XFall: Domain Adaptive Wi-Fi-Based Fall Detection With Cross-Modal Supervision

Guoxuan Chi, *Member, IEEE*, Guidong Zhang, *Member, IEEE*, Xuan Ding, *Member, IEEE*, Qiang Ma, *Member, IEEE*, Zheng Yang, *Fellow, IEEE*, Zhenguo Du, Houfei Xiao, and Zhuang Liu

***Abstract*— Recent years have witnessed an increasing demand for human fall detection systems. Among all existing methods, Wi-Fi-based fall detection has become one of the most promising solutions due to its pervasiveness. However, when applied to a new domain, existing Wi-Fi-based solutions suffer from severe performance degradation caused by low generalizability. In this paper, we propose *XFall*, a domain-adaptive fall detection system based on Wi-Fi. *XFall* overcomes the generalization problem from three aspects. To advance cross-environment sensing, *XFall* exploits an environment-independent feature called speed distribution profile, which is irrelevant to indoor layout and device deployment. To ensure sensitivity across all fall types, an attention-based encoder is designed to extract the general fall representation by associating both the spatial and temporal dimensions of the input. To train a large model with limited amounts of Wi-Fi data, we design a cross-modal learning framework, adopting a pre-trained visual model for supervision during the training process. We implement and evaluate *XFall* on one of the latest commercial wireless products through a year-long deployment in real-world settings. The result shows *XFall* achieves an overall accuracy of 96.8%, with a miss alarm rate of 3.1% and a false alarm rate of 3.3%, outperforming the state-of-the-art solutions in both in-domain and cross-domain evaluation.**

***Index Terms*— Domain adaptation, fall detection, statistical electric field, transformer encoder, cross-modal supervision.**

I. Introduction

HUMAN falls constitute a major public health issue. An estimated 684,000 fatal falls occur each year, making it the second leading cause of unintentional injury death worldwide [1]. During the past decade, extensive research efforts have been devoted to preventing death and injury caused by accidental falls.

Despite the proliferation of fall detection solutions, including vision-based systems, wearable devices, and sensor-based approaches, each of these methods comes with its set of limitations. Vision-based systems [2], [3], [4], for example, are hampered by privacy concerns and the inherent limitations of camera field-of-view (FoV), rendering them ineffective in non-line-of-sight (NLoS) conditions. Wearable devices [5], [6], although innovative, often suffer from user acceptance issues due to their intrusive nature. On the other hand, solutions that rely on advanced radar [7] and acoustic sensors [8] face significant cost barriers, hindering their widespread adoption. This background highlights the urgent need for a fall detection method that is widespread, secure, and non-intrusive, filling the essential gaps present in today's indoor fall detection solutions.

Over the past decades, Wi-Fi infrastructures have been widely deployed. As a pervasive wireless signal filling out our indoor spaces, Wi-Fi has been leveraged to build various types of contactless sensing systems [9], [10], [11], [12], [13]. While initial studies [14], [15], [16] have highlighted the capability of Wi-Fi channel state information (CSI) for fall detection, a significant challenge of these existing approaches is their lack of adaptability to environments beyond those in which they were initially trained. This limitation in cross-domain generalization limits their practical deployment. To build a *one-fits-all* Wi-Fi-based fall detection system, which is able to *train once, use anywhere*, we face three major challenges.

Challenge 1: How to extract the environment-independent feature. Previous Wi-Fi-based fall detection systems leverage either primitive features such as the raw CSI amplitude and phase [14], [17], or the spectrogram feature like doppler frequency spectrogram (DFS) [15] and discrete wavelet transformation (DWT) spectrogram [18] generated from the CSI. Unfortunately, due to the multipath effect, the raw CSI patterns vary significantly across different indoor structures, which obscure the characteristic caused by human activity. Although both DFS and DWT reflect the environment dynamics independent from the indoor structure, they highly rely on the device deployment and user's location [19]. To conclude, none of the above-mentioned features can directly adapt to a different environment.

Challenge 2: How to design a model for learning general human fall representation. Human fall is a complex activity, and different fall types induce specific signal disturbances over time and space. Existing solutions resort to either classic machine learning models like support vector machine (SVM) [15], or deep neural networks such as multi-layer perception (MLP) and convolutional neural network (CNN) [7]. However, neither of them explores the association between

Manuscript received 14 November 2023; revised 18 March 2024; accepted 19 April 2024. Date of publication 14 June 2024; date of current version 21 August 2024. This work was supported in part by NSFC under Grant 62372265. *(Corresponding author: Zheng Yang.)*

Guoxuan Chi, Guidong Zhang, Xuan Ding, Qiang Ma, and Zheng Yang are with the School of Software and BNRist, Tsinghua University, Beijing 100084, China (e-mail: chiguoxuan@gmail.com; zhanggd18@gmail.com; dingxuan@tsinghua.edu.cn; tsinghuamq@gmail.com; hmilyyz@gmail.com).

Zhenguo Du, Houfei Xiao, and Zhuang Liu are with Huawei Technologies Company Ltd., Shenzhen 518129, China (e-mail: zhenguo.du@huawei.com; xiaohoufei@huawei.com; liuzhuang2@huawei.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSAC.2024.3413997.

Digital Object Identifier 10.1109/JSAC.2024.3413997

the spatial and the temporal dimensions of the input feature. For example, some people may hold on the furniture or lean against the wall as they fall, which might not induce significant signal fluctuations. And these types of human fall activity are usually difficult to detect by existing models.

Challenge 3: How to train a large model with limited Wi-Fi data. To achieve system generalizability, it is imperative to employ a deep neural network model with tens of millions of parameters. Such models necessitate extensive labeled Wi-Fi sensing data for training. Unfortunately, the collection and labeling of Wi-Fi datasets are notably labor-intensive. Particularly, on-line labeling[1] is essential for Wi-Fi sensing data, as these data types are not interpretable by human observation [20]. Therefore, collecting enough labeled Wi-Fi data for training poses a significant challenge.

To tackle the above challenges, we design and implement *XFall*, a domain-adaptive fall detection system based on Wi-Fi. To derive an environment-independent wireless feature, we dig deeper into the fundamental principles of electromagnetic field propagation. Inspired by statistical electromagnetic field theory, we construct a wireless feature called speed distribution profile (SDP), which can adapt to different environments accommodating differences in indoor structures, device deployment, and user locations. To exploit the general representation of human fall, we design a spatial-temporal-attention-based transformer encoder (STATE), which learns the associations and dependencies across the spatial and temporal dimensions of the input feature, and thus characterizes continuous activity at a higher level of abstraction. To train a large model with the limited amount of labeled data, we design a cross-model unified feature learning (CURL) framework, which adopts the idea of knowledge distillation [21]. With the CURL framework, a vision-based fall detection network is adopted to supervise the training process, which effectively reduces the demand on labeled Wi-Fi data.

We implement *XFall* on a commercial Wi-Fi product, Huawei AX3 Pro, and perform a year-long evaluation across 70 different real-world settings, including diverse indoor layouts, AP deployments, and user locations. We also compare *XFall* with three state-of-the-art Wi-Fi-based fall detection systems, including DeFall [16], TLFall [18], and FallDeFi [15]. The evaluation result shows that *XFall* achieves an average miss alarm rate (MAR) and false alarm rate (FAR) of 3.1% and 3.3% respectively, with an overall accuracy of 96.8%, surpassing DeFall, TLFall, FallDeFi by 14.5%, 22.3%, and 23.1% respectively, thereby demonstrating *XFall*'s superior performance for real-world application.

Our contributions are summarized as follows:

- We propose *XFall*, the first domain-adaptive fall detection system based on Wi-Fi. Our proposed design enables *XFall* to adapt to different environments (with different indoor layouts, AP deployments, and user locations) and keep sensitive to all human fall types, making a promising step towards practical and ubiquitous Wi-Fi sensing.
- Both the environment-independent feature and the spatio-temporal encoder proposed in *XFall* have their unique advantages. They can be directly applied to other types of Wi-Fi sensing applications (e.g., human gesture recognition, gait recognition) to improve their generalizability.
- Our proposed CURL framework effectively reduces the need of labeled Wi-Fi data, which provides few-shot-learning capability to Wi-Fi-based sensing systems.
- We implement and evaluate *XFall* on AX3 Pro, which is a pioneer attempt to build up a real-world wireless sensing application based on the latest commercial wireless product supporting 802.11ax (Wi-Fi 6) standard.

The rest of this paper is organized as follows. We begin by reviewing the related work in Section II, and provide an overview of *XFall* in Section III, followed by detailed design of the speed distribution profile (SDP) in Section IV, the spatio-temperal attention-based encoder (STATE) in Section V, and the cross-modal unified representation learning (CURL) framework in Section VI. Our implementation and evaluation of *XFall* is shown in Section VII, followed by the conclusion in Section VIII.

II. Related Work

This section briefly summarizes the most related works in the following categories.

A. CSI-Based Fall Detection

The utilization of Channel State Information (CSI) has been a focal point in Wi-Fi sensing applications, particularly for fall detection, showcasing the potential of commercial devices in capturing intricate movement patterns. WiFall [14] achieves fall detection based on CSI amplitude. On this basis, RTFall [17] extracts both CSI amplitude and phase differences between different antennas to detect fall events. FallDeFi [15] adopts time-frequency analysis on CSI amplitude and presents a sequential forward selection algorithm to single out the features that are resilient to environmental changes. TLFall [18] extracts DWT profiles and applies transfer learning to reduce the impact of different environments. DeFall [16] is a pioneering work that extracts environment-independent speed features for fall detection without intensive training. FallViewer [22] proposes a series of CSI calibration algorithms to reduce the component of environments and improve the impact of fall activities. The above-mentioned works acquire several statistical characteristics from the features and discriminate fall activities by classical machine learning algorithms, such as SVM classifiers and random forest algorithms. Despite their contributions, the quest for a system that adapts seamlessly across different environments without the need for reconfiguration underscores our research motivation. In contrast to the above methods, our system, *XFall*, introduces SDP, a feature agnostic to domain variations such as environment and deployment diversities. This innovation enables *XFall* to robustly withstand the challenges posed by varying conditions. Moreover, by employing the STATE model, *XFall* adeptly identifies fall-specific high-level features, thereby enhancing the accuracy of fall detection.

[1]On-line labeling entails labeling data concurrently with its collection.

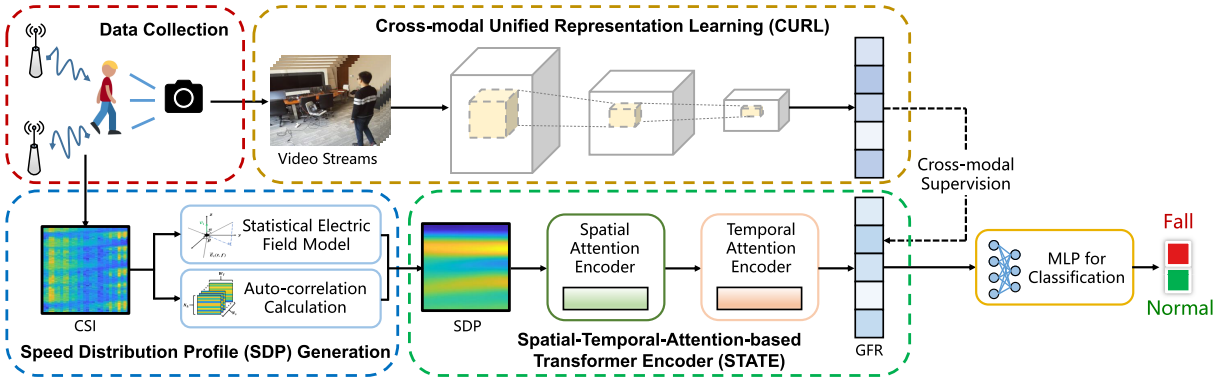


Fig. 1. System architecture of *XFall*.

B. Fall Detection Based on Other Modalities

Beyond CSI, fall detection research has explored a variety of sensing technologies, such as vision-based, wearable device-based, and dedicated radar-based approaches. Vision-based methods capture images and extract human movements by different kinds of cameras, including RGB cameras, event cameras, and depth cameras [2], [3], [4], [23]. Wearable device-based methods usually leverage the Inertial Measurement Unit (IMU), including accelerometers, gyroscopes, and inclinometers to recognize fall events [5], [6], [24], [25], [26]. In terms of dedicated radar-based approaches, RF-Fall [7] extracts spatial heatmaps by Frequency Modulated Continuous Wave (FMCW) radars and adopts CNN to capture high-level spatial features to recognize fall activities. Acoustic-FADE [8] utilizes a circular microphone array to recognition fall activities. Inspired by statistical wireless sensing, VeCare [27] introduces statistical acoustic sensing for the first time, which can support speed estimation using audio signals. Our research, *XFall*, distinctly fills the technical gap by introducing a ubiquitous, non-invasive detection system that leverages pervasive Wi-Fi signals. This approach capitalizes on the omnipresence of Wi-Fi to realize fall detection, setting our system apart from previous endeavors by harnessing a signal modality that is ubiquitous and seamlessly integrated into everyday environments, thereby offering a practical solution for fall detection.

C. CSI-Based Activity Recognition

The scope of CSI's application extends beyond fall detection, encompassing a broad array of activities like gesture and gait recognition, and passive human tracking. E-eyes [9] is a pioneer work to use commercial Wi-Fi signals to distinguish in-place activities. WiFiU [28], Indotrack [29] and Widar [30] achieve gait recognition and passive human tracking based on DFS profiles, respectively. WiSpeed [31] is the first to introduce statistical EM approaches for Wi-Fi-based speed estimation via the ACF of CSI power, which is then extended to the ACF of CSI in GaitWay [32]. Widar3.0 [19] extracts body-coordinate velocity profile, which captures body-coordinate velocity profiles (BVP) of human gestures at the lower signal level. EI [33], and CrossSense [34] incorporate adversarial networks and transfer learning models to realize human activity sensing, respectively. SLNet [35] proposes the first complex-valued neural network tailored for RF signals by integrating spectrum analysis with deep learning model in a novel co-design, which demonstrates versatility across a wide range of wireless sensing tasks. Recently proposed RF-Diffusion [13] successfully enhances the accuracy and robustness of existing wireless sensing systems through data augmentation. Nevertheless, applying these insights to fall detection demands an approach that overcomes the environmental and situational variability inherent to real-world applications. Our proposed *XFall* distinguishes itself by extracting domain-independent features, enabling precise detection without the need for domain-specific training, thus broadening the horizon for fall detection technologies.

III. System Overview

XFall is a human fall detection system based on off-the-shelf Wi-Fi devices. As shown in Figure 1, a pair of Wi-Fi devices are deployed around the monitoring area to capture CSI streams. Concurrently, the system gathers visual data from an RGB camera, facilitating the cross-modal training process.

XFall consists of three key components: the speed distribution profile (SDP) generation, the spatial-temporal-attention-based transformer encoder (STATE) and the cross-modal unified representation learning (CURL) framework. Once receiving CSI series of fall activities and normal activities, *XFall* performs the SDP generation algorithm. *XFall* establishes the statistical electric field model and calculates the auto-correlation function of CSI data to generate the SDP, which contains environment-independent human motion information. The SDP series are input to the STATE, which extracts high-level spatial features at each moment and temporal features throughout the whole series. The application of attention modules prompts the network to focus on the spatio-temporal features corresponding to fall activities, which advances classification accuracy. After the STATE, the general fall representation (GFR) can be extracted from the SDP. The CURL framework is proposed to perform cross-modal learning during the training process, in which a pre-trained visual model is introduced to supervise STATE.

It's worth mentioning that the CURL framework only works during the training process. In the testing stage, *XFall* works with Wi-Fi data only and doesn't require any visual input.

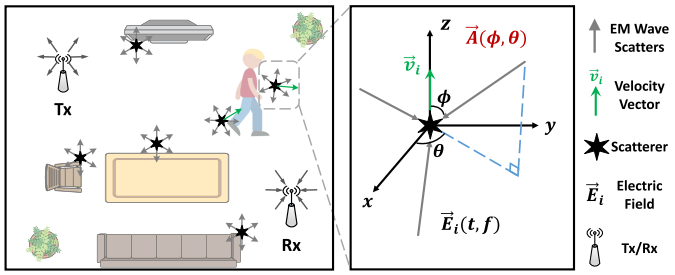


Fig. 2. Illustration of a rich scattering environment, where the transmitted waves are diffused by the surroundings before arriving at the receiver.

IV. Speed Distribution Profile

As demonstrated in previous researches [36], the unique velocity distribution across all human body parts is one of the most effective indicators for activity recognition. Among all wireless features (e.g., attenuation, AoA, ToF), the DFS profile contains the environment's velocity information and is widely used for activity recognition [37]. However, the DFS profile is environment-dependent, which means it is highly related to the transceiver deployment and the user's location and orientation, preventing it from cross-domain pervasive applications [19].

In this section, we exploit a domain-independent feature called **speed distribution profile (SDP)**, which adapts to different indoor layouts, user locations, and device deployment. Fundamental theory of the statistical electric field model (Section IV-A) is first introduced. On this basis, the formulation of electric field from CSI (Section IV-B) is proposed, followed by the calculation of SDP (Section IV-C).

A. Statistical Electric Field Model

As shown in the left part of Figure 2, most indoor spaces can be modeled as a rich scattering environment, where the scatterers (e.g., the human body parts, furniture, and building walls) are assumed to be diffusive and can reflect the wireless signals toward all directions [31]. The continuous electric magnetic (EM) waves emitted by the transmitter (Tx) are finally received by the receiver (Rx) after a series of scattering processes.

Typically, in an indoor environment, the EM wave can be fully characterized by its electric field. Therefore, let $\vec{E}_{\mathrm{Rx}}(t,f)$ denote the electric field received at time t with a frequency f. To analyze the spatial dynamics of the environment, we decompose $\vec{E}_{\mathrm{Rx}}(t,f)$ as follows:

$$\vec{E}_{\mathrm{Rx}}(t,f) = \sum_{\alpha\in\Omega_{\mathrm{s}(t)}} \vec{E}_\alpha(t,f) + \sum_{\beta\in\Omega_{\mathrm{d}(t)}} \vec{E}_\beta(t,f), \quad (1)$$

where $\Omega_{\mathrm{s}(t)}$ and $\Omega_{\mathrm{d}(t)}$ are the set of static and dynamic scatterers respectively.

To fully understand $\vec{E}_{\mathrm{Rx}}(t,f)$, we dive into each component $\vec{E}_\alpha(t,f)$, which denotes the EM wave scattered by the i-th scatterer. We notice that $\vec{E}_\alpha(t,f)$ can be interpreted as an integral of the incident (a.k.a. incoming) EM waves over all directions, as shown in the right part of the Figure 2. For each incident wave with elevation ϕ and azimuth θ, define its angular spectrum as $\vec{A}(\phi,\theta)$, and its wave-number vector as $\vec{k} = -\frac{2\pi f}{c}(\sin(\phi)\cos(\theta), \sin(\phi)\sin(\theta), \cos(\phi))^{\mathsf{T}}$, where c is the speed of light. Suppose the speed of the α-th scatterer is $\vec{v}_\alpha$, based on the Maxwell's equations [38], the $\vec{E}_\alpha(t,f)$ can be represented as follows:

$$\vec{E}_\alpha(t,f) = \int_0^{2\pi}\int_0^{\pi} \vec{A}(\phi,\theta)\exp\left(-j\vec{k}\cdot\vec{v}_\alpha t\right)\sin(\phi)\mathrm{d}\phi\mathrm{d}\theta \quad (2)$$

where the z-axis is aligned with the moving direction of the scatterer α. With Equation 2, we successfully bridge the gap between the speed v_α and the received electric field.

However, with no prior knowledge of the indoor structure, the radio propagation process is generally difficult to analyze since neither the location of each scatterer nor the value of $\vec{A}(\phi,\theta)$ can be determined. Therefore, instead of constructing a deterministic model, we treat the indoor space as a reverberation cavity [38] and build up a statistical model of EM fields. Specifically, $\vec{E}_\alpha(t,f)$ is assumed to be a superposition of a large number of independent waves from uniformly distributed directions with random phases and polarizations. Therefore, the following two conclusions can be derived: 1) The angular spectrum $\vec{A}(\phi,\theta)$ is a circularly symmetric Gaussian random variable [39]; 2) $\forall\alpha,\beta\in\Omega_{\mathrm{d}}, \forall t_1,t_2$, the incoming EM wave component $\vec{E}_\alpha(t_1,f)$ and $\vec{E}_\beta(t_2,f)$ are uncorrelated.

On this basis, $\vec{E}_\alpha(t_1,f)$ can be approximated as a stationary random process, with an auto-correlation function (ACF) in the following form:

$$\rho_{\vec{E}_\alpha}(\tau,f) = \frac{\langle\vec{E}_\alpha(t,f), \vec{E}_\alpha(t+\tau,f)\rangle}{\sqrt{\|\vec{E}_\alpha(t,f)\|^2\|\vec{E}_\alpha(t+\tau,f)\|^2}}, \quad (3)$$

where $\langle\cdot\rangle$ is the inner product operator, and τ is the time lag. With detailed proof in previous studies [40], we have:

$$\langle\vec{E}_\alpha(t,f), \vec{E}_\alpha(t+\tau,f)\rangle = E_\alpha^2(f)\frac{\sin(kv_\alpha\tau)}{kv_\alpha\tau}, \quad (4)$$

where we define $E_\alpha^2(f)$ as the power of $\vec{E}_\alpha(t,f)$. Therefore, by aggregating all the independent scatter components, we get the ACF of the received electric field:

$$\rho_{\vec{E}_{\mathrm{Rx}}}(\tau,f) = \frac{1}{\sum_{\beta\in\Omega_{\mathrm{d}}}E_\beta^2(f)}\sum_{\alpha\in\Omega_{\mathrm{d}}}E_\alpha^2(f)\frac{\sin(kv_\alpha\tau)}{kv_\alpha\tau}. \quad (5)$$

With Equation 5, we successfully bridge the gap between the received electric field and the environmental dynamics (i.e., the speed v_α of each scatter). In other words, by analyzing the ACF of the received electric field, we can infer the speed distribution of the surrounding environment.

B. From CSI to Scatter Speed

One significant challenge remains to get the received electric field from the commercial Wi-Fi device. Let $X(t,f)$ and $Y(t,f)$ be the transmitted and the received signal waves of a subcarrier with frequency f at time t. Then, we notice that $Y(t,f) = \|\vec{E}_{\mathrm{Rx}}\|$. Based on the definition of CSI, $H(t,f) = Y(t,f)/X(t,f)$. Similar to Equation 1, we can decompose the CSI as follows:

$$H(t,f) = \sum_{\alpha\in\Omega_{\mathrm{s}(t)}} H_\alpha(t,f) + \sum_{\beta\in\Omega_{\mathrm{d}(t)}} H_\beta(t,f), \quad (6)$$

where $H_\alpha(t,f)$ is the CSI component corresponding to the α-th scatterer. Notice that $X(t,f)$ is the long training field predefined in the 802.11 protocol, and thus can be treated as a constant. Therefore, the CSI amplitude $|H(t,f)|$ is proportional to the norm of the electric field:

$$|H(t,f)| \propto \|\vec{E}_{\mathrm{Rx}}(t,f)\|, |H_\alpha(t,f)| \propto \|\vec{E}_\alpha(t,f)\|. \quad (7)$$

According to the Equation 4 and Equation 5, both the covariance function $\langle \vec{E}_{\mathrm{Rx}}(t,f), \vec{E}_{\mathrm{Rx}}(t+\tau,f)\rangle$ and auto-correlation function $\rho_{\vec{E}_{\mathrm{Rx}}}(\tau,f)$ degenerate to a scalar independent of the direction of the electric field vector. Therefore, we can directly substitute Equation 7 into Equation 4 and Equation 5, and construct the statistical relationship between CSI and the speed distribution in the environment:

$$\begin{aligned}\rho_H(\tau,f) &= \frac{\mathrm{cov}(H(t,f),H(t+\tau,f))}{\mathrm{cov}(H(t,f),H(t,f))} \\ &= \frac{1}{\sum_{\beta\in\Omega_\mathrm{d}}\sigma_\beta^2(f)}\sum_{\alpha\in\Omega_\mathrm{d}}\sigma_\alpha^2(f)\frac{\sin(kv_\alpha\tau)}{kv_\alpha\tau} \\ &= \frac{\sum_{\alpha\in\Omega_\mathrm{d}}\sigma_\alpha^2(f)\ \mathrm{sinc}(kv_\alpha\tau)}{\sum_{\beta\in\Omega_\mathrm{d}}\sigma_\beta^2(f)} \\ &= \sum_{\alpha\in\Omega_\mathrm{d}} w_\alpha \mathrm{sinc}(kv_\alpha\tau), \end{aligned} \quad (8)$$

where the function $\mathrm{cov}(\cdot,\cdot)$ indicates the covariance of two random variables, and $\sigma_\alpha(t,f)$ refers to the standard deviation of the amplitude of the α-th CSI component $|H_\alpha(t,f)|$. The Equation 8 shows that the ACF of the CSI can be considered as a weighted-sum of the scatterers' speed v_α after a nonlinear filter $\mathrm{sinc}(\cdot)$.

C. Generate the Speed Distribution Profile

In practice, the CSI data reported by the Wi-Fi NIC forms into a 2-D complex matrix $\boldsymbol{H} \in \mathbb{C}^{N_T\times N_S}$, where N_T and N_S are the number of packets and the number of subcarriers respectively. Therein $H(t_i,f_j)$ indicates a complex-valued sample of the CSI from the i-th packet and the j-th subcarrier.

Typically, SDP $\boldsymbol{S}$ is determined by three key parameters: the number of CSI samples N_T, the time lag resolution Δ_T, and the number of lag samples N_Δ. Each SDP is generated from a 3-D ACF tensor $\boldsymbol{\rho} \in \mathbb{R}^{N_\Delta\times W_T\times N_S}$, where each element $\rho(n,i,j)$ can be represented as follows:

$$\begin{aligned}\rho(n,i,j) &= \rho_H(n\Delta_T, f_j) \\ &= \frac{|H(t_i,f_j)H^*(t_i-n\Delta_T,f_j)|}{|H(t_i,f_j)||H(t_i-n\Delta_T,f_j)|}. \end{aligned} \quad (9)$$

On this basis, by merging all the subcarriers and performing probabilistic normalization to each column of the ACF tensor, the SDP $\boldsymbol{S} \in \mathbb{R}^{N_\Delta\times W_T}$ is generated, with each element $S(n,i)$ as follows:

$$S(n,i) = \frac{\sum_{j=1}^{N_S}\rho(n,i,j)/N_S}{\sum_{n=1}^{N_\Delta}\rho(n,i,j)}. \quad (10)$$

For ease of understanding, define a random process $u(\tau,v) = \sum_{\alpha\in\Omega_\mathrm{d}} w_\alpha \mathrm{sinc}(kv_\alpha\tau)$ based on Equation 8. Then, as shown in Figure 3, each column in $\boldsymbol{S}$ represents a discrete sample of u at different τ, which contains the speed distribution of the environment. Therefore, the SDP can be treated as a matrix, with two dimensions containing the spatial and temporal information of the environmental dynamics, respectively.

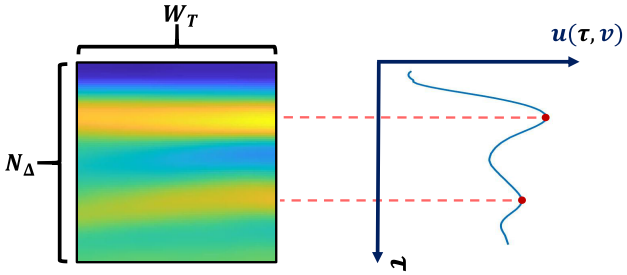


Fig. 3. Illustration of a speed distribution profile (SDP), with two dimensions containing the spatial and temporal distributions of the speed, respectively.

Following our investigation, we determine that the SDP extraction algorithm exhibits polynomial time complexity. Specifically, the element-wise operations defined in Equation 9 and 10 operate on a three-dimensional tensor, leading to a time complexity for each operation of $\mathcal{O}(N_\Delta \times W_T \times N_S)$. In practice, the dimensions of this tensor tend to be of similar scales. Therefore, the cumulative time complexity of the SDP extraction process is approximately $2\times\mathcal{O}(N_\Delta \times W_T \times N_S) \approx \mathcal{O}(N^3)$, affirming its practical applicability for real-world implementation.

D. Comparison With Different Features

In this subsection, to provide more intuitive explanations of our proposed SDP and its unique advantages, we compare it with two other distinct features, phase difference and DFS, across various domains.

Experiments were conducted in both a living room and a meeting room, as depicted in Figure 8. In these settings, Wi-Fi devices record CSI measurements as a volunteer performs almost identical fall activities in varying orientations and locations. The phase difference of CSI is calculated as described in RTFall [17], and DFS information is obtained through STFT, following FallDeFi's approach [15], with a band-pass filter applied to minimize noise. Additionally, SDP is extracted using the method outlined in Section IV-C.

Figure 4 showcases the pattern of phase difference, DFS, and SDP features during human falls. As can be seen, the CSI phase difference is highly sensitive to the orientation, location and environment. The CSI phase difference fluctuates drastically with the changes of domain, making it difficult to represent human activities. The DFS profiles exhibit different frequency shifts when the volunteer falls at the same location (location #1) but to different orientations (orientation #1 and #2) and the volunteer falls to the same orientation (orientation #1) but at different locations (location #1 and #2). The DFS profiles also demonstrate different modes with the changes of environment. Theoretically, the DFS profiles can only capture the radial velocity of the target, but has no perception ability of the tangential motion. Therefore, the DFS profiles depend on the deployment of the device, as well as the location and orientation of the user. In contrast, SDP demonstrates almost consistent characteristics

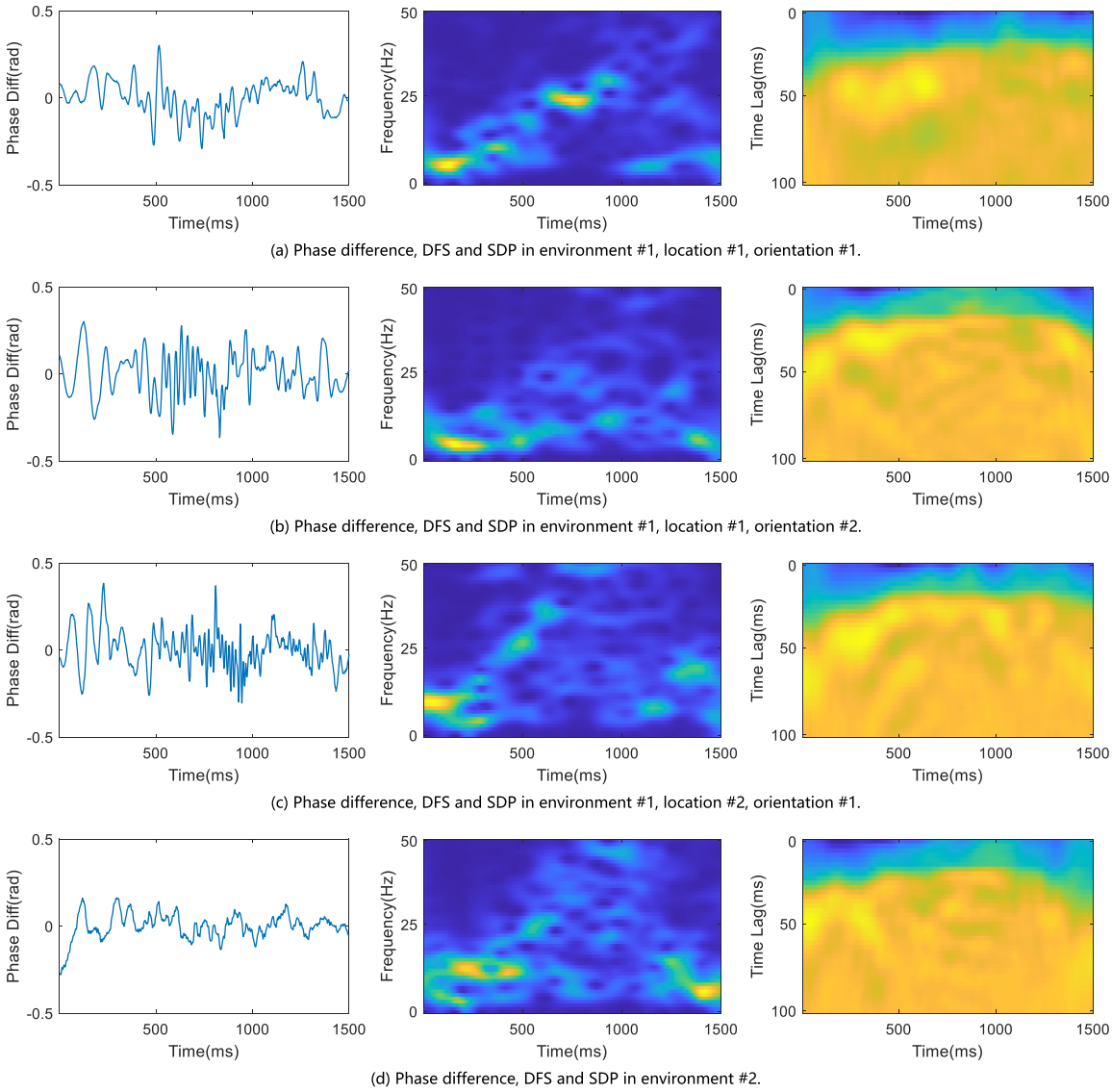


Fig. 4. Phase difference, DFS and SDP of falls in different domain settings, including different environments, locations, and orientations.

for the same fall type under different orientations, locations, and environments, thereby showcasing its strong capability for domain generalization.

V. Spatial and Temporal Attention Network

In *XFall*, the **Spatio-Temporal-Attention-based Transformer Encoder (STATE)** shown in Figure 5 is proposed to fully exploit the general fall representation (GFR) from the input SDP. In STATE, the transformer encoder (Section V-A) [41] is the basic building block, which learns the associations and dependencies of the input features. Inspired by the network structure of the basic transformer encoder, we design the spatial transformer encoder (Section V-B) and the temporal transformer encoder (Section V-C) for feature extraction on each dimension of the SDP, respectively. Specifically, in Figure 5, the SDP is first split in to W_T column vectors $\boldsymbol{S}_i$, each with a size of $N_\Delta \times 1$, representing the spatial speed distribution at a specific time t_i. A sequence of $\boldsymbol{S}_i$ is input into different spatial encoders to extract the environment speed information. All the encoded spatial embeddings are gathered and treated as series data, and put into the temporal encoder.

A. Transformer Encoder

Figure 6 illustrates the general structure of a transformer encoder block. A transformer encoder is a self-attention-based model, which effectively extracts a high-level description of the input embedding like video streams and text sequences [41]. Firstly, the linear projection and position embedding operations are applied to the input sequence. Then, a multi-head self-attention operation is performed to capture the dependencies among the input patches, followed by the fully-connected feed-forward blocks to capture the dependencies among the input patches. It's worth mentioning that a residual connection followed by layer normalization is implemented after each block, to increase the sensitivity of the network and avoid the vanishing gradient problem [42].

Among all the blocks in the transformer encoder, the multi-head self-attention block attempts to build the interactions of

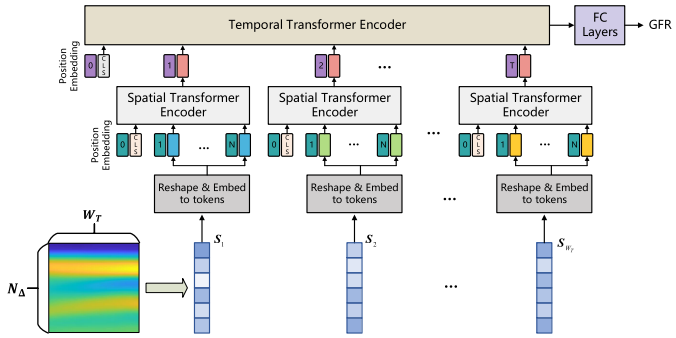


Fig. 5. Illustration of the spatio-temporal attention-based transformer encoder.

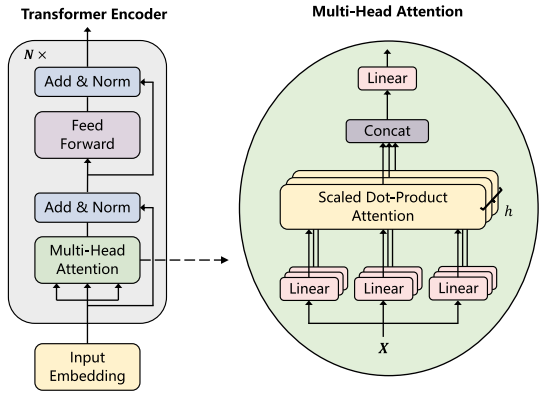


Fig. 6. The network architecture of the transformer encoder.

the input features at different time patches. Given the input $\boldsymbol{X} \in \mathbb{R}^{L\times d_{in}}$, the linear projection model generates the queries $\boldsymbol{Q} \in \mathbb{R}^{L\times d_Q}$, keys $\boldsymbol{K} \in \mathbb{R}^{L\times d_K}$, and values $\boldsymbol{V} \in \mathbb{R}^{L\times d_V}$ respectively as follows:

$$\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_Q, \boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_K, \boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_V, \tag{11}$$

where $\boldsymbol{W}_Q \in \mathbb{R}^{d_{in}\times d_Q}$, $\boldsymbol{W}_K \in \mathbb{R}^{d_{in}\times d_K}$, and $\boldsymbol{W}_V \in \mathbb{R}^{d_{in}\times d_V}$ are projection parameters, and $d_Q = d_K$. The self-attention model outputs a weighted sum of the values, where weights are calculated by the dot-product of the query with the corresponding key, which can be defined as:

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathsf{T}}}{\sqrt{d_K}}\right)\boldsymbol{V}. \tag{12}$$

In practice, we employ multi-head attention with h attention heads, in which the self-attention calculation is calculated for h times independently to capture the information from various demonstration sub-spaces. The outputs of projections are concatenated and projected again to obtain the final output:

$$\text{MultiHead}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = (\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_h)\,\boldsymbol{W}_O, \tag{13}$$

where $\boldsymbol{a}_i = \text{Attention}\left(\boldsymbol{X}\boldsymbol{W}_Q^i, \boldsymbol{X}\boldsymbol{W}_K^i, \boldsymbol{X}\boldsymbol{W}_V^i\right)$, and $\boldsymbol{W}_O \in \mathbb{R}^{hd_V\times d_O}$ is the final projection parameter. To ensure the inputs and outputs have the same dimensions, set $d_{in} = hd_V$.

Through the multi-head self-attention blocks, we acquire the deep interactions of the input sequences.

B. Spatial Feature Extraction

Transformer encoder as introduced in Section V-A can extract high-level representation for the input data. However, applying the transformer encoder to extract spatial features from wireless signals presents two primary challenges. The first challenge is the substantial number of output vectors generated by the encoder, leading to increased computational demands. This, in turn, may result in overfitting on the training set when used for classification tasks. Secondly, while the transformer encoder does not necessitate specifying the position of each patch, the sequential order of input patches in wireless signals typically denotes the power distribution across various spatial states, bearing significant physical implications.

To deal with the above-mentioned problems, we propose a spatial transformer encoder based on the basic transformer encoder by adding both the **CLS** token and the learnable position embeddings.

Specifically, the spatial transformer encoder focuses on the spatial speed distribution instead of the temporal change. The input of each spatial transformer encoder is a column vector $\boldsymbol{S}_i$ with a size of $N_\Delta \times 1$, which describes a random process with respect to the spatial speed distribution. To accelerate the training and inference process, each vector $\boldsymbol{S}_i$ is first reshaped to a 2-D matrix $\boldsymbol{X}_i \in \mathbb{R}^{L\times d_S}$, where $N_\Delta = L \times d_S$. After the reshaping, each row of $\boldsymbol{X}_i$ can be viewed as an input patch, and there are L patches for each spatial transformer encoder. To solve the first problem and achieve effective learning for classification tasks, a classification token $\mathbf{CLS} \in \mathbb{R}^{1\times d_S}$ is inserted into the beginning of embedded patches as the representation of the entire input patches [43]. The output state of the transformer encoder of the **CLS** token serves as the classification feature of the input patches. Therefore, the final input data of a transformer encoder is $\boldsymbol{X}_i' = (\mathbf{CLS}; \boldsymbol{X}_i) \in \mathbb{R}^{(L+1)\times d_s}$.

In addition, we add a standard learnable position embeddings $\boldsymbol{P}_i \in \mathbb{R}^{(L+1)\times d_s}$ to each patch embeddings to overcome the second problem and preserve the position information. After the position embedding module, the input element of each transformer encoder can be illustrated as $\tilde{\boldsymbol{X}}_i' = \boldsymbol{X}_i' + \boldsymbol{P}_i$. Through the spatial transformer encoder, a set of high-level features $\boldsymbol{Y}_i \in \mathbb{R}^{(L+1)\times d_s}$ can be obtained corresponding to the input $\tilde{\boldsymbol{X}}_i'$. We choose the output feature of **CLS** token $\boldsymbol{y}_i^0 \in \mathbb{R}^{1\times d_s}$ as the output of the spatial transformer encoder. The output series of all W_T spatial transformer encoders are used as the input for following temporal modeling.

For each moment, the spatial encoder focuses on the fall-specific speed spatial distribution and pays less attention to other speed components, since the self-attention module can adaptively allocate different attentions to different power distributions of the input.

C. Temporal Feature Extraction

In this subsection, we introduce the temporal encoder, which aims to extract the temporal feature. Like the spatial feature encoder, the temporal attention encoder also faces problems with many output vectors and missing input patch positions. To address the above problems, we propose the temporal attention encoder, which pays more attention to the position where the speed distribution fluctuates dramatically.

The input series of the temporal attention network are the output of W_T spatial transformer encoders, denoted as

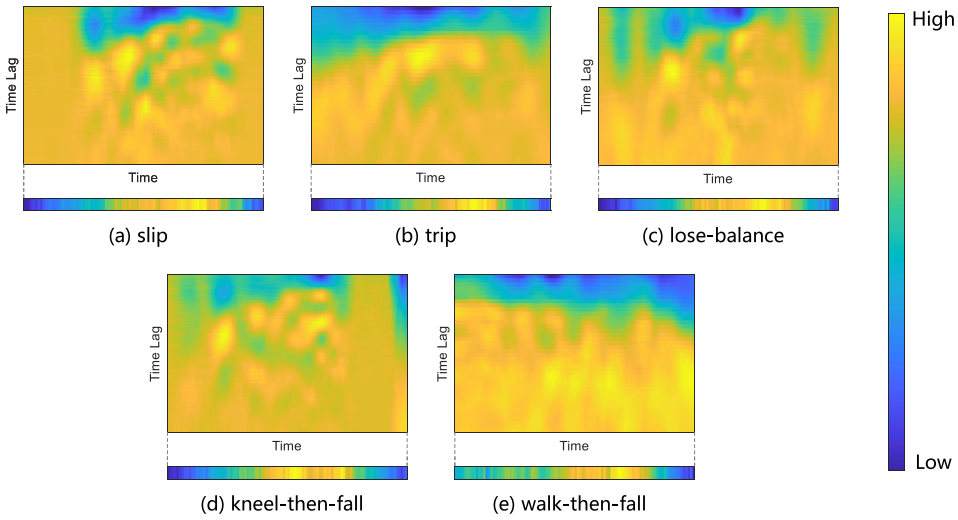


Fig. 7. The SDP and temporal attention of different fall types, including slip, trip, loss-balance, kneel-then-fall, and walk-then-fall.

$\boldsymbol{y}^0 = \left(\boldsymbol{y}_1^0, \boldsymbol{y}_2^0, \ldots, \boldsymbol{y}_{W_T}^0\right) \in \mathbb{R}^{W_T \times d_s}$, where each element $\boldsymbol{y}_i^0$ can be viewed as the input patch and there are W_T patches for the temporal attention network. Same as Section V-B, we add the classification token **CLS** and the position embeddings to respond to the two previous problems and convert the transformer encoder in Section V-A for temporal feature extraction.

After the temporal transformer encoder, we capture the high-level features. We extract the output feature of **CLS** token $\boldsymbol{z}^0 \in \mathbb{R}^{1 \times d_s}$ as the output of the temporal transformer encoder. Then, several fully-connected layers are applied to the output feature $\boldsymbol{z}^0$ to achieve GFR, demonstrating the general representation of human fall.

To demonstrate the performance of the temporal attention mechanism, a volunteer performs different fall types (including slip, trip, lose-balance, kneel-then-fall and walk-then-fall) under the same setting. While these activities were performed, Wi-Fi devices collected CSI series. We extract the SDP features as introduced in Section IV-C, and calculate the attention weights at each moment by the temporal attention encoder. Figure 7 shows the SDP and the attention weights at different moments of the five fall types. As can be seen, there are large variances among the SDP of different fall types. The temporal attention encoder provides different attention for each moment. During the fall process, the encoder assigns more attention to the moment with higher speed, which is a key element to determine whether a person falls. With the help of the attention mechanism, the temporal encoder focuses on the key motion during the fall process. Thus, the temporal attention encoder captures the essential temporal information.

VI. Cross-Modal Unified Representation Learning

Our proposed STATE is an effective model to extract the spatio-temporal representation of human activities. However, with a large number of learnable parameters, the transformer model usually requires a large amount of training data. Unfortunately, the data collection and labeling process of Wi-Fi datasets are labor-intensive because on-line labeling process is necessary for the Wi-Fi sensing dataset, which is different from image data or text data.

Inspired by knowledge distillation [21], we put forward the **Cross-modal Unified Representation Learning (CURL)** framework, which utilizes visual signals collected in synchronization with Wi-Fi data to transfer the classification capacity from the visual domain into the wireless feature domain during the training process. The CURL framework benefits our *XFall* in two aspects. First, the feature maps from the visual classification network are leveraged to supervise the training of STATE, which effectively reduce the demand of labeled Wi-Fi data. Moreover, the CURL framework inspires STATE to learn the associations between the visual representation and the Wi-Fi feature, improving our system accuracy and speeding up the training process.

It's worth mentioning that the vision-based network is only leveraged to "guide" the STATE during the training process. Once the Wi-Fi-based STATE is trained, *XFall* only takes Wi-Fi signals as the input.

A. Vision-Based Network

For effective cross-modal learning, we choose a simple-yet-effective model called C3D [44], which is consisted of several 3D convolution layers and fully connected layers. To build up a vision-based fall detection network, we first choose a pre-trained C3D model, and fine-tune it based on our collected videos of human fall and normal activities. Specifically, during the fine-tuning process, the learnable parameters of convolution layers are frozen, and only the fully connected layers can be adjusted. After we get a trained C3D model for fall detection, we extract feature map $\widetilde{\boldsymbol{r}}$ from the middle part of the fully connected layers, which will be used for the following cross-modal learning.

B. Cross-Modal Supervision

During the training stage, the synchronized video and Wi-Fi data are put into the CURL framework to train both the

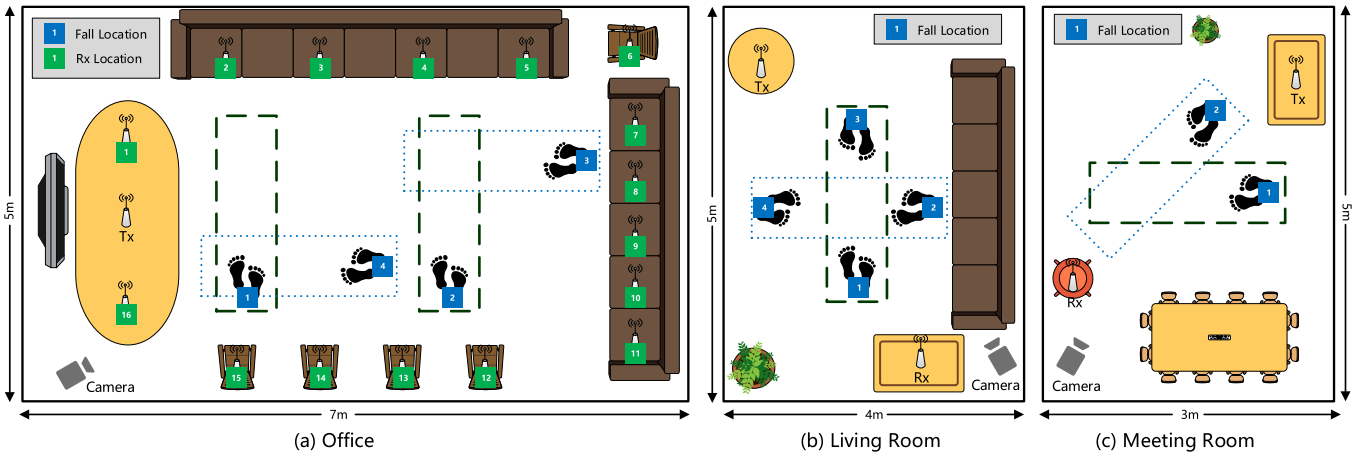


Fig. 8. Layouts of three evaluation environments.

STATE and the MLP. Based on [45], it will achieve better performance to learn the feature maps of the network and the final classification results simultaneously. Consequently, there are two training objectives for STATE: 1) The output GFR of STATE should match the output feature map $\widetilde{\boldsymbol{r}}$; 2) The final fall detection result should match the ground truth.

Consequently, the loss function can be formulated into two parts. On the one hand, we adopt the mean square error (MSE) to evaluate the effectiveness of GFR. Assuming the GFR from STATE and the supervision information of the visual model for i-th sample is $\boldsymbol{r}_i$ and $\widetilde{\boldsymbol{r}}_i$ respectively, the supervision loss function can be defined as:

$$L_{\mathrm{sv}} = \sum_{i=1}^{N} \|\widetilde{\boldsymbol{r}}_i - \boldsymbol{r}_i\|^2 / N, \tag{14}$$

where N is the total number of training samples.

On the other hand, we use binary cross-entropy to evaluate the classification performance, which can be illustrated as:

$$L_{\mathrm{c}} = -\sum_{i=1}^{N_0} \left(Y_i \log P\left(Y_i'\right) + \left(1 - Y_i\right) \log\left(1 - P\left(Y_i'\right)\right)\right), \tag{15}$$

where Y_i and Y_i' are the ground truth label and predicted label of the i-th sample. Overall, the final loss function can be written in the following forms:

$$L = L_{\mathrm{c}} + \lambda L_{\mathrm{sv}}, \tag{16}$$

where λ is a balance factor. Performing backpropagation and optimization algorithms, the STATE and the MLP layers for classification are trained.

VII. Evaluation

A. Experiment Methodology

1) Implementation: We implement *XFall* prototype on the commercial Wi-Fi product Huawei AX3 Pro, which supports the latest 802.11ax standard. Both the transmitter and the receiver work at a central frequency of 5.825 GHz. All the transmitters and receivers are equipped with two antennas, forming into a 2×2 MIMO array. The transmitter sends Wi-Fi packets at an injection rate of 350 packets per second to extract CSI information. The software part of *XFall* is implemented in a hybrid-programming way. Specifically, we utilize MATLAB for signal processing and SDP generation, and PyTorch for building and training deep learning models.

TABLE I
Detailed Evaluation Setup

Aspect	Details
Device Config	
Device Type	Huawei AX3 Pro
Standard	IEEE 802.11ax
Frequency	5.825 GHz
Antenna	2×2 MIMO
Packet Rate	350 packets per second
Data Collection	
Total Samples	1,000 fall samples, 2,800 normal activity samples
Environments	Office, Living Room, Meeting Room
Volunteers	5 volunteers with diverse heights and body types
Fall Types	Slip, Trip, Lose-balance, Kneel-fall, Walk-fall
Normal Activities	Walking, Sitting, Standing, Jumping, Bending, Squatting
User States	2-4 different locations and orientations each scenario
AP Deployment	18 different Tx-Rx settings
Groundtruth	Recorded by 1080P RGB camera at 20 FPS
Evaluation	
Metrics	Missed Alarm Rate, False Alarm Rate, F1-Score
Experiments	Cross-type, Cross-environment, Cross-user

2) Evaluation Setup: The configuration and methodology of our evaluation are shown in Table I, demonstrating the extensive efforts to reflect real-world applicability of *XFall*. Our experiments span three typical indoor environments: an office room, a living room, and a small meeting room, encompassing a comprehensive range of over 70 diverse domains (e.g., varying indoor layouts, access point deployments, user locations), as illustrated in Figure 8. This broad spectrum of testing conditions is integral to ensuring that *XFall*'s evaluation is reflective of the myriad scenarios it may encounter in practical deployment. Specifically, in the office, we deploy the transmitter at a fixed location, and the receiver at 16 different locations. In the living room and meeting room, both the transmitter and receiver are placed at given locations. For each receiver's deployment, we set 2-4 different locations where the volunteer may fall. A 1080P RGB camera, operating

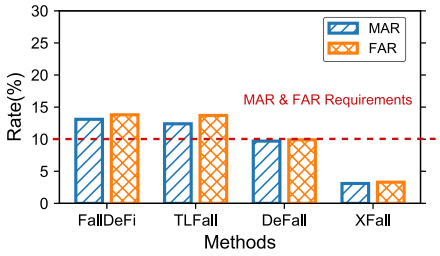


Fig. 9. Overall comparison with SOTAs.

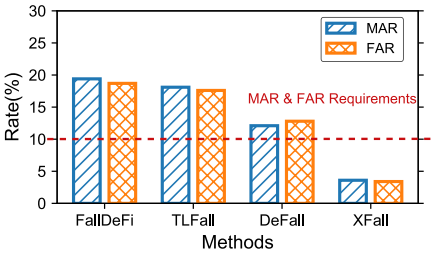


Fig. 10. Cross-type comparison with SOTAs.

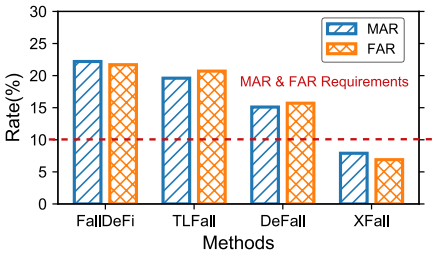


Fig. 11. Cross-environment comparison with SOTAs.

at 20 FPS, was employed alongside the CSI data acquisition, ensuring precise synchronization of observed fall events with the collected data.

Our experimental approach not only validates the functionality of *XFall* across a broad spectrum of real-life situations but also highlights the system's operational compatibility with standard, commercially available Wi-Fi equipment. This aspect is crucial, as *XFall*'s ability to function effectively with a single Wi-Fi link aligns perfectly with most home usage scenarios. Furthermore, the relevance and practicality of our experimental setup and device deployment have been endorsed by wireless device manufacturers with significant expertise in real-world deployment, confirming the practical viability of our system.

3) Data Collection: Five volunteers with different heights and somatotypes participate in our experiments. To capture fall samples, we require the volunteers to accomplish fall motions with different fall types at the given locations in Figure 8. The volunteers are asked to perform daily movements in the experiment scenario when obtaining normal samples. In total, we collect over 1,000 fall samples and 2,800 normal samples during the experiment. The types of fall samples include slip, trip, lose-balance, kneel-then-fall, and walk-then-fall. The normal activities include slow walking, fast walking, sitting down, standing up, jumping, bend-and-pickup, and squatting. All experiments are approved by our IRB.

4) Metrics: In accordance with previous researches [15], we utilize two straightforward metrics for performance evaluation: the Missed Alarm Rate (MAR) and the False Alarm Rate (FAR). The MAR, indicating the system's sensitivity to fall activities, is defined as the proportion of fall events that are incorrectly undetected. Conversely, the FAR, reflecting the system's precision in avoiding false alerts, is the proportion of non-fall events erroneously detected as falls.

B. Overall Performance

We compare *XFall* with three state-of-the-art (SOTAs) fall detection solutions, FallDeFi [15], TLFall [18], and DeFall [16]. FallDeFi and TLFall extract DFS and DWT profiles, respectively, and select several statistical features for fall detection. Then, they adopt SVM to classify falls and normal activities. DeFall extracts speed streams from raw CSI and classifies fall activities with a template matching method. To compare the system performance, we split the total data samples into train and test datasets and adopt a ten-fold cross-validation method for evaluation. Figure 9 shows the results. *XFall* achieves an overall accuracy of 96.8% with an average MAR of 3.1% and FAR of 3.3%, outperforming the state-of-the-art works.

To evaluate the system robustness, we conduct cross-type and cross-environment comparisons. Cross-type and cross-environment signify that the testing sets comprise data encompassing different fall types and environments, distinct from those in the training set. As shown in Figure 10 and Figure 11, *XFall* achieves consistent higher performance across different domains than other existing works, demonstrating its domain-adaptive performance.

Adhering to the industry-informed empirical benchmarks, maintaining that MAR and FAR below 10%, especially under cross-type and cross-environment conditions, is critical for ensuring the practical applicability of commercial fall detection systems. Existing solutions like FallDeFi and TLFall struggle to align with this recommendation, and DeFall's performance only aligns marginally under consistent conditions. Our system, in contrast, successfully adheres to this broadly-recognized guideline across various domains, underlining its practical utility in real-world scenarios.

Compared with *XFall*, FallDeFi and TLFall take SVM as the classification model, which is much simpler than our proposed STATE, and thus fails to fully exploit the information from the input features. DeFall adopts an estimated speed value as the input feature, which can demonstrate fall activities. However, considering that speed values of various activities may be similar, the template matching method has limitations for fall classification. Instead, *XFall* extracts the speed distribution, which describes human activities better than a single speed value. Moreover, our adopted spatio-temporal attention-based model also helps *XFall* to achieve better performance.

C. Robustness Analysis

1) Performance Across Different Fall Types: In this experiment, we evaluate the performance of *XFall* across different fall types. We collect fall samples with five typical fall types,

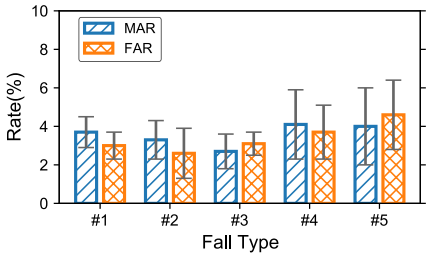


Fig. 12. Performance across fall types.

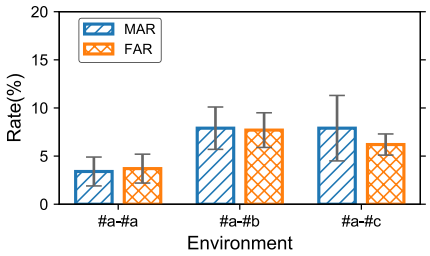


Fig. 13. Performance across environments.

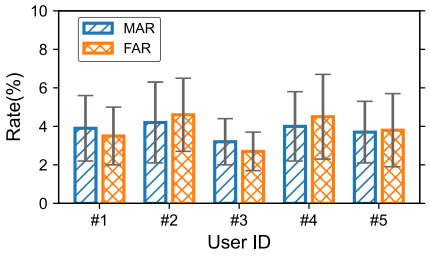


Fig. 14. Performance across users.

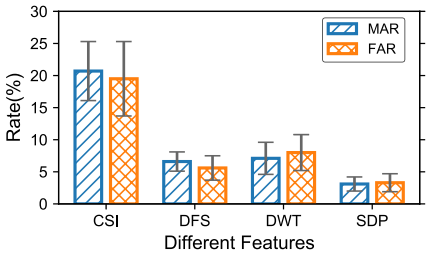


Fig. 15. Impact of features.

including slip, trip, lose-balance, kneel-then-fall, and walk-then-fall, which are labeled from #1 to #5. We select data from one fall type and part of the normal types for testing and data of the other four fall types and the remaining normal data for training. The results depicted in Figure 12 show that *XFall* maintains consistent performance across a variety of fall types. This demonstrates *XFall*'s ability to apply what it has learned from observed falls to accurately identify falls it hasn't been directly trained on, markedly boosting its ability to generalize. This makes *XFall* a dependable solution for real-world application. The rational behind the delightful result lies in two folds: 1) SDP focuses on the spatio-temporal speed distribution rather than singular speed values, capturing a more comprehensive pattern of various fall types instead of relying solely on contrasting speed values; 2) Our proposed STATE employs an attention mechanism to analyze fall activities more thoroughly, thereby extracting more detailed, fall-specific information. Consequently, *XFall* is adaptive to different fall types.

2) Performance Across Different Environments: Assessing the adaptability of *XFall* across diverse environments is critical for understanding its practical application scope. In this experiment, we trained *XFall* on data collected exclusively from environment #a, as depicted in Figure 8, and then tested its efficacy in all three environments including two unseen environment. The results, illustrated in Figure 13, reveal a marginal performance variation when transitioning across different testing environments. Remarkably, both the Missed Alarm Rate (MAR) and False Alarm Rate (FAR) remained approximately at 7.0%, despite the shift to novel environments. This stability not only demonstrates *XFall*'s robust cross-environment capability but also underscores its significant generalization potential. The foundational principle enabling *XFall* to maintain high performance across unfamiliar environments lies in the design of the SDP. By focusing on universal characteristics of human falls, independent of environmental context, SDP effectively isolates fall recognition from environment-specific noise. Consequently, *XFall*, trained in any given environment, is adept at generalizing and accurately detecting falls in any unseen environment, showcasing its profound adaptability and readiness for real-world deployment.

3) Performance Across Different Users: Understanding how *XFall* performs among a variety of users is vital for assessing its practicality in real-world scenarios. This experiment aims to test *XFall*'s ability to generalize fall detection capabilities across different individuals, each with unique behavior patterns and physical characteristics. We engaged five volunteers, labeled from #1 to #5, to participate in this study. The model was trained using data from combinations of four users and then tested on the remaining unseen user's data. The strategy was designed to evaluate how well *XFall* could adapt to new users it had not encountered during training. The findings, as detailed in Figure 14, highlight that *XFall* achieved an impressive consistency in its performance, with both the Missed Alarm Rate (MAR) and False Alarm Rate (FAR) averaging around 4.0% across different users. This level of robustness testifies to *XFall*'s capacity to discern user-independent, robust fall characteristics effectively.

D. Micro Benchmarks

1) Impact of Features: We compare four types of features with different levels of abstraction from raw CSI measurements, including denoised CSI, DFS profiles, DWT profiles, and SDP, by feeding them into the same deep learning model as introduced in Section V. The training and testing datasets are both composed of samples from the three environment settings. The evaluation results are shown in Figure 15. As can be seen, the SDP outperforms the denoised CSI, DFS, and DWT with an average decrease of 33.8%, 5.8%, and 8.7% in terms of MAR+FAR. Compared with CSI, SDP digs into the speed distribution information with physical significance by the prior knowledge of electromagnetic fields, which makes

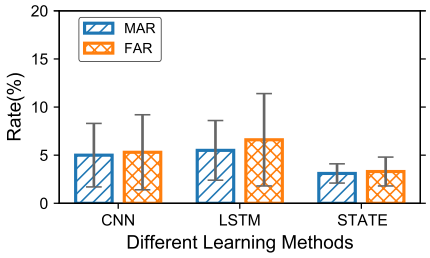


Fig. 16. Impact of learning models.

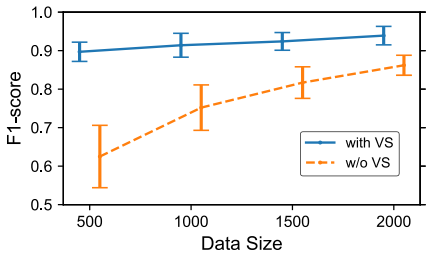


Fig. 17. Effectiveness of video supervision.

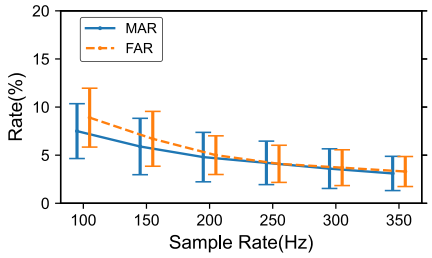


Fig. 18. Impact of sample rates.

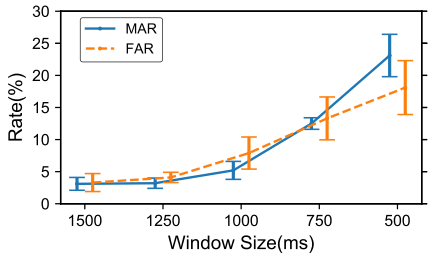


Fig. 19. Impact of window size.

the deep model easier to learn and optimize. In addition, DFS and DWT contain information related to environments and human activities. In contrast, SDP pays more attention to activity-specific information, irrelevant to the environment, location, and orientation. Consequently, SDP can achieve better performance in practice. The result shows that SDP is an ideal feature for fall detection.

2) Impact of Learning Models: We evaluate the effectiveness of the proposed STATE classifier. With SDP as input, we classify fall and normal activity with CNN, LSTM, and STATE deep network. As shown in Figure 16, STATE outperforms CNN and LSTM models with an average decrease of 4.0% and 5.7% in terms of MAR+FAR. our model effectively exploits the information of human activity from spatial and temporal aspects respectively. In addition, the attention mechanism helps to focus the significant moments and spatial features. Thus, compared with other traditional models, our model can tap into the information of human action, which performs better performance.

3) Effectiveness of Video Supervision: In this experiment, we evaluate the impact of video supervision by utilizing different numbers of samples as the training sets and compare the performance of the system with and without video supervision. We use F1-score as the evaluation metric, which is the combined result of MAR and FAR. As illustrated in Figure 17, the F1-score of the supervised system exceeds the one without supervision. In other words, to achieve the same F1-score, the former requires fewer training samples than the latter. As the number of training samples increases, the performance gap gradually decreases. Based on the evaluation result, we can conclude that supervision from the visual network improves the efficiency of the deep learning model and reduces the amount of Wi-Fi data for training.

4) Impact of Sampling Rates: In the above experiments, we evaluate *XFall* with the sample rate of 350 Hz. To demonstrate the system performance with different sampling rates, we downsample the CSI to different sampling rates from 100 Hz to 300 Hz for fall detection. We also adjust the input shape of the STATE according to different sampling rates. The evaluation results are shown in Figure 18. As can be seen, MAR and FAR slightly increase by 1.7% and 1.6% when the sampling rate decreases from 350 Hz to 200 Hz; when the sampling rate further decreases from 200 Hz to 100 Hz, MAR and FAR increase by 2.7% and 4.0%. Theoretically, fall movements are fast activities, which can be captured sensitively by a high sampling rate. From the experimental results, higher sampling rates can achieve better system performance. In addition, the improvement of system performance is not significant with the sampling rate above 200 Hz. Consequently, we could achieve ideal performance on COTS devices without the demand for an extremely high sampling rate.

5) Impact of Window Size: Based on empirical observation, the duration of fall movements is around 1000 ms to 1500 ms. In this experiment, we evaluate the impact of the window size on the system performance. We select different time windows of 1500 ms, 1250 ms, 1000 ms, 750 ms, and 500 ms. The results are shown in Figure 19. MAR and FAR slightly increase by 2.1% and 4.6% when the window size decreases from 1500 ms to 1000 ms. However, MAR and FAR increase by 17.9% and 10.2% when the window size further decreases from 1000 ms to 500 ms. Theoretically, too short window sizes would result in the incomplete capture of fall activities, leading to system performance degradation. Results demonstrate that we should set the window size to 1500 ms for the best performance.

6) System Latency Analysis: To validate the efficiency of *XFall*, we further evaluate the system latency. Figure 20 shows the end-to-end latency of *XFall* running on a low-power commercial laptop. The latency consists of SDP calculation delay and STATE classification delay. As shown, with 1500 ms of CSI as input, the average end-to-end latency is 59.9 ms, with an average of 46.8 ms for SDP calculation and 13.1 ms for STATE classification. Results show that most of the system latency is induced by the SDP calculation, particularly

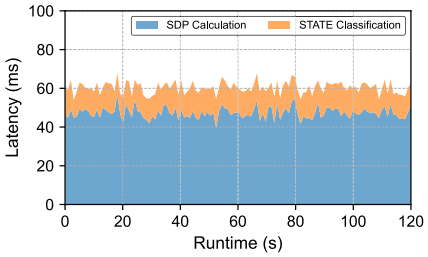


Fig. 20. System latency analysis.

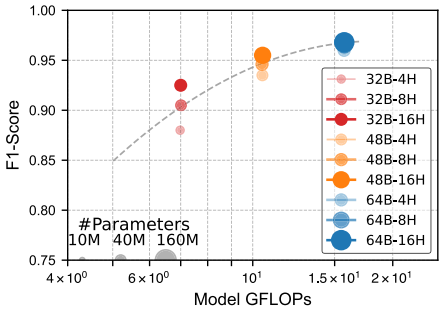


Fig. 21. Model scalability analysis.

due to the auto-correlation function computation. To conclude, the latency evaluation results indicate that *XFall* could achieve real-time monitoring for fall detection, emphasizing its readiness and reliability for deployment in scenarios where immediacy is critical.

7) Model Scalability Analysis: Evaluating the scalability of the STATE model within *XFall* is crucial for understanding its performance under varying configurations. To achieve this, we examined how various configurations affect model parameters, computation overhead, and overall accuracy, providing insights for optimal model choice. Specifically, we experimented with 9 different configurations, stemming from 3 variations of attention block numbers (32B, 48B, 64B) and 3 different amounts of attention heads (4H, 8H, 16H), as illustrated in Figure 21. The results reveal that increasing both the number of blocks and attention heads leads to a rise in the model parameters and performance. However, we observed a diminishing return in performance gains. For instance, expanding the parameter size from 10M to 40M resulted in over a 5% increase in performance. Yet, a further increase from 40M to 160M yielded only around a 3% improvement. Moreover, it was found that adding more attention heads enhances both model performance and the number of parameters without significantly adding to the overall computational load. This is attributed to the decreased computational demand per attention path as the number of attention heads increases. The above findings indicate that our STATE model exhibits commendable scalability. Further augmentation in the number of blocks and attention heads is anticipated to enhance model precision further, showcasing *XFall*'s potential for high accuracy in fall detection. This scalability analysis underscores the STATE model's flexibility and its ability to adapt to various computational platforms, making *XFall* a versatile tool for real-time fall detection.

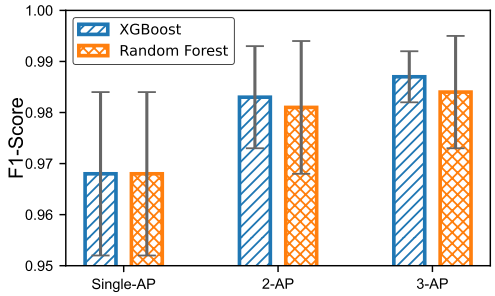


Fig. 22. Effectiveness of multiple AP collaboration.

8) Effectiveness of Multiple AP Collaboration: While *XFall* is designed to operate effectively with a single Wi-Fi link, exploring the potential for performance enhancement through multi-link collaboration was deemed essential to unveil its performance limits. In this context, we investigated the feasibility of joint fall detection by integrating CSI data from multiple APs deployed within the same environment. For this purpose, 1 to 3 sets of *XFall* units were deployed, and their outputs were combined using two classical ensemble learning strategies: XGBoost [46] and Random Forest [47]. As depicted in Figure 22, integrating the data from 2 APs using XGBoost and Random Forest enhanced the system's accuracy to 98.3% and 98.1%, respectively. Further expanding the collaboration to 3 APs can push the accuracy rates to 98.7% and 98.4%, albeit at the cost of over 2× computational complexity. Notably, XGBoost emerged as the more favorable ensemble learning strategy, offering higher and more consistent performance enhancements. In summary, multiple AP collaboration could significantly exceed *XFall*'s inherent performance upper bound, highlighting a promising direction for future enhancements of *XFall*.

VIII. Conclusion

This paper proposes the design and implementation of *XFall*, the first domain-adaptive fall detection system based on Wi-Fi. *XFall* derives an environment-independent feature called spatial distribution profile (SDP), and employs a novel spatio-temporal attention-based encoder (STATE) to learn the general fall representation. With our proposed cross-modal unified representation learning framework, a visual model can be leveraged to supervise the training process, which effectively reduces the need of labeled Wi-Fi data. Our result shows that *XFall* outperforms state-of-the-art Wi-Fi-based fall detection solutions in both in-domain and cross-domain evaluation, making a promising step towards practical and ubiquitous Wi-Fi sensing.

References

[1] (2021). *Falls*. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/falls

[2] G. Mastorakis and D. Makris, "Fall detection system using Kinect's infrared sensor," *J. Real-Time Image Process.*, vol. 9, pp. 635–646, Dec. 2014.

[3] C. Krupitzer, T. Sztyler, J. Edinger, M. Breitbach, H. Stuckenschmidt, and C. Becker, "Hips do lie! A position-aware mobile fall detection system," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, Mar. 2018, pp. 1–10.

[4] Z.-P. Bian, J. Hou, L.-P. Chau, and N. Magnenat-Thalmann, "Fall detection based on body part tracking using a depth camera," *IEEE J. Biomed. Health Informat.*, vol. 19, no. 2, pp. 430–439, Mar. 2015.

[5] L.-J. Kau and C.-S. Chen, "A smart phone-based pocket fall accident detection, positioning, and rescue system," *IEEE J. Biomed. Health Informat.*, vol. 19, no. 1, pp. 44–56, Jan. 2015.

[6] T. Theodoridis, V. Solachidis, N. Vretos, and P. Daras, "Human fall detection from acceleration measurements using a recurrent neural network," in *Proc. Int. Conf. Biomed. Health Informat.*, 2017, pp. 145–149.

[7] Y. Tian, G.-H. Lee, H. He, C.-Y. Hsu, and D. Katabi, "RF-based fall monitoring using convolutional neural networks," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 3, pp. 1–24, Sep. 2018.

[8] Y. Li, K. C. Ho, and M. Popescu, "A microphone array system for automatic fall detection," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 5, pp. 1291–1301, May 2012.

[9] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, "E-eyes: Device-free location-oriented activity identification using fine-grained WiFi signatures," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, Sep. 2014, pp. 617–628.

[10] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of WiFi signal based human activity recognition," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, Sep. 2015, pp. 65–76.

[11] G. Chi et al., "Wi-Drone: Wi-Fi-based 6-DoF tracking for indoor drone flight control," in *Proc. 20th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2022, pp. 56–68.

[12] Y. Gao, G. Chi, G. Zhang, and Z. Yang, "Wi-Prox: Proximity estimation of non-directly connected devices via Sim2Real transfer learning," in *Proc. IEEE Global Commun. Conf.*, Dec. 2023, pp. 5629–5634.

[13] G. Chi et al., "RF-diffusion: Radio signal generation via time-frequency diffusion," 2024, *arXiv:2404.09140*.

[14] Y. Wang, K. Wu, and L. M. Ni, "Wifall: Device-free fall detection by wireless networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 2, pp. 581–594, Feb. 2016.

[15] S. Palipana, D. Rojas, P. Agrawal, and D. Pesch, "FallDeFi: Ubiquitous fall detection using commodity Wi-Fi devices," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 4, pp. 1–25, Jan. 2018.

[16] Y. Hu, F. Zhang, C. Wu, B. Wang, and K. J. R. Liu, "DeFall: Environment-independent passive fall detection using WiFi," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8515–8530, Jun. 2022.

[17] H. Wang, D. Zhang, Y. Wang, J. Ma, Y. Wang, and S. Li, "RT-Fall: A real-time and contactless fall detection system with commodity WiFi devices," *IEEE Trans. Mobile Comput.*, vol. 16, no. 2, pp. 511–526, Feb. 2017.

[18] L. Zhang, Z. Wang, and L. Yang, "Commercial Wi-Fi based fall detection with environment influence mitigation," in *Proc. 16th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Jun. 2019, pp. 1–9.

[19] Y. Zheng et al., "Zero-effort cross-domain gesture recognition with Wi-Fi," in *Proc. 17th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2019, pp. 313–325.

[20] S. Ding, Z. Chen, T. Zheng, and J. Luo, "RF-Net: A unified meta-learning framework for RF-enabled one-shot human activity recognition," in *Proc. 18th Conf. Embedded Netw. Sens. Syst.*, 2020, pp. 517–530.

[21] A. Romero, N. Ballas, S. Ebrahimi Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for thin deep nets," 2014, *arXiv:1412.6550*.

[22] Y. Wang, S. Yang, F. Li, Y. Wu, and Y. Wang, "FallViewer: A fine-grained indoor fall detection system with ubiquitous Wi-Fi devices," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 12455–12466, Aug. 2021.

[23] E. E. Stone and M. Skubic, "Fall detection in homes of older adults using the Microsoft Kinect," *IEEE J. Biomed. Health Informat.*, vol. 19, no. 1, pp. 290–301, Jan. 2015.

[24] J. Wang, Z. Zhang, B. Li, S. Lee, and R. S. Sherratt, "An enhanced fall detection system for elderly person monitoring using consumer home networks," *IEEE Trans. Consum. Electron.*, vol. 60, no. 1, pp. 23–29, Feb. 2014.

[25] P. Pierleoni, A. Belli, L. Palma, M. Pellegrini, L. Pernini, and S. Valenti, "A high reliability wearable device for elderly fall detection," *IEEE Sensors J.*, vol. 15, no. 8, pp. 4544–4553, Aug. 2015.

[26] Q. T. Huynh, U. D. Nguyen, L. B. Irazabal, N. Ghassemian, and B. Q. Tran, "Optimization of an accelerometer and gyroscope-based fall detection algorithm," *J. Sensors*, vol. 2015, pp. 1–8, Aug. 2015.

[27] Y. Zhang, W. Hou, Z. Yang, and C. Wu, "Vecare: Statistical acoustic sensing for automotive in-cabin monitoring," in *Proc. 20th USENIX Symp. Networked Syst. Design Implement.*, 2023, pp. 1185–1200.

[28] W. Wang, A. X. Liu, and M. Shahzad, "Gait recognition using WiFi signals," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, pp. 363–373.

[29] X. Li et al., "IndoTrack: Device-free indoor human tracking with commodity Wi-Fi," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 3, pp. 1–22, Sep. 2017.

[30] K. Qian, C. Wu, Z. Yang, Y. Liu, and K. Jamieson, "Widar: Decimeter-level passive tracking via velocity monitoring with commodity Wi-Fi," in *Proc. 18th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2017, pp. 1–10.

[31] F. Zhang, C. Chen, B. Wang, and K. J. Ray Liu, "WiSpeed: A statistical electromagnetic approach for device-free indoor speed estimation," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2163–2177, Jun. 2018.

[32] C. Wu, F. Zhang, Y. Hu, and K. J. R. Liu, "GaitWay: Monitoring and recognizing gait speed through the walls," *IEEE Trans. Mobile Comput.*, vol. 20, no. 6, pp. 2186–2199, Jun. 2021.

[33] W. Jiang et al., "Towards environment independent device free human activity recognition," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, 2018, pp. 289–304.

[34] J. Zhang, Z. Tang, M. Li, D. Fang, P. Nurmi, and Z. Wang, "CrossSense: Towards cross-site and large-scale WiFi sensing," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, 2018, pp. 305–320.

[35] Z. Yang, Y. Zhang, K. Qian, and C. Wu, "SLNet : A spectrogram learning neural network for deep wireless sensing," in *Proc. 20th USENIX Symp. Netw. Syst. Design Implement.*, 2023, pp. 1221–1236.

[36] C. Li, Z. Cao, and Y. Liu, "Deep AI enabled ubiquitous wireless sensing: A survey," *ACM Comput. Surveys*, vol. 54, no. 2, pp. 1–35, 2021.

[37] Y. Zhang, Y. Zheng, G. Zhang, K. Qian, C. Qian, and Z. Yang, "GaitSense: Towards ubiquitous gait-based human identification with Wi-Fi," *ACM Trans. Sensor Netw.*, vol. 18, no. 1, pp. 1–24, Feb. 2022.

[38] D. A. Hill, *Electromagnetic Fields in Cavities: Deterministic and Statistical Theories*. Hoboken, NJ, USA: Wiley, 2009, pp. 91–150.

[39] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[40] F. Zhang, C. Wu, B. Wang, H.-Q. Lai, Y. Han, and K. J. R. Liu, "WiDetect: Robust motion detection with a statistical electromagnetic model," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 3, no. 3, pp. 1–24, Sep. 2019.

[41] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

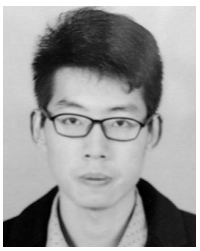
[43] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[44] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.

[45] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1365–1374.

[46] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[47] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, Oct. 2001.

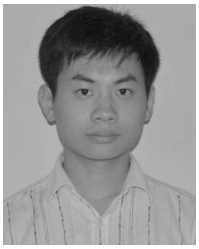


Guoxuan Chi (Member, IEEE) received the B.E. degree from the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, in 2019, and the Ph.D. degree from the School of Software, Tsinghua University, in 2024. He is currently a Research Assistant with Tsinghua University. His research interests include wireless sensing and mobile computing.

Guidong Zhang (Member, IEEE) received the B.E. degree from the Department of Electronic Engineering and Information Science, University of Science and Technology of China, in 2018, and the Ph.D. degree from the School of Software, Tsinghua University, in 2023. His research interests include wireless sensing and mobile computing.

Xuan Ding (Member, IEEE) received the B.S. and Ph.D. degrees from Tsinghua University, China, in 2008 and 2014, respectively. He is currently a Research Assistant Professor with the School of Software and BNRist, Tsinghua University. His research interests include privacy-preserving computing, blockchain, RFID, and wireless sensing.

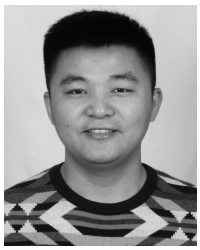
Qiang Ma (Member, IEEE) received the B.S. degree from the Department of Computer Science and Technology, Tsinghua University, China, in 2009, and the Ph.D. degree from the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, in 2013. He is currently an Assistant Researcher with the Software School, Tsinghua University. His research interests include wireless sensor networks, mobile computing, and privacy.



Zheng Yang (Fellow, IEEE) received the B.E. degree in computer science from Tsinghua University, Beijing, China, in 2006, and the Ph.D. degree in computer science from The Hong Kong University of Science and Technology, Hong Kong, in 2010. He is currently an Associate Professor with Tsinghua University. His main research interests include the Internet of Things and mobile computing. He is the PI of National Natural Science Fund for Excellent Young Scientist. He was a recipient of the State Natural Science Award (second class).

Zhenguo Du received the B.S. degree from the Chien-Shiung Wu College, Southeast University, China, in 2007, and the Ph.D. degree from the Department of Electronic Engineering and Information Science, University of Science and Technology of China, in 2012. He is currently with Huawei Technology Company Ltd., working on wireless communication, wireless sensing, and wireless localization.

Houfei Xiao received the B.S. degree from the School of Electronic Information and Engineering, Huazhong University of Science and Technology, China, in 2009, and the Ph.D. degree from the State Key Laboratory of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications, in 2014. He is currently with Huawei Technology Company Ltd., working on wireless communication, wireless sensing, and wireless localization.



Zhuang Liu received the B.S. degree in automation from Changsha University of Science and Technology, China, in 2013, and the M.S. degree in electronic engineering from the Huazhong University of Science and Technology in 2016. He is currently with Huawei Technology Company Ltd., working on wireless communication, wireless sensing, and wireless localization.