag (Horizontal)", and "Push&Pull") have more than 4000 samples, while the minority classes (e.g., "Draw-0", "Draw-2", "Draw-5", "Draw-6", "Draw-7", and "Draw-O (Vertical)") have only around 500 samples. Therefore, an obvious imbalance exists in the real-world WiFi data.

To ensure a uniform evaluation of performance, we also utilize the open dataset and leverage the advanced BVP feature, following the state-of-the-art WiFi HGR works.

B. Preliminary Analysis for Dynamic Threshold

The performance of deep learning models on imbalanced datasets is generally poor for minority classes [31], [54], as the models tend to be biased toward majority classes. This bias is further aggravated when SSL approach has a fixed threshold for unlabeled data, since the model tends to be conservative with minority classes and makes them lose many unlabeled samples with correct proxy labels. Consequently, many actual minority classes in testing set can be misclassified as majority classes, resulting in low recall for the minority classes. Generally, to fairly compare the performance of model on unbalanced dataset, the testing set is balanced and the averaged recall is also known as accuracy on balanced testing set [30]. Here, we utilize the popular fixed-threshold semi-supervised framework, FixMatch [55], to evaluate precision and recall on the imbalanced Widar3.0 dataset with 10% labeled data and 90% unlabeled samples. The experimental results are presented in Fig. 4, and it is evident that the minority gesture classes exhibit low recall but high precision. This suggests that the model is risk-averse and tends to avoid predicting the actual minority classes. In other words, for minority gesture classes, only samples with high confidence

[1] Open dataset address: http://tns.thss.tsinghua.edu.cn/widar3.0/

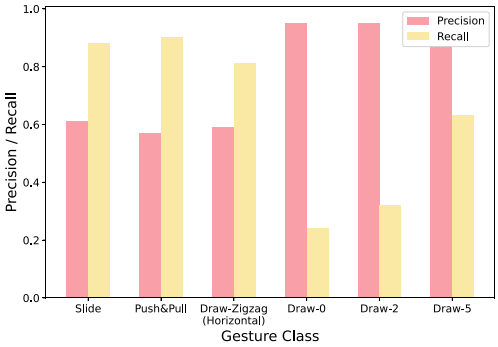


Fig. 4. An example of precision/recall experimental results with a fixed threshold on the imbalanced Widar3.0 dataset ("Slide", "Push&Pull", and "Draw-Zigzag (Horizontal)" are majority gesture classes; "Draw-0", "Draw-2", and "Draw-5" are minority gesture classes).

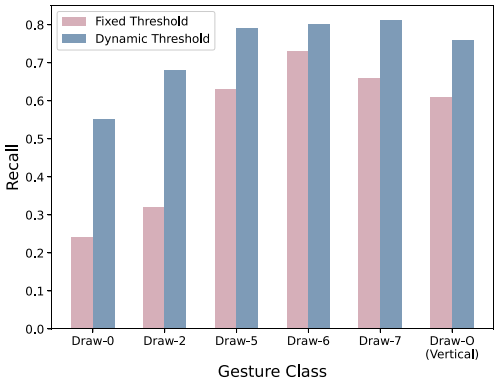


Fig. 5. Recall comparisons of minority classes with fixed and dynamic thresholds.

are predicted, while many samples are misclassified as majority classes. On the contrary, majority classes demonstrate high recall since their predicted samples far outnumber the actual corresponding class data, due to the model's bias.

To alleviate the bias, we propose using dynamic thresholds instead of a fixed one. Based on the aforementioned analysis, lowering thresholds for minority classes can introduce more unlabeled samples and mitigate the bias, leading to an improvement in overall accuracy. In our approach, we dynamically adjust the thresholds of each class during the training process. To validate the effectiveness of our method, we conducted preliminary experiments on the Widar3.0 dataset using 10% labeled data. Fig. 5 illustrates the recall of six minority classes. It can be observed that our DT mechanism significantly improves the recall of the minority classes compared to the fixed threshold approach.

IV. Technical Details of the Proposed Method

Our approach aims to tackle the issue of imbalanced WiFi gesture recognition, where a small portion of the data has labels, a larger portion remains unlabeled, and the class distribution is imbalanced. Our framework, depicted in Fig. 6, facilitates joint learning by combining both labeled and unlabeled data. DA_1 and DA_2 are proposed adaptive data augmentations (see Section IV-A). $\mathcal{F}(\theta)$ is proposed spatial-temporal attention model and θ is the model parameters (see Section IV-B). Both labeled and unlabeled data are processed by the shared STA model $\mathcal{F}(\cdot)$, yielding corresponding gesture probabilities g. Given an unlabeled sample, two augmented unlabeled samples are obtained via the proposed data augmentations DA_1 and DA_2, passing through two branches to generate corresponding gesture probabilities. The labeled data follows normal supervised learning with the standard cross-entropy loss function (denoted as CE). Our method maximizes the potential of unlabeled data to acquire additional information, including two-part losses during the training process. On one hand, we generate pseudo-labels (if exceed the threshold) from one branch to calculate CE loss with another branch. On the other hand, for the rest of the unlabeled data which does not exceed the threshold, we maintain consistency regularization by calculating similarity loss between gesture probabilities of the two branches. To adapt and fully mine the imbalanced characteristics of data, we propose DT mechanism (see Section IV-C) instead of a fixed value during the training process.

In the following sections, the module details of Fig. 6 are introduced. We begin by introducing the proposed adaptive data augmentations, as well as the structure of the STA network. Next, we present the DT mechanism for generating pseudo-labels. Subsequently, we delve into the details of the loss function employed in our framework. Finally, we describe several effective training strategies that were implemented during the training process.

A. Adaptive Data Augmentation

Currently, there lacks systematic exploration of data augmentation methods for BVP, such as in the case of Widar3.0 where data augmentations are not utilized. Unlike non-structural image data, the BVP is devoid of color transformations, and some WiFi HGR works just simply add Gaussian noises into the data [26]. As mentioned in Section III-A, the feature values in BVP can be considered as structural triplets (m, n, p_v), where m and n are positional coordinates, and p_v is corresponding power value. Therefore, if data augmentations, such as image geometric distortions and coordinate transformations, completely alter these triplets, they can severely disrupt the data distribution. To align with data augmentation-driven SSL and enhance model generalization, our approach proposes four adaptive data augmentation techniques for BVP: Random Frame Erasing (RFE), Random Frame Mask (RFM), Random Frame Permutation (RFP), and Random Frame Splicing (RFS). RFE and RFM modify specific portions of the data to introduce adversarial examples, thereby improving model robustness while ensuring minimal alteration of the data distribution. Inspired by time series augmentations in Natural Language Processing (NLP) [56], RFP and RFS consider a frame as a word and apply data augmentations accordingly.

Random frame erasing (RFE): For the BVP frame x_t, we randomly erase certain features and replace them with zeros. The width and height of erasing region are W_e and H_e respectively,

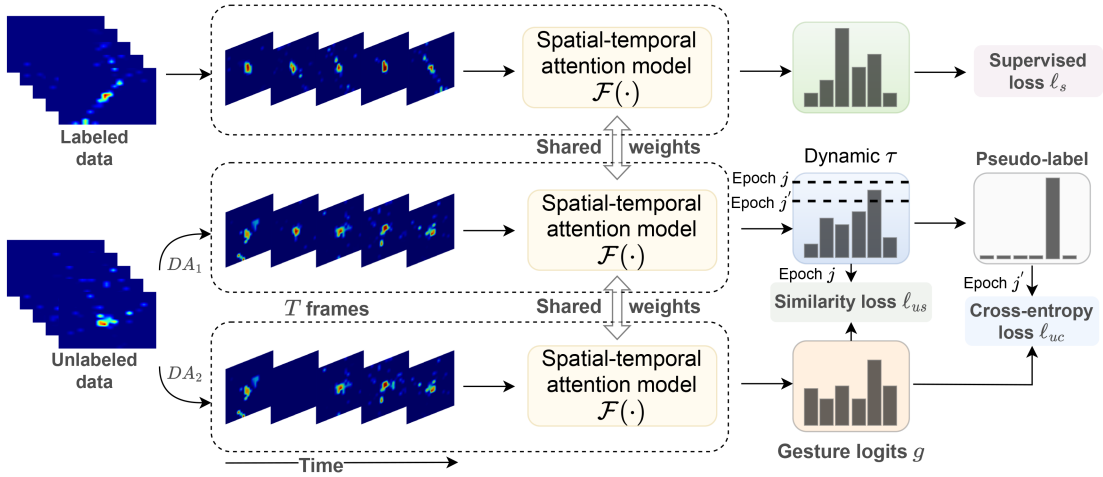


Fig. 6. Illustration of our semi-supervised WiFi HRG framework.

and we denote $r_e = \frac{W_e \times H_e}{20 \times 20}$ as the area ratio of erasing region. In our study, we set $0.05 \leq r_e \leq 0.2$. RFE initializes the point position (m, n), so the position of erasing region is $(m,\ n,\ m + H_e,\ n + W_e)$.

Random frame mask (RFM): In the context of all BVP frames $X = (x_1, \ldots, x_T)$, we randomly pick up a frame and mask it with zeros. For example, we mask the frame x_t, and the RFM can be defined as:

$$\mathsf{RFM}(x_1,\ \ldots,\ x_t,\ \ldots,\ x_T) = (x_1,\ \ldots,\ \mathbf{0}_t,\ \ldots,\ x_T). \quad (1)$$

Random frame permutation (RFP): The proposed scheme involves randomly permuting two frames, namely $x_{t'}$ and x_t, within the set of BVP frames X, where $1 \leq t' < t \leq T$. The mathematical definition of RFP is:

$$\begin{aligned}&\mathsf{RFP}(x_1,\ \ldots,\ x_{t'-1},\ x_{t'},\ \ldots,\ x_{t-1},\ x_t,\ \ldots,\ x_T)\\ &= (x_1,\ \ldots,\ x_{t'-1},\ x_t,\ \ldots,\ x_{t-1}, x_{t'},\ \ldots,\ x_T). \quad (2)\end{aligned}$$

Random frame splicing (RFS): Motivated by combining a sentence with another style, we employ RFS on X, which can be defined as:

$$\begin{aligned}&\mathsf{RFS}(x_1,\ \ldots,\ x_{t-1},\ x_t,\ x_{t+1},\ \ldots,\ x_T)\\ &= (x_t,\ x_{t+1},\ \ldots,\ x_T,\ x_1,\ \ldots,\ x_{t-1}). \quad (3)\end{aligned}$$

In the training stage, the proposed data augmentations undergo with probability p, the probability of them being kept unchanged is $1 - p$. Our study sets p to be 0.5. Fig. 7 illustrates an example showing these four augmentations on an original sample with 5 frames. The RFE randomly erases a portion of the 4th frame, while the RFM applies to the 2-nd frame by masking it with zeros. The RFP and RFS operate on the time dimension. For instance, RFP permutes the 3-rd and 5th frames in Fig. 7, while RFS splices the last two frames before the first frame. Our findings highlight the effectiveness of the proposed adaptive data augmentation methods in enhancing WiFi HGR.

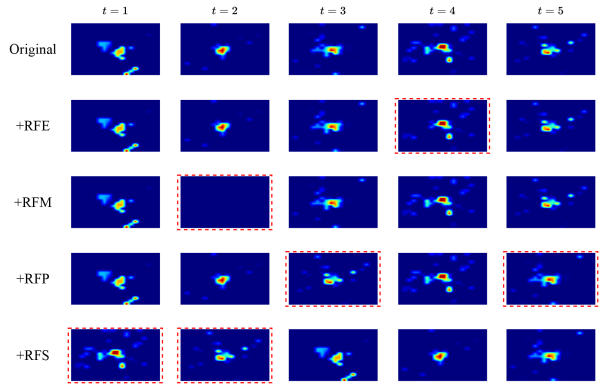


Fig. 7. Data augmentation visualization. The differences between augmented and original samples are in red dotted box.

B. STA Network Structure

The input to the model is the BVP feature, which comprises multiple temporal frames, as depicted in Fig. 1(b). In order to learn the BVP data, Widar3.0 utilizes a CNN-RNN architecture. Specifically, the CNN component learns spatial features from each frame, while the RNN component captures the temporal sequence of the frames. However, each BVP frame contains large areas of trivial blue regions, and the importance of different frames varies. To address this issue, we propose a high-level semantic model that incorporates a spatial-temporal attention mechanism. On one hand, our model utilizes self-attention [40], [57] mechanism to focus on spatial salient features. On the other hand, considering the varying importance of different frames in the time series, we employ a temporal attention mechanism to dynamically learn important scores for all frames.

Attention mechanisms help models to highlight important content of data, which benefits various deep learning applications [58], [59], [60], [61], [62]. To effectively learns salient spatial features and important temporal frames, our approach proposes the spatial-temporal dual-attention model $\mathcal{F}(\theta)$. As

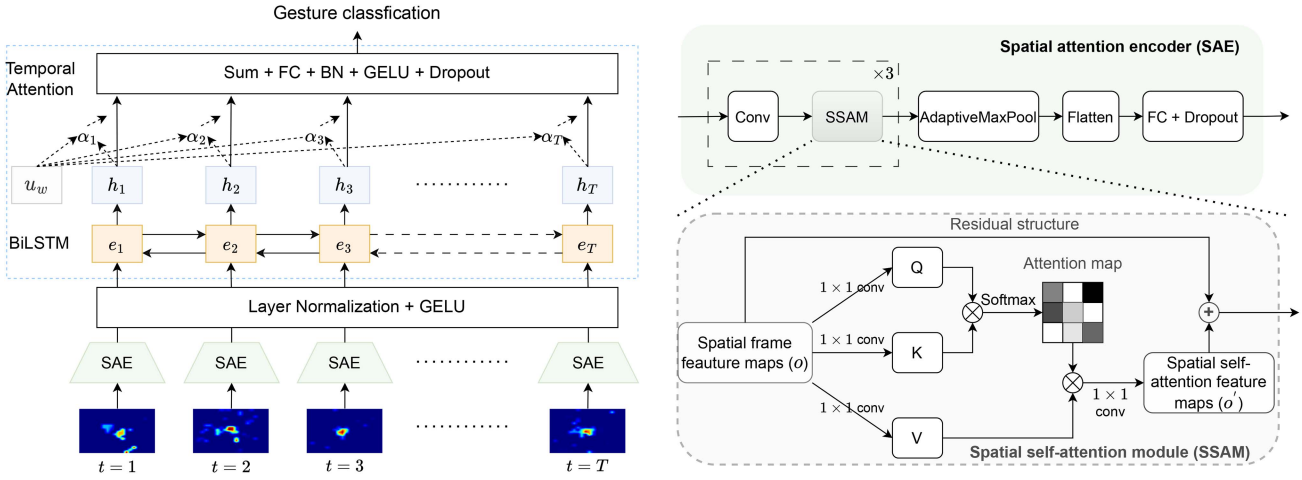


Fig. 8. The sketch of spatial-temporal attention structure. The SAE is described in the upper right, while the SSAM is presented in the lower right.

illustrated in Fig. 8, the proposed model $\mathcal{F}(\theta)$ consists of a spatial-attention encoder (abbreviated as SAE) and a temporal-attention bidirectional long short-term memory (BiLSTM). The SAE captures salient spatial features from each frame x_t, while the temporal-attention BiLSTM (abbreviated as TAB) dynamically assigns different importance scores to the T frames. The dual-attention HGR model can be defined as:

$$E = \sigma(\mathsf{LN}(\mathsf{SAE}(X))), \tag{4}$$

$$r = \sigma(\mathsf{BN}(\mathsf{FC}(\mathsf{TAB}(E)))), \tag{5}$$

$$g = \mathsf{Softmax}(\mathsf{FC}(r)), \tag{6}$$

where σ is the activation function GELU [63], LN is layer normalization [64], FC is fully-connected layer, and BN is batch normalization [65]. The embedding of each frame x_t is denoted as e_t, where $E = (e_1, e_2, \ldots, e_T)$ is the set of frame embeddings. The learned representation of the gesture is denoted as r, and the resulting gesture probability g is obtained through a fully-connected classification layer. In the following, we will describe the core SAE and TAB of the proposed dual-attention model in detail.

The SAE in our model is designed to prioritize salient spatial features in each frame x_t, i.e., non-zero values. This is achieved through a spatial self-attention module (abbreviated as SSAM) that dynamically learns an attention map for the spatial frame features. The input to the SSAM is denoted as o, and the SSAM can be represented as:

$$Q = \phi_1(o),\ K = \phi_2(o),\ V = \phi_3(o), \tag{7}$$

$$AM = \mathsf{Softmax}(Q^T K), \tag{8}$$

$$o' = \phi_4(AM * V) + o, \tag{9}$$

where $\phi_{1,2,3,4}(\cdot)$ are 1×1 convolution [66], AM is the learned attention map, and SSAM also contains residual structure [67] (9). The SAE consists of L stacked SSAMs, where each SSAM is connected to a convolutional layer. The SAE for frame x_t can be defined as:

$$v_t^i = \mathsf{SSAM}(\mathsf{conv}(v_t^{i-1})),\quad 1 \le i \le L \text{ and } v_t^0 = x_t \tag{10}$$

$$e'_t = \mathsf{FC}(\mathsf{AMP}(v_t^L)), \tag{11}$$

where AMP is adaptive max-pool, conv is the convolutional layer. E' is the output of SAE, where $E' = (e'_1, e'_2, \ldots, e'_T)$. In our work, the L is set to be 3.

The TAB in our model is designed to capture temporal correlations and assign important scores to pivotal frame embeddings in E. The TAB utilizes a BiLSTM with attention mechanism to calculate the weight α_t for each embedding e_t. The TAB can be expressed as:

$$H = \mathsf{BiLSTM}(E), \tag{12}$$

$$u_t = \tanh(W_w h_t), \tag{13}$$

$$\alpha_t = \frac{\exp(u_t^T u_w)}{\sum_t \exp(u_t^T u_w)},\ s = \sum_t \alpha_t h_t, \tag{14}$$

where $H = (h_1, \ldots, h_t, \ldots, h_T)$ are the hidden states of BiLSTM, and W_w and u_w are learnable parameters. The output s of TAB is calculated by weighting and summing the hidden states based on their attention weights, emphasizing important frames for a more informative representation. The experimental findings demonstrate that our proposed STA aids in enhancing model generalization and learning more discriminative representations.

C. Dynamic Pseudo-Label Threshold

As discussed and explained in Section III-B, the dynamic pseudo-label threshold mechanism helps mitigate the impact of class-imbalance bias on the model. During the training process of SSL, when the predicted probability of an unlabeled sample exceeds a certain threshold, we can assign a pseudo-label to that data and incorporate it into the supervised learning phase. Unlike a fixed threshold applied to all unlabeled data, our proposed approach sets different thresholds for various classes and dynamically adjusts them during the training process. In the

following section, we provide a detailed explanation of the DT mechanism.

Let us assume that there are Z classes, and we need to calculate the dynamic threshold for each class, denoted as τ_z $(1 \leq z \leq Z)$. Our basic idea is to assign class-dependent thresholds by encoding pseudo-label distributions and determine the dynamic threshold τ_z for each class based on the proportion of pseudo-labels and the sorted probability values. To be specific, in the jth training epoch, given the model parameters θ and Z lists (termed as C_z, $1 \leq z \leq Z$), we calculate the τ_z by three steps. First, the model predicts the unlabeled data X_u and obtain the probability $g_u = F(X_u; \theta)$, and the corresponding predicted label is $\hat{g_u} = \arg\max(g_u)$. Second, adding the predicted probability in the list $C_{\hat{g_u}}$, which can be defined as follows:

$$C_{\hat{g_u}} \leftarrow Append(\max(g_u)), \tag{15}$$

where $Append$ is the operation of adding elements to list. After predicting all unlabeled data and adding their probability values into corresponding list C_z, we sort C_z in descending order, which can be defined as follows:

$$C_z \leftarrow sort(C_z), \quad 1 \leq z \leq Z. \tag{16}$$

Finally, we determine the class-dependent thresholds for minority classes based on encoding class-proportion distribution. For example, when the majority class has a proportion λ of samples that exceed the threshold among all the predicted samples for that class, the dynamic thresholds for the other classes are calculated as $C_z[\lambda \times len(C_z)]$, where $len(C_z)$ represents the number of unlabeled samples predicted as class z. Here, as the labels of unlabeled samples are unknown, we select the class with the highest number of samples in labeled data as reference majority class, denoted as z_{most}. We assign a user-defined high threshold (e.g., $\tau_{z_{most}} = 0.95$) to this majority class. Assuming that there are $l_{z_{most}}$ samples whose probability values exceed the threshold $\tau_{z_{most}}$ for this majority class. For the other classes, we determine their thresholds τ_z as follows:

$$\lambda = \frac{l_{z_{most}}}{len(C_{z_{most}})}. \tag{17}$$

$$C_{z_{most}}[l_{z_{most}}] \geq \tau_{z_{most}}, C_{z_{most}}[l_{z_{most}} + 1] < \tau_{z_{most}}, \tag{18}$$

$$\tau_z = C_z[\lambda \times len(C_z)], 1 \leq z \leq Z \text{ and } z \neq z_{most}. \tag{19}$$

The τ_z may be too small (e.g., less than 0.5) for minority classes during training, so we define a hyper-parameter lower boundary $\tau_{\min}$, namely $\tau_z = \max(\tau_z, \tau_{\min})$. After using the above three steps to obtain dynamic threshold τ_z for class z, we can calculate the pseudo-label loss (see Section IV-D) and optimize the model parameters θ. In the next training epoch $j+1$, the threshold τ_z can be updated dynamically based on the same three-step process.

D. Loss Function

The total loss contains labeled and unlabeled data losses. The labeled data loss we adopt the standard CE loss. For the unlabeled data, we calculate loss from two aspects: pseudo-labeling loss for exceeding threshold and similarity loss for not exceeding. Next, we present the loss functions in detail.

Labeled data loss: Given the labeled batch data $\mathcal{X}_l = \{(X_l^{b_l}, Y_l^{b_l}), b_l = 1, \ldots, B_l\}$ during the training stage, where $X_l^{b_l}$ represents a labeled BVP sample and $Y_l^{b_l}$ represents its corresponding target, the labeled data loss (ℓ_s) is defined as:

$$g_l^{b_l} = \mathcal{F}(X_l^{b_l}; \theta), \tag{20}$$

$$\ell_s = \frac{1}{B_l} \sum_{b_l=1}^{B_l} \mathsf{CE}(Y_l^{b_l}, g_l^{b_l}), \tag{21}$$

where $g_l^{b_l}$ is the gesture probability of b_lth sample in a batch, and B_l is the batch size of labeled data.

Unlabeled data loss: The unlabeled batch data $\mathcal{X}_u = \{X_u^{b_u}\}$ contains $B_u = \mu B_l$ samples during the training stage, where $b_u = \{1, \ldots, \mu B_l\}$ and μ determines the relative bath size of $\mathcal{X}_l$ and $\mathcal{X}_u$. As shown in Fig. 6, by applying DA_1 and DA_2, the augmented samples are obtained as $X_u^{b_u^1}$ and $X_u^{b_u^2}$, respectively. Our approach incorporates two unlabeled data losses. First, if the maximum gesture probability of $X_u^{b_u^1}$ exceeds the corresponding dynamic threshold, its pseudo-label is generated and used to compute the cross-entropy loss (ℓ_{uc}) along with the gesture probability of $X_u^{b_u^2}$. Second, for the remaining data that does not exceed the threshold, we calculate the unsupervised similarity loss (ℓ_{us}) using their respective gesture probabilities, since $X_u^{b_u^1}$ and $X_u^{b_u^2}$ originate from the same sample and should be similar. The unlabeled data loss can be defined as:

$$g_u^{b_u^1} = \mathcal{F}(X_u^{b_u^1}; \theta), \quad g_u^{b_u^2} = \mathcal{F}(X_u^{b_u^2}; \theta), \tag{22}$$

$$\hat{g}_u^{b_u^1} = \arg\max(g_u^{b_u^1}), \tag{23}$$

$$\ell_{uc} = \frac{1}{B_u} \sum_{b_u=1}^{B_u} \mathbb{1}(\max(g_u^{b_u^1}) \geq \tau_{\hat{g}_u^{b_u^1}}) \, \mathsf{CE}(\hat{g}_u^{b_u^1}, g_u^{b_u^2}), \tag{24}$$

$$\mathsf{SIM}(g_u^{b_u^1}, g_u^{b_u^2}) = 2 - 2 \cdot \frac{< g_u^{b_u^1}, g_u^{b_u^2} >}{\|g_u^{b_u^1}\|_2 \cdot \|g_u^{b_u^2}\|_2}, \tag{25}$$

$$\ell_{us} = \frac{1}{B_u} \sum_{b_u=1}^{B_u} \mathbb{1}(\max(g_u^{b_u^1}) < \tau_{\hat{g}_u^{b_u^1}}) \, \mathsf{SIM}(g_u^{b_u^1}, g_u^{b_u^2}), \tag{26}$$

where $g_u^{b_u^1}$ and $g_u^{b_u^2}$ are gesture probabilities of $X_u^{b_u^1}$ and $X_u^{b_u^2}$, respectively. In summary, the total loss is expressed as:

$$\ell_{total} = \ell_s + \ell_{uc} + \ell_{us}. \tag{27}$$

E. Training Strategy

To enhance the generalization performance of the model, we employ an effective training strategy, Post-Balanced Sampling (PBS), throughout the training process.

Typically, re-balanced sampling is a common strategy for handling imbalanced data in supervised learning. However, directly

Algorithm 1: Model Training.

Input : Labeled data D_l, unlabeled data D_u, total training epochs $epoch_1$, post-balanced sampling training epochs $epoch_2$, iteration steps (IS) for each epoch, unlabeled data ratio μ, number of classes Z, user-defined threshold for the most majority class $\tau_{z_{most}}$.
Output: Model parameters θ.

Employ the STA network $\mathcal{F}$ with initial model parameters θ;
Initialize thresholds for all classes with $\tau_z = \tau_{z_{most}}$ $(1 \leq z \leq Z)$;
for $j = 1$ *to* $epoch_1 - epoch_2$ **do**
  for *iter* $= 1$ *to* IS **do**
    Random sample labeled batch data $\mathcal{X}_l = \{(X_l^{b_l}, Y_l^{b_l}), b_l = 1, \ldots, B_l\}$ from D_l;
    Random sample unlabeled batch data $\mathcal{X}_u = \{(X_u^{b_u}), b_u = 1, \ldots, \mu B_l\}$ from D_u;
    Employ the proposed data augmentations on $\mathcal{X}_l$ and $\mathcal{X}_u$;
    Calculate the total loss ℓ_{total} by Eq. (27);
    $\theta \leftarrow$ Gradient Optimization(θ, ℓ_{total}) *// Optimize model parameters θ via gradient* ;
  end
  for $z = 1$ *to* Z **do**
    Calculate and update the dynamic threshold τ_z by Eq. (19);
  end
end
for $j = epoch_1 - epoch_2$ *to* $epoch_1$ **do**
  for *iter* $= 1$ *to* IS **do**
    Balanced sample labeled batch data $\mathcal{X}_l = \{(X_l^{b_l}, Y_l^{b_l}), b_l = 1, \ldots, B_l\}$ from D_l;
    Repeat Line 6 $\sim$ 9;
  end
  Repeat Line 11 $\sim$ 13;
end
return Model parameters θ;

using re-balanced sampling for labeled data often leads to heavy overfitting to minority classes and can make the optimization process challenging [31], [54], [68]. In our study, we use a simple yet effective training strategy: we train the model with random sampling first and then fine-tune it using balanced sampling for a few epochs in later stages of training. Specifically, due to the unknown labels in the unlabeled data, we only use random sampling during training stage. However, for the labeled data in later training stages, each batch size B_l contains the same number of samples (e.g., 4 samples) for each class. The model trained in the random sampling stage already contains a significant amount of data information and serves as a good initialization for the later stage of re-balanced sampling. Moreover, a small learning rate is used during the later stage of re-balanced sampling training. This training strategy enables the model to learn more balanced information, paying more attention to minority classes, while avoiding moving the model parameters too far from their original random sampling initialization. Experimental results demonstrate that the PBS strategy improves model accuracy by large margins.

In addition to the PBS, some common training techniques, such as cosine warm-up learning rate [69], [70] and Exponential Moving Average (EMA) [71], are used to enhance model generalization. These techniques have demonstrated their effectiveness in various deep learning tasks [72], [73]. The total model training epochs are denoted as $epoch_1$, and PBS stage contains $epoch_2$ epochs. The pseudo-code for our model training is outlined in Algorithm 1.

TABLE I
CATEGORY QUANTITY STATISTICS OF DATASET

Sample Size	≥ 5000	(3500, 5000]	(1000,3500]	≤ 1000	Total classes
Class Numbers	1	5	5	11	22
Class	{20}	{17, 13, 19, 21, 0}	{11, 16, 15, 18, 12}	{14, 1 $\sim$ 10}	{0 $\sim$ 21}

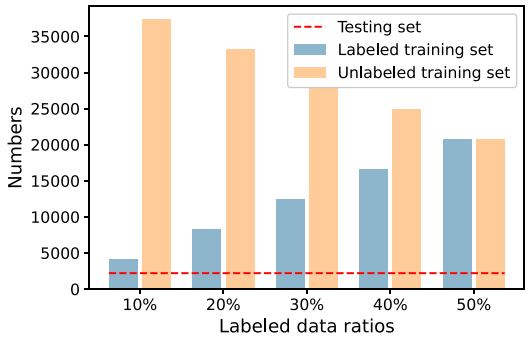


Fig. 9. The descriptions of the training and testing sets.

V. EXPERIMENTATION & EVALUATION

In this Section, we comprehensively evaluate our proposed approach on the Widar3.0 dataset. First, we describe the dataset and experimental setup in Section V-A. Second, we introduce the baselines in Section V-B. Third, overall performance comparisons with the current schemes are presented in Section V-C. Next, Section V-D gives the ablation study. Finally, Section V-E describes the parameter experiments.

A. Dataset and Setup

We evaluate the performance of our scheme on the Widar3.0 dataset, which is currently the largest open gesture dataset available. Widar3.0 contains 22 gesture classes [2] and 43,652 BVP samples. The Widar3.0 is class-imbalanced, and we calculate the sample sizes for each class, as shown in Table I. There are 11 classes (minority classes) that are less than 1,000 samples and 6 classes that are more than 3,500 samples. To explore the influence of different levels of labeled data on the model's performance and its generalization capabilities, we adopt an incremental strategy during the training process, selecting labeled data proportions of 10%, 20%, 30%, 40% and 50%, while the rest of the data remains unlabeled. In the inference stage, the testing set has 2,200 samples, and each class has 100 samples. Detailed descriptions of the training and testing sets are provided in Fig. 9.

Our deep learning framework utilizes PyTorch on the powerful RTX3090 machine. To optimize performance, we configure the labeled data batch size (B_l) to 32. The training process consists of $epoch_1 = 96$ epochs, and $epoch_2$ is set to be 16. We

[2] 22 classes = {0: Clap, 1: Draw-0, 2: Draw-1, 3: Draw-2, 4: Draw-3, 5: Draw-4, 6: Draw-5, 7: Draw-6, 8: Draw-7, 9: Draw-8, 10: Draw-9, 11: Draw-N (Horizontal), 12: Draw-N (Vertical), 13: Draw-O (Horizontal), 14: Draw-O (Vertical), 15: Draw-Rectangle (Horizontal), 16: Draw-Triangle (Horizontal), 17: Draw-Zigzag (Horizontal), 18: Draw-Zigzag (Vertical), 19: Push&Pull, 20: Slide, 21: Sweep}

employ the Adam optimizer [74] with a weight decay of $1e$-4. For learning rate, we set it to $5e$-4 with cosine warm-up over 8 epochs. To prevent overfitting, we apply a dropout rate [75] of 0.2.

B. Baselines

Our work represents a groundbreaking advancement as the first imbalanced semi-supervised WiFi HGR. To provide all-round insights into its efficacy, we have conducted thorough comparative analyses against the state-of-the-art supervised [14], [25] and semi-supervised [26], [27], [33] schemes. Furthermore, we compare various prevalent SSL framework [44], [45], [46], [47], [48], [49], [55], [76], typically tailored for image and textual data. To ensure fair comparisons, we adapt their methodologies to the WiFi dataset. Below, we briefly outline these baselines:

- **WIDAR** [14] denotes the original Widar3.0.
- **CAWGR** [25] proposes to learn cross-domain gestures by using adversarial learning.
- **UDARF** [27] proposes a semi-supervised WiFi HGR by amalgamating consistency regularization and fixed pseudo-label threshold techniques.
- **AutoFi** [26] and **RF-URL** [33] are self-supervised WiFi HGR approaches which can achieve semi-supervision by fine-tuning with labeled data.
- **FixMatch** [55] applies two different data augmentation methods to produce augmented samples for unlabeled data. Subsequently, one augmented sample generates a pseudo-label for another based on a fixed threshold.
- **DS ^{3} L** [44] proposes an open-set SSL approach, where unlabeled samples encompass unseen classes within labeled data. To prevent unseen classes from destroying model performance, DS^3L weakens unlabeled data with unseen classes while strengthening the labeled data.
- **FlexMatch** [45] introduces a dynamic adjustment method for pseudo-label thresholds based on curriculum learning. According to the learning status of the model, the flexible thresholds are adjusted class-wise in each iteration.
- **DRw** [46] proposes a reweighted long-tailed distribution for SSL. Drawing inspiration from the concept of effective numbers [68] in supervised learning, DRw dynamically designs the effective numbers for unlabeled data across diverse iterations.
- **UDAL** [76] integrates the concept of distribution alignment into logit adjustment [77], devising a novel loss function to address imbalanced SSL.
- **FreeMatch** [47] proposes an adaptive approach to calibrate confidence thresholds, estimating global and local class thresholds through exponential moving averages of unlabeled data.
- **SoftMatch** [48] introduces a strategy for balancing the quantity and quality of pseudo-labels, efficiently leveraging unlabeled data. SoftMatch formulates a truncated Gaussian function to assign sample weights.
- **TCBC** [49] employs a twice correction approach to handle model and pseudo-label biases. Initially estimating the class distribution of training samples to rectify the model's learned posterior probabilities, TCBC subsequently refines pseudo-label biases during training by estimating class biases under the current parameters.

TABLE II
ACCURACY COMPARISONS WITH CURRENT SCHEMES

Schemes	Labeled ratios				
	10%	20%	30%	40%	50%
Supervised					
WIDAR [14]	0.569	0.658	0.712	0.732	0.764
WIDAR [14]+STA	0.589	0.681	0.733	0.764	0.786
CAWGR [25]	0.587	0.686	0.732	0.765	0.782
Semi-supervised					
AutoFi [26]	0.703	0.746	0.779	0.809	0.821
RF-URL [33]	0.716	0.760	0.791	0.823	0.843
UDARF [27]	0.728	0.776	0.805	0.836	0.850
FixMatch [55]	0.735	0.781	0.819	0.843	0.857
DS^{3}L [44]	0.721	0.766	0.807	0.833	0.841
FlexMatch [45]	0.759	0.814	0.834	0.849	0.858
DRw [46]	0.785	0.828	0.845	0.859	0.870
UDAL [76]	0.787	0.830	0.848	0.860	0.872
FreeMatch [47]	0.798	0.836	0.853	0.862	0.875
SoftMatch [48]	0.801	0.835	0.852	0.861	0.877
TCBC [49]	0.803	0.839	0.854	0.865	0.878
Ours	**0.818**	**0.850**	**0.869**	**0.877**	**0.884**

In summary, the works [14], [25] are supervised baselines. Additionally, we employ the proposed STA networks to replace the original network model of WIDAR [14] as another supervised method (denoted as WIDAR+STA). For the semi-supervised baselines, the works [27], [55] are based on a fixed threshold, while the works [45], [47], [48] design dynamic pseudo-label thresholds. DS ^{3} L [44] does not assign pseudo-labels for unlabeled data. Furthermore, the works [46], [49], [76] modify loss functions by adjusting distribution for imbalanced SSL. Notably, these works [14], [25], [26], [27], [33] selectively employ a portion of data and classes from open-source Widar3.0 dataset, such as only 6 classes. In contrast, our study incorporates all available BVP samples and classes, ensuring a more inclusive analysis. Throughout the training phase, these supervised schemes only leverage labeled data, utilizing supervised learning with a focus on a certain percentage (e.g., 50%) of labeled examples. Baseline Experimental Parameters primarily follow their original implementations. Since we reproduce baselines on WiFi datasets, parameters like batch size, training epochs, and weight decay are configured according to our experimental setup. For baseline-specific parameters, we select them according to the original papers.

C. Overall Performance

Table II presents the overall performance comparisons with the above mentioned baselines. We can see that our approach

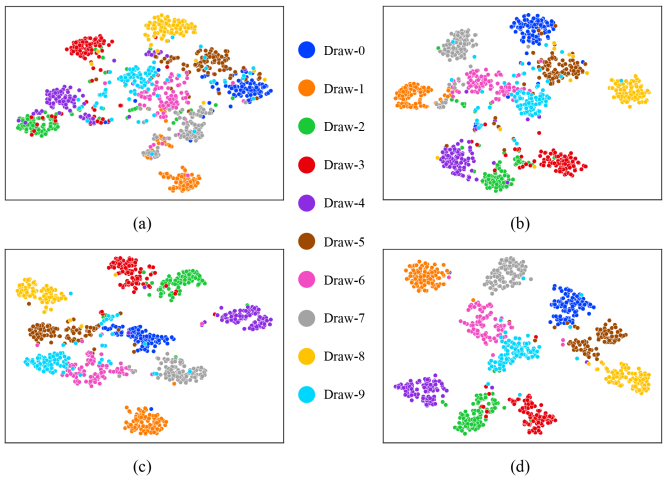


Fig. 10. T-SNE visualization. (a) supervised WIDAR with 10% labeling; (b) supervised WIDAR with 30% labeling; (c) ours with 10% labeling; (d) ours with 30% labeling.

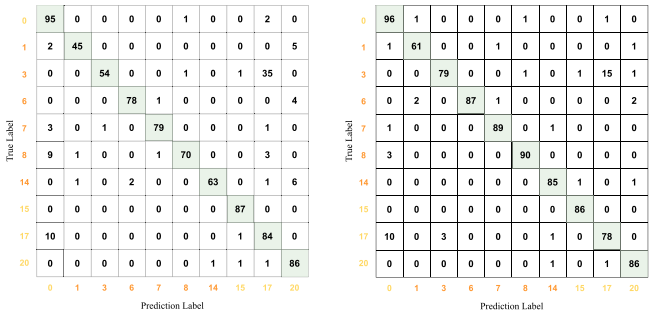


Fig. 11. We select 10 classes to present the confusion matrix with 20% labeled data. The $\{1, 3, 6, 7, 14\}$ are minority classes and the $\{0, 15, 17, 20\}$ are majority classes. The whole 22-classes confusion matrix can be referred at our open code link.

achieves better accuracy than current schemes on different labeled data.

Compared with the supervised schemes, our proposed approach significantly improves model accuracy by approximately $20\% \sim 25\%$. This result demonstrates that SSL can effectively leverage unlabeled data to capture the underlying distribution and improve model generalization performance. For instance, when utilizing 10% labeled data on the Widar3.0 dataset, our approach achieves an impressive accuracy of 81.8%, outperforming WIDAR [14] and CAWGR [25] by approximately 24.9%, and 23.1%, respectively. To further evaluate the performance of our approach, we employ t-SNE [78] to visualize the high-dimensional semantic spaces, using 10% and 30% labeled data. We select 10 classes (1: Draw-0, 2: Draw-1, 3: Draw-2, 4: Draw-3, 5: Draw-4, 6: Draw-5, 7: Draw-6, 8: Draw-7, 9: Draw-8, 10: Draw-9) from the testing set. Fig. 10 depicts the visualization results of WIDAR [14] and our approach. On comparing the inter-class semantic spaces, we observe significant improvements with our approach. For example, when only using 10% labeled data, WIDAR exhibits disorder and poor separation among the 10 classes. In contrast, our proposed scheme achieves clearer inter-class separation. Similarly, with 30% labeled data, WIDAR exhibits unclear boundaries and mis-classifications for classes such as "Draw-6", "Draw-9", "Draw-5", and "Draw-7". However, our approach achieves clearer boundaries and closer intra-class semantic grouping with 30% labeled data. Furthermore, from Table II, we can observe that our SSL models with $10\% \sim 40\%$ labeled data even outperform supervised learning with 50% labeled data in terms of accuracy. This observation is also supported by the t-SNE visualizations, as the semantic space of our 10% labeled data (Fig. 10(c)) is superior to that of supervised learning with 30% labeled data (Fig. 10(b)). Specifically, the distances between different classes are wider, and the number of mis-classified samples is reduced. This indicates that SSL can achieve superior performance using less labeled data, thereby reducing the manual annotation overhead and enabling effective performance in challenging scenarios with limited labeled data.

Compared to existing semi-supervised schemes, our methodology significantly enhances model accuracy across different labeled data scenarios. For example, in contrast to semi-supervised HGR schemes [26], [27], [33], our approach improves accuracy by approximately $3\% \sim 8\%$. By leveraging WiFi data within prevailing SSL frameworks [44], [45], [46], [47], [48], [49], [55], [76], our approach also outperforms them by approximately $0.6\% \sim 1.5\%$. The works [26], [27], [33], [44], [55] tend to overlook the imbalanced nature of the data, such as FixMatch [44] and UDARF [27] employ a fixed pseudo-label threshold. In contrast, our proposed unbalanced processing method effectively improves model accuracy. For example, our scheme achieves improvements of 8.3%, 6.9%, 5.0%, 3.5%, and 2.7% compared to FixMatch [55], with $10\% \sim 50\%$ labeled data. Furthermore, the works [45], [46], [47], [48], [49], [76] devise corresponding modules to cope with imbalanced SSL, such as dynamic thresholds, thus achieving superior model accuracy than [26], [27], [33], [44], [55]. Our approach not only introduces dynamic thresholds by encoding class-wise pseudo-label distribution but also designs effective PBS module to handle imbalanced WiFi data, yielding superior performance to these imbalanced SSL frameworks [45], [46], [47], [48], [49], [76]. Additionally, all semi-supervised schemes demonstrate better accuracy with an increase in labeled data. Our performance advantages over existing methods slightly decrease as the labeled data increases, and this observation emphasizes the significant influence of data on the model's performance.

To visually represent the performance of our solution, we present confusion matrices with 20% labeled data and compare them with FixMatch. The results, as shown in Fig. 11, demonstrate substantial improvements in the accuracy of minority classes. For instance, in FixMatch, only 45 samples of the minority class "Draw-0" (1-st class) are correctly classified. Similarly, the minority class "Draw-2" (3-rd class) has only 54 samples classified correctly, with 35 samples misclassified as "Draw-Zigzag (Horizontal)" (17th class). Due to the similarity

TABLE III
MODEL ACCURACY UNDER DIFFERENT IMBALANCED RATIOS

Schemes	N_1=3500				N_1=1500			
	10	20	50	100	10	20	50	100
FixMatch [55]	0.851	0.783	0.714	0.607	0.799	0.751	0.683	0.535
DRw [46]	0.865	0.806	0.758	0.650	0.811	0.773	0.702	0.579
FreeMatch [47]	0.872	0.814	0.769	0.658	0.827	0.781	0.715	0.588
Ours	0.887	0.831	0.782	0.679	0.841	0.794	0.736	0.607

between "Draw-2" and "Draw-Zigzag (Horizontal)" gestures, and the larger number of samples for "Draw-Zigzag (Horizontal)" (approximately 8 times more), the model in FixMatch is biased towards the 17th class, resulting in low accuracy for the "Draw-2" gesture. However, our approach greatly improves this situation through the proposed DT mechanism and PBS training strategy. For instance, we increase the number of correctly classified "Draw-2" gestures to 79, a 25 improvement compared to FixMatch. Similar improvements are observed for other minority classes in Fig. 11. For example, the correct samples of the 1-st class "Draw-0" increases from 45 to 61, the 14th class "Draw-O (Vertical)" increases from 63 to 85, the 8th class "Draw-7" increases from 70 to 90, the 7th class "Draw-7" increases from 79 to 89, and the 6th class "Draw-7" increases from 78 to 87. It is noted that in our approach, a few majority classes, such as the 17th and 15th classes, have fewer correct samples compared to FixMatch, reducing from 84 to 78 and 87 to 86, respectively. This situation is normal in imbalanced learning [30], as the model needs to focus more on minority classes and allocate less attention to majority classes. The other majority classes either remain unchanged (e.g., class 20) or experience a slight increase (e.g., class 0). While we sacrifice a small amount of performance in certain majority classes, the overall model performance has been significantly improved. In summary, our proposed approach enhances model generalization and substantially improves the accuracy of minority classes.

Table II shows the semi-supervised results on original imbalance ratio of the Widar3.0 dataset [14]. To investigate the impact of varying imbalance ratios, we manually adjusted the data distribution following established imbalanced works [45], [47], [48]. Consider Z classes with a total labeled sample size N, where each class has N_z labeled samples (i.e., $\sum_{z=1}^{Z} N_z = N$). Without loss of generality, we assume classes are sorted in descending order of sample size, namely $N_1 \geq N_2 \geq \cdots \geq N_Z$. We use parameter γ to represent the imbalance ratio, where $\gamma = \frac{N_1}{N_Z}$. Specifically, given N_1 and γ, we set $N_z = N_1 \cdot \gamma^{-\frac{z-1}{Z-1}}$. In our experiments, we tested configurations with $N_1 = 3500, \gamma = \{10, 20, 50, 100\}$ and $N_1 = 1500, \gamma = \{10, 20, 50, 100\}$, with results shown in Table III. The results demonstrate that model performance degrades as the imbalance ratio increases, consistent with findings in prior works [45], [46], [47], [48], [49], [53]. For instance, when $N_1 = 3500$, accuracy decreases from 0.887 to 0.831, 0.782, and 0.679 as γ increases from 10 to 20, 50, and 100, respectively. Additionally, reduced labeled data leads to performance degradation. For example, when N_1 decreases from 3500 to 1500 at $\gamma = 10$, accuracy drops from 0.887 to 0.841. These results confirm that both labeled data quantity and imbalance ratio affect model performance.

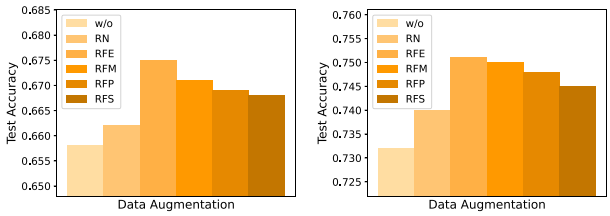
TABLE IV
ABLATION EXPERIMENT RESULTS OF OUR APPROACH ON DIFFERENT LABELED DATA (RN: RANDOM NOISE; ADA: ADAPTIVE DATA AUGMENTATION; DT: DYNAMIC THRESHOLD; PBS: POST-BALANCED SAMPLING)

Module	Labeled ratios				
	10%	20%	30%	40%	50%
+RN	0.704	0.752	0.786	0.812	0.823
+ADA	0.722	0.767	0.801	0.824	0.836
+STA	0.735	0.781	0.819	0.843	0.857
+DT	0.791	0.823	0.841	0.856	0.868
+PBS	0.818	0.850	0.869	0.877	0.884

D. Ablation Experiment

Our approach proposes a series of effective techniques to enhance model generalization on imbalanced dataset, including adaptive data augmentations, STA networks, DT mechanism, and PBS training strategy. Here, we use an ablation study to explore the influences of these modules on model accuracy, incrementally adding them and presenting the results in Table IV. The study demonstrates the effectiveness of these modules in improving model accuracy. We provide detailed descriptions of each module below.

Adaptive data augmentation (ADA): Since our SSL framework relies on data augmentations, it is essential to use appropriate augmentations. Existing WiFi HGR solutions either do not utilize data augmentation or only apply random noise, limiting the model's representation ability. Due to the properties of electromagnetic waves, the CSI and DFS of WiFi signals exhibit significant environmental dependency, prompting Widar3.0 [14] to propose advanced domain-independent BVP features. To eliminate environmental influences, our work employs the publicly available BVP data from Widar3.0 [14]. However, there are no dedicated data augmentation methods for BVP. Unlike images that permit various rotations and color transformations, WiFi constitutes structured complex-valued signals, precluding direct application of image-based augmentation techniques [32], [79]. While some existing works [26], [79], [80] have proposed data augmentation methods for CSI or DFS, no online data augmentation scheme has been developed for BVP. Notably, BVP data differs substantially from CSI and DFS in physical interpretation, rendering existing CSI/DFS augmentation methods inapplicable to BVP. Consequently, we designed adaptive data augmentation methods specifically for BVP. We compare our adaptive data augmentation methods with simple random noise addition, the results are shown in Table IV. It can be observed that an improvement of approximately $1.2\% \sim 1.8\%$ in model accuracy across different labeled data scenarios, such as from 0.704 to 0.722 on 10% labeling. Furthermore, we apply our adaptive data augmentations to supervised WIDAR [14] (only supervised learning with labeled data). The results in Fig. 12



(a) 20% labeling (b) 40% labeling

Fig. 12. WIDAR [14] with our adaptive data augmentations and alongside only supervised learning using labeled data. "RN" means random noise. "w/o" represents no augmentations, namely original WIDAR [14].

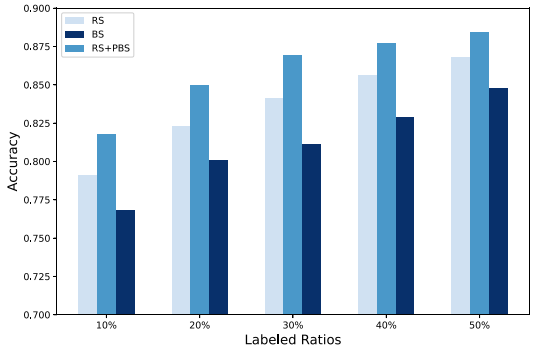


Fig. 13. Comparisons of different sampling strategies (RS: random sampling; BS: balanced sampling).

TABLE V
RECALL OF MINORITY CLASSES WITH DIFFERENT MODULES (10% AND 20% LABELED SAMPLES)

Gesture		Draw-0	Draw-2	Draw-7	Draw-O (Vertical)	Draw-5	Draw-6
10%	Fixed threshold	0.24	0.32	0.66	0.61	0.63	0.73
	DT	0.55	0.68	0.81	0.76	0.79	0.80
	DT+PBS	0.58	0.71	0.86	0.82	0.83	0.86
20%	Fixed threshold	0.45	0.54	0.70	0.63	0.78	0.79
	DT	0.57	0.72	0.83	0.79	0.82	0.85
	DT+PBS	0.61	0.79	0.90	0.85	0.87	0.89

TABLE VI
MODEL ACCURACY WITH UNSUPERVISED SIMILARITY LOSS (26)

Labeled ratios	10%	20%	30%	40%	50%
w/o	0.813	0.847	0.865	0.874	0.880
w	0.818	0.850	0.869	0.877	0.884

"w/o" represents without the similarity loss and "w" represents with it.

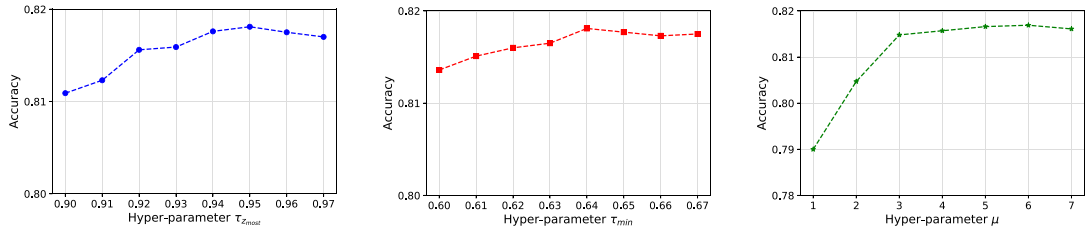
demonstrate that our proposed augmentations outperform random noise, particularly for the RFE and RFM augmentations. In summary, our four adaptive data augmentations significantly enhance model robustness and improve accuracy.

STA networks: Our STA networks enhance the model's ability to learn discriminative representations by incorporating spatial-temporal attention. As shown in Table IV, compared to models without attention mechanisms, STA improves accuracy from 0.836 to 0.857 with 50% labeled samples. This trend continues across labeled data scenarios ranging from 10% to 40% . STA also improves model accuracy in the context of supervised learning, as demonstrated by the results ("WIDAR [14]+STA") in Table II. Notably, STA significantly enhances the performance of WIDAR, increasing accuracy from 0.569 to 0.587 with 10% labeled supervised learning.

Dynamic threshold & post-balanced sampling: The DT mechanism and PBS strategy play crucial roles in addressing imbalanced datasets in semi-supervised WiFi HGR. From Table IV, we observe that the DT improves accuracy by more than 5% and 4% on 10% and 20% labeled data, respectively. Additionally, the PBS strategy enhances model accuracy by approximately 2% across labeled data scenarios ranging from 10% to 50% . Therefore, for semi-supervised WiFi HGR on imbalanced datasets, our proposed methods are vital for leveraging more information and enhancing model representation ability. As discussed in Section III-B, without imbalance operations, minority classes typically exhibit low recall on imbalanced datasets. To evaluate the impact of the DT and PBS modules on recall, we compare the recall of six minority classes. The results in Table V demonstrate that these modules significantly improve recall for minority classes. For instance, the recall of gesture "Draw-0" is improved by 31% with the DT mechanism, and the recall of gesture "Draw-2" is improved by 36% with 10% labeled data. The proposed PBS also improves recall by approximately $3\% \sim 6\%$ for these minority classes. In Section IV-E, we mentioned that directly applying balanced sampling to labeled data during training may lead to overfitting for minority classes, which in turn affects unlabeled data and overall performance. We compare the performance of different sampling methods on labeled data: random sampling (RS), balanced sampling (BS), and our proposed sampling (RS+PBS). The results in Fig. 13 demonstrate that directly using BS leads to lower performance compared to RS, while our proposed sampling strategy achieves the highest model accuracy. Our approach effectively handles imbalanced-class WiFi HGR datasets, improving the performance of minority classes and overall dataset. The comparative evaluations highlight the potential of our approach as a groundbreaking solution for imbalanced semi-supervised WiFi HGR.

Unsupervised similarity loss: In addition to supervised loss and pseudo-label loss, we integrate an unsupervised similarity loss (26) into our total loss function (27). When unlabeled data lacks pseudo-labels, triggering the absence of a pseudo-label loss, the unsupervised similarity loss becomes significant. Unsupervised similarity loss promotes consistency between augmented unlabeled data, thereby facilitating the acquisition of richer representations from unlabeled samples. Comparing the impact of including or omitting the similarity loss, as depicted in Table VI, reveals that integrating the unsupervised similarity loss can enhance model accuracy by approximately $0.3\% \sim 0.5\%$, underscoring its role in augmenting the learning efficacy of unlabeled samples.

Model Computational Costs Analysis: As shown in Table VII, our model size is 13.31 MB with 3.48 M parameters and a computational cost of 1.39 G FLOPs. Since the input BVP dimension is relatively small (20×20), the model's computational



(a) $\tau_{z_{\text{most}}}$ (b) τ_{min} (c) μ

Fig. 14. Hyper-parameter sensitivity.

TABLE VII
MODEL PARAMETERS, SIZE, COMPUTATIONAL FLOPS, AND TRAINING TIME

Model Size	Parameters	FLOPs	Training Batch Time	
			Fixed Threshold	Dynamic Threshold
13.31MB	3.48M	1.39G	0.29s	0.30s

TABLE VIII
LABELED RATIO GRADIENT EXPERIMENTS

Labeled ratios	5%	10%	20%	30%	40%	50%	60%	70%	80%
Accuracy	0.741	0.818	0.850	0.869	0.877	0.884	0.889	0.892	0.894

overhead is not particularly high. Additionally, during training, we compute dynamic thresholds, which introduces some extra training time. As the dynamic threshold computation occurs when gradient propagation is stopped, it does not introduce additional model parameters. Compared to fixed thresholds, dynamic thresholds increase the processing time per batch by 0.01 s, which remains acceptable. In the future, we will explore methods to reduce the number of parameters, such as model compression techniques, to make the models more lightweight and suitable for resource-constrained devices.

More Varying Labeled Ratios: As shown in Table VIII, we expand experiments to include annotation ratios ranging from 5% to 80% . The results reveal that: (1) Model generalization improves with increasing annotation ratios, indicating greater benefits from more labeled data. (2) Performance rises sharply when the annotation ratio increases from 5% to 30% . For example, accuracy improves from 0.741 to 0.818 (+7.7%) when the ratio increases from 5% to 10%, and from 0.818 to 0.850 (+3.2%) when increasing from 10% to 20% . (3) Beyond 40% labeled ratios, performance gains diminish significantly. For instance, accuracy increases by only 0.6% when the ratio rises from 50% to 60%, and by about 1% even at 80% annotation compared to 50% . This suggests that with suitable annotation, our model achieves strong generalization without relying on excessive labeled data.

E. Parameter Study

Our approach involves three hyper-parameters: $\tau_{z_{most}}$, $\tau_{\min}$, and μ. To assess the sensitivity of our proposed scheme to different values of these hyper-parameters, we conducted experiments with a labeled ratio of 10% . The results are presented in Fig. 14.

For $\tau_{z_{most}}$, we tested values ranging from {0.90, 0.91, 0.92, 0.93, 0.94, 0.95, 0.96, 0.97}. Fig. 14(a) demonstrates that $\tau_{z_{most}} = 0.95$ yields the best performance. Regarding $\tau_{\min}$, which sets the minimum value of the dynamic threshold to avoid incorporating samples with excessively low confidence probability, we explored values from {0.60, 0.61, 0.62, 0.63, 0.64, 0.65, 0.66, 0.67}. Fig. 14(b) reveals that $\tau_{\min} = 0.64$ achieves the optimal performance. Although other values of $\tau_{z_{most}}$ and $\tau_{\min}$ lead to slight fluctuations in accuracy, they do not significantly compromise the overall performance, underscoring the robustness of our approach to the selection of these hyper-parameters.

The hyper-parameter μ is dependent on the relative batch size of labeled and unlabeled data. We examined values from {1, 2, 3, 4, 5, 6, 7}. Fig. 14(c) demonstrates that smaller μ values (e.g., $\mu = 1$) result in a significant decrease in model performance, with a decrease of more than 2% in accuracy compared to $\mu \geq 3$. This observation aligns with the findings of FixMatch [55]. In our study, we selected $\mu = 6$ as it achieved the best performance.

VI. DISCUSSION

There are several potential directions to further explore in the field of semi-supervised WiFi HGR.

Unknown gesture class: The current schemes assume that the classes of unlabeled data and the testing set are the same as those in the labeled training set. However, in real-world scenarios, the collected unlabeled training data may contain complex and diverse gestures, including additional unknown classes. Training directly with labeled and unknown-class unlabeled samples can disrupt the data distribution. Moreover, during real-world use, users may perform invalid gestures that are unknown to the model, but the model may still attempt to recognize them using a fixed trained classifier. Therefore, addressing the issue of unknown classes can enhance model robustness and orient to practical scenarios. In the future, we will explore corresponding techniques to handle this problem.

Lightweight model: To learn more discriminative representations from WiFi data, our model employs STA networks, which increases model parameters. However, some mobile terminal devices have limited resources, making it challenging to deploy models with a large number of parameters. This issue is prevalent in current deep learning-based WiFi HGR schemes, as they strive for high accuracy, often resulting in models with numerous parameters. In the future, we will explore methods to

reduce the number of parameters, such as model compression techniques, to make the models more lightweight and suitable for resource-constrained devices.

Multi-source gesture datasets: The existing methodologies operate under the assumption that unlabeled samples originate from a single source dataset. Nonetheless, collected unlabeled data could stem from multi-source datasets, which may introduce notable disparities in data distribution. The divergence in data sources poses challenges in aligning data distributions and significantly impairs the model's generalization capabilities. To enhance the resilience of semi-supervised WiFi HGR in real-world scenarios, future efforts must address the adversarial impact of multi-source datasets, potentially through the adoption of multi-source domain adaptation technologies.

VII. Conclusion

This paper introduces a novel imbalanced semi-supervised WiFi HGR framework featuring STA networks. Rather than employing fixed threshold for unlabeled samples, we design class-independent thresholds for all classes and dynamically adjust them during training process, which significantly alleviates model bias on imbalanced dataset. To learn more discriminative representations of WiFi signals, the proposed STA structure dynamically learns spatial salient features and crucial temporal frames. Furthermore, we present four adaptive data augmentations tailored for WiFi signal data to enhance model generalization performance. Experimental results on Widar3.0 dataset show significant accuracy improvements over existing semi-supervised schemes, highlighting its potential as a powerful imbalanced semi-supervised WiFi HGR solution.

References

[1] Y. Song and R. Davis, "Continuous body and hand gesture recognition for natural human-computer interaction: Extended abstract," in *Proc. Int. Joint Conf. Artif. Intell.*, AAAI Press, 2015, pp. 4212–4216.

[2] H. Liu et al., "Real-time arm gesture recognition in smart home scenarios via millimeter wave sensing," in *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 4, no. 4, pp. 140:1–140:28, 2020.

[3] N. M. Mahmoud, H. Fouad, and A. M. Soliman, "Smart healthcare solutions using the internet of medical things for hand gesture recognition system," *Complex Intell. Syst.*, vol. 7, pp. 1253–1264, 2021.

[4] F. Li and J. Fei, "Gesture recognition algorithm based on image information fusion in virtual reality," *Pers. Ubiquitous Comput.*, vol. 23, no. 3/4, pp. 487–497, 2019.

[5] 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.

[6] G. Gkioxari, R. B. Girshick, P. Dollár, and K. He, "Detecting and recognizing human-object interactions," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8359–8367.

[7] T. Zhang, T. Song, D. Chen, T. Zhang, and J. Zhuang, "WiGrus: A WiFi-based gesture recognition system using software-defined radio," *IEEE Access*, vol. 7, pp. 131102–131113, 2019.

[8] E. Hayashi et al., "RadarNet: Efficient gesture recognition technique utilizing a miniature radar sensor," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2021, pp. 5:1–5:14.

[9] S. Palipana, D. Salami, L. A. Leiva, and S. Sigg, "Pantomime: Mid-air gesture recognition with sparse millimeter-wave radar point clouds," in *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 5, no. 1, pp. 27:1–27:27, 2021.

[10] R. Gao et al., "Towards robust gesture recognition by characterizing the sensing quality of WiFi signals," in *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 6, no. 1, pp. 11:1–11:26, 2022.

[11] Z. Yang, Y. Zhang, G. Chi, and G. Zhang, "Hands-on wireless sensing with Wi-Fi: A tutorial," 2022, *arXiv:2206.09532*.

[12] J. Yang et al., "Deep learning and its applications to WiFi human sensing: A benchmark and A tutorial," 2022, *arXiv:2207.07859*.

[13] J. Zhang, Y. Li, H. Xiong, D. Dou, C. Miao, and D. Zhang, "HandGest: Hierarchical sensing for robust-in-the-air handwriting recognition with commodity WiFi devices," *IEEE Internet of Things J.*, vol. 9, no. 19, pp. 19529–19544, Oct. 2022.

[14] Y. Zhang et al., "Widar3.0: Zero-effort cross-domain gesture recognition with Wi-Fi," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8671–8688, Nov. 2022.

[15] Y. Ma et al., "SignFi: Sign language recognition using WiFi," in *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 2, no. 1, pp. 23:1–23:21, 2018.

[16] J. Yang, H. Zou, Y. Zhou, and L. Xie, "Learning gestures from WiFi: A siamese recurrent convolutional architecture," *IEEE Internet of Things J.*, vol. 6, no. 6, pp. 10763–10772, Dec. 2019.

[17] G. Yin, J. Zhang, G. Shen, and Y. Chen, "FewSense, towards a scalable and cross-domain Wi-Fi sensing system using few-shot learning," *IEEE Trans. Mobile Comput.*, vol. 23, no. 1, pp. 453–468, Jan. 2024.

[18] Y. Liu et al., "UniFi: A unified framework for generalizable gesture recognition with Wi-Fi signals using consistency-guided multi-view networks," in *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 7, no. 4, pp. 168:1–168:29, 2023.

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

[20] Y. Zhang et al., "Zero-effort cross-domain gesture recognition with Wi-Fi," in *Proc. Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2019, pp. 313–325.

[21] Y. Gu et al., "WiGRUNT: WiFi-enabled gesture recognition using dual-attention network," *IEEE Trans. Hum. Mach. Syst.*, vol. 52, no. 4, pp. 736–746, Aug. 2022.

[22] C. Li, M. Liu, and Z. Cao, "WiHF: Gesture and user recognition with WiFi," *IEEE Trans. Mobile Comput.*, vol. 21, no. 2, pp. 757–768, Feb. 2022.

[23] B. Li, W. Cui, W. Wang, L. Zhang, Z. Chen, and M. Wu, "Two-stream convolution augmented transformer for human activity recognition," in *Proc. AAAI Conf. Artif. Intell.*, AAAI Press, 2021, pp. 286–293.

[24] G. Tong, Y. Li, H. Zhang, and N. Xiong, "A fine-grained channel state information-based deep learning system for dynamic gesture recognition," *Inf. Sci.*, vol. 636, 2023, Art. no. 118912.

[25] H. Kang, Q. Zhang, and Q. Huang, "Context-aware wireless-based cross-domain gesture recognition," *IEEE Internet of Things J.*, vol. 8, no. 17, pp. 13503–13515, Sep. 2021.

[26] J. Yang, X. Chen, H. Zou, D. Wang, and L. Xie, "AutoFi: Toward automatic Wi-Fi human sensing via geometric self-supervised learning," *IEEE Internet of Things J.*, vol. 10, no. 8, pp. 7416–7425, Apr. 2023.

[27] B. Zhang, D. Zhang, Y. Li, Y. Hu, and Y. Chen, "Unsupervised domain adaptation for RF-based gesture recognition," *IEEE Internet of Things J.*, vol. 10, no. 23, pp. 21026–21038, Dec. 2023.

[28] X. Yang, Z. Song, I. King, and Z. Xu, "A survey on deep semi-supervised learning," 2021, *arXiv:2103.00550*.

[29] Y. Ouali, C. Hudelot, and M. Tami, "An overview of deep semi-supervised learning," 2020, *arXiv: 2006.05278*.

[30] C. Wei, K. Sohn, C. Mellina, A. L. Yuille, and F. Yang, "CReST: A class-rebalancing self-training framework for imbalanced semi-supervised learning," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10857–10866.

[31] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5375–5384.

[32] T. Li, L. Fan, Y. Yuan, and D. Katabi, "Unsupervised learning for human sensing using radio signals," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2022, pp. 1091–1100.

[33] R. Song et al., "RF-URL: Unsupervised representation learning for RF sensing," in *Proc. Annu. Int. Conf. Mobile Comput. Netw.*, 2022, pp. 282–295.

[34] H. Abdelnasser, M. Youssef, and K. A. Harras, "WiGest: A ubiquitous WiFi-based gesture recognition system," in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 1472–1480.

[35] Q. Pu, S. Gupta, S. Gollakota, and S. N. Patel, "Whole-home gesture recognition using wireless signals," in *Proc. Annu. Int. Conf. Mobile Comput. Netw.*, 2013, pp. 27–38.
[36] H. Li et al., "WiFinger: Talk to your smart devices with finger-grained gesture," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, pp. 250–261.
[37] S. Tan and J. Yang, "WiFinger: Leveraging commodity WiFi for fine-grained finger gesture recognition," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2016, pp. 201–210.
[38] D. Wu et al., "WiTraj: Robust indoor motion tracking with WiFi signals," *IEEE Trans. Mobile Comput.*, vol. 22, no. 5, pp. 3062–3078, May 2023.
[39] L. Sun, S. Sen, D. Koutsonikolas, and K. Kim, "WiDraw: Enabling hands-free drawing in the air on commodity WiFi devices," in *Proc. Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 77–89.
[40] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[41] C. Li, M. Liu, and Z. Cao, "WiHF: Enable user identified gesture recognition with WiFi," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 586–595.
[42] R. Xiao, J. Liu, J. Han, and K. Ren, "OneFi: One-shot recognition for unseen gesture via COTS WiFi," in *Proc. ACM Conf. Embedded Netw. Sensor Syst.*, 2021, pp. 206–219.
[43] M. Zheng, S. You, L. Huang, F. Wang, C. Qian, and C. Xu, "SimMatch: Semi-supervised learning with similarity matching," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 14451–14461.
[44] L. Guo et al., "Safe deep semi-supervised learning for unseen-class unlabeled data," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2020, pp. 3897–3906.
[45] B. Zhang et al., "FlexMatch: Boosting semi-supervised learning with curriculum pseudo labeling," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 18408–18419.
[46] H. Peng, W. Pian, M. Sun, and P. Li, "Dynamic re-weighting for long-tailed semi-supervised learning," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2023, pp. 6453–6463.
[47] Y. Wang et al., "FreeMatch: Self-adaptive thresholding for semi-supervised learning," in *Proc. Int. Conf. Learn. Representations*, 2023.
[48] H. Chen et al., "SoftMatch: Addressing the quantity-quality trade-off in semi-supervised learning," in *Proc. Int. Conf. Learn. Representations*, 2023.
[49] L. Li et al., "Twice class bias correction for imbalanced semi-supervised learning," in *Proc. AAAI Conf. Artif. Intell.*, AAAI Press, 2024, pp. 13563–13571.
[50] Y. Wang et al., "Semi-supervised semantic segmentation using unreliable pseudo-labels," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 4238–4247.
[51] T. Chen et al., "Big self-supervised models are strong semi-supervised learners," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 22243–22255.
[52] Y. Wang, D. Ramanan, and M. Hebert, "Learning to model the tail," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 7029–7039.
[53] L. Guo and Y. Li, "Class-imbalanced semi-supervised learning with adaptive thresholding," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2022, pp. 8082–8094.
[54] K. Cao, C. Wei, A. Gaidon, N. Aréchiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 1565–1576.
[55] K. Sohn et al., "FixMatch: Simplifying semi-supervised learning with consistency and confidence," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 596–608.
[56] J. W. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," in *Proc. Conf. Empir. Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process.*, Association for Computational Linguistics, 2019, pp. 6381–6387.
[57] H. Zhang, I. J. Goodfellow, D. N. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2019, pp. 7354–7363.
[58] S. Yao et al., "SADeepSense: Self-attention deep learning framework for heterogeneous on-device sensors in Internet of Things applications," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 1243–1251.
[59] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based LSTM for aspect-level sentiment classification," in *Proc. Empir. Methods Natural Lang. Process.*, The Association for Computational Linguistics, 2016, pp. 606–615.
[60] P. Zhou et al., "Attention-based bidirectional long short-term memory networks for relation classification," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, The Association for Computer Linguistics, 2016, pp. 207–212.
[61] Q. Feng et al., "DHAN: Encrypted JPEG image retrieval via DCT histograms-based attention networks," *Appl. Soft Comput.*, vol. 133, 2023, Art. no. 109935.
[62] Q. Feng et al., "End-to-end privacy-preserving image retrieval in cloud computing via anti-perturbation attentive token-aware vision transformer," *Inf. Fusion*, vol. 121, 2025, Art. no. 103153.
[63] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016, *arXiv:1606.08415*.
[64] L. J. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.
[65] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
[66] C. Szegedy et al., "Going deeper with convolutions," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.
[67] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[68] Y. Cui, M. Jia, T. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9268–9277.
[69] T. Wolf et al., "Transformers: State-of-the-art natural language processing," in *Proc. Empir. Methods Natural Lang. Process.*, Association for Computational Linguistics, 2020, pp. 38–45.
[70] Q. Feng et al., "EViT: Privacy-preserving image retrieval via encrypted vision transformer in cloud computing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 8, pp. 7467–7483, Aug. 2024.
[71] Z. Cai, A. Ravichandran, S. Maji, C. Fowlkes, Z. Tu, and S. Soatto, "Exponential moving average normalization for self-supervised and semi-supervised learning," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 194–203.
[72] Y. Li, T. Yao, Y. Pan, and T. Mei, "Contextual transformer networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 1489–1500, Feb. 2023.
[73] H. Yin, A. Mallya, A. Vahdat, J. M. Alvarez, J. Kautz, and P. Molchanov, "See through gradients: Image batch recovery via GradInversion," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 16337–16346.
[74] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.
[75] N. Srivastava et al., "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[76] J. Lazarow, K. Sohn, C. Lee, C. -L. Li, Z. Zhang, and T. Pfister, "Unifying distribution alignment as a loss for imbalanced semi-supervised learning," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2023, pp. 5633–5642.
[77] A. K. Menon et al., "Long-tail learning via logit adjustment," in *Proc. Int. Conf. Learn. Representations*, 2021.
[78] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.
[79] W. Hou and C. Wu, "RFBoost: Understanding and boosting deep WiFi sensing via physical data augmentation," in *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 8, no. 2, 2024, Art. no. 58.
[80] J. Zhang et al., "Data augmentation and dense-LSTM for human activity recognition using WiFi signal," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4628–4641, Mar. 2021.

Qihua Feng received the MS degree in computer technology from Jinan University, Guangzhou, China, in 2022. He is currently working toward the PhD degree with the Beijing Institute of Technology, Beijing, China. His research interests include privacy preserving, deep learning, and Internet of Things.

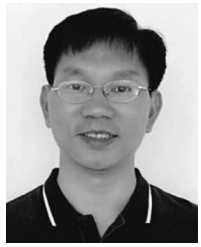
Chunhui Duan received the BS and PhD degrees from the School of Software, Tsinghua University, Beijing, China, in 2013 and 2018, respectively. Previously, she was a postdoctoral research fellow with Tsinghua University. She is currently an associate professor with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing. Her research interests include RFID, Internet of Things, wireless sensing, and mobile computing.

Jiawei Xue received the BS degree from the Beijing Institute of Technology, Beijing. Currently, he is working toward the MS degree with the Beijing Institute of Technology, Beijing, with Internet of Things as his research interest.

Chaozhuo Li received the PhD degree in computer software and theory from the School of Computer Science and Engineering, Beihang University, Beijing, China, in 2020. He is currently an associate researcher with the Beijing University of Posts and Telecommunications. He has published more than 100 papers in conferences such as NeurIPS, AAAI, SIGIR, ICDM, and CIKM. His research interests include data mining and social network analysis.

Feiran Huang received the PhD degree from the School of Computer Science and Engineering, Beihang University, Beijing, China, in 2018. He is currently a professor with the College of Information Science and Technology, Jinan University. He has more than 100 publications appearing in top conferences and journals, such as SIGIR, KDD, WWW, MM, *IEEE Transactions on Image Processing*, *International Journal of Computer Vision*, and *IEEE Transactions on Knowledge and Data Engineering*. He received 5 best paper awards, such as the Best Paper Award Honourable Mention in SIGIR 2023 and Best Paper Award Runner Up in PAKDD 2023. He holds more than 50 US, Chinese, and international granted patents. He is an editorial board member of *IEEE Transactions on Affective Computing*, *ACM Transactions on Recommender Systems*, etc. His research interests include recommender systems, social networks, sentiment analysis, and large language models.

Xi Zhang (Member, IEEE) received the PhD degree in computer science from Tsinghua University. He is a professor with the Beijing University of Posts and Telecommunications, and is also the vice director of the Key Laboratory of Trustworthy Distributed Computing and Service, Ministry of Education, China. He was a visiting scholar with the University of Illinois, Chicago. His research interests include data mining and computer architecture.



Jian Weng (Senior Member, IEEE) received the PhD degree in computer science and engineering from Shanghai Jiao Tong University, Shanghai, China, in 2008. He is currently a professor and the dean of the College of Information Science and Technology, Jinan University, Guangzhou, China. His research interests include public key cryptography, cloud security, and blockchain. He was the PC co-chair or a PC member of more than 30 international conferences. He also serves as an associate editor for the *IEEE Transactions on Vehicular Technology*.

Philip S. Yu (Life Fellow, IEEE) received the PhD degree in electrical engineering from Stanford University, Stanford, California. He is a distinguished professor in computer science with the University of Illinois, Chicago, Illinois and is the Wexler chair in Information Technology. His research interests include Big Data, data mining, data stream, database, and privacy. He was the editor-in-chief for the *IEEE Transactions on Knowledge and Data Engineering* and the *ACM Transactions on Knowledge Discovery from Data*. He was the recipient of ACM SIGKDD 2016 Innovation Award, a Research Contributions Award from the IEEE International Conference on Data Mining (2003), and a Technical Achievement Award from the IEEE Computer Society 2013. He is a fellow of ACM.