freeDoppler: A Doppler Effect Learning Network for Accurate RF-based Velocity Estimation

DAWEI YAN, School of Cyber Security and Computer, Hebei University, Baoding, China and USTC, Hefei, China
FEIYU HAN*, Nanjing University of Information Science and Technology, Nanjing, China
SHANG GAO, USTC, Hefei, China
FEI SHANG, University of Science and Technology of China, Hefei, China
PANLONG YANG*, Nanjing University of Information Science and Technology, Nanjing, China
YUBO YAN, School of Computer Science and Technology, University of Science and Technology of China, Hefei, China

Accurately estimating the velocity (including speed and direction) of moving targets has recently attracted widespread attention in augmented reality, security monitoring and sports health. In particular, the *Doppler Frequency Shift* (DFS)-based velocity estimation schemes using WiFi devices have shown great potential and have been widely studied. However, previous *Fast Fourier Transform* (FFT)-based and path-parameter-based arts have inherent limitations in DFS estimation and, worse still, ignore the nonlinear measurement errors caused by the relative orientation between the moving target and the WiFi transceiver. The above limitations make it difficult to meet the requirements for fine-grained velocity estimation in practical applications. To cope with these limitations, in this paper, we propose a learning-based velocity estimation framework, named *freeDoppler*, to achieve fine-grained, multi-target and orientation-independent velocity estimation. Specifically, we construct a WiFi-based *Velocity Estimation Network* (VEN), which leverages continuous complex-valued *Channel State Information* (CSI) sequences as input, to fully learn the inherent information of the Doppler effect and accurately predict velocity series. In addition, we adopt the electric field scattering model of Maxwell's equations to construct a physics-informed *CSI Generation Model* (CGM), thereby generating large-scale and high-quality simulated CSI samples to improve the generalization of the VEN model. Throughout extensive real-world experiments, *freeDoppler* can achieve median errors of 7.98 cm/s for speed estimation, 28° for direction estimation and 35 cm for human tracking in one or two moving targets, significantly outperforming the state-of-the-art methods.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing design and evaluation methods**; **Ubiquitous and mobile computing systems and tools**.

Additional Key Words and Phrases: WiFi-based Sensing, Doppler Frequency Shift, Velocity Estimation, Deep Learning Network.

*Corresponding Authors

Authors' Contact Information: Dawei Yan, School of Cyber Security and Computer, Hebei University, Baoding, Hebei, China and USTC, Hefei, China; e-mail: yandw@mail.ustc.edu.cn; Feiyu Han, Nanjing University of Information Science and Technology, Nanjing, Jiangsu, China; e-mail: fyhan@nuist.edu.cn; Shang Gao, USTC, Hefei, Anhui, China; e-mail: shang_gao@mail.ustc.edu.cn; Fei Shang, University of Science and Technology of China, Hefei, China; e-mail: shf_1998@outlook.com; Panlong Yang, Nanjing University of Information Science and Technology, Nanjing, Jiangsu, China; e-mail: plyang@nuist.edu.cn; Yubo Yan, School of Computer Science and Technology, University of Science and Technology of China, Hefei, China; e-mail: yuboyan@ustc.edu.cn.

ACM 1550-4867/2025/10-ART
https://doi.org/10.1145/3772369

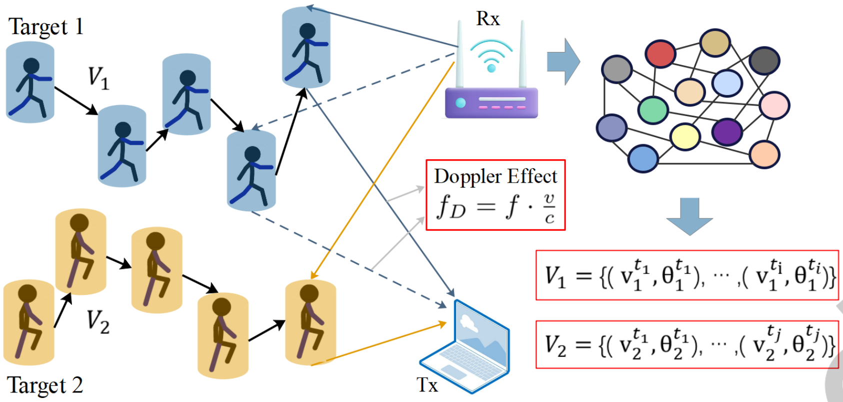

Target 1
V_1
Rx
Doppler Effect
$f_D = f \cdot \frac{v}{c}$
$V_1 = \{(\mathrm{v}_1^{t_1}, \theta_1^{t_1}), \cdots, (\mathrm{v}_1^{t_i}, \theta_1^{t_i})\}$
$V_2 = \{(\mathrm{v}_2^{t_1}, \theta_2^{t_1}), \cdots, (\mathrm{v}_2^{t_j}, \theta_2^{t_j})\}$
V_2
Target 2
Tx

Fig. 1. **By learning the fine-grained Doppler effect through neural network, *freeDoppler* can accurately estimate a series of velocities $\mathrm{V}_1, \mathrm{V}_2, \cdots$ (include speed and direction) of continuous motion of multiple targets using wireless signal, where f_D is the Doppler frequency shift, f is the carrier frequency, v is the target speed and c is the speed of light.**

1 Introduction

In real life, fine-grained estimation of human movement velocity, including both speed and direction, can significantly benefit various human-centric applications, such as virtual reality, augmented reality, security monitoring, and sports health. In particular, in sports health, accurately estimating the movement velocity of each joint in the human body provides valuable insights into motion patterns, helps assess athletic skills, and can identify abnormal activities or even potential injuries [11, 15]. For example, tracking the velocity of knee and elbow joints during physical activity enables more effective training programs, allowing for targeted interventions to improve performance and reduce the risk of injury.

Traditional approaches for velocity estimation require users to wear additional devices like motion sensors or accelerometers. While these devices can provide accurate measurements, they often create inconveniences, especially in daily life or sports settings, as they can be uncomfortable, cumbersome, or intrusive. Other non-contact methods, such as camera-based or radar-based solutions, provide a more convenient alternative. Camera-based systems [2, 6] can track human movements by analyzing visual data, but these methods face significant privacy concerns and are often constrained by *Line-of-Sight* (LoS) requirements, limiting their effectiveness in certain environments. Similarly, radar-based methods [10, 36] offer the advantage of non-line-of-sight (NLoS) sensing, but they often require the deployment of specialized equipment, such as radar antennas or sensor arrays, which can be costly and impractical for widespread use.

Benefiting from less privacy concerns, NLoS coverage, and ubiquitous infrastructures, WiFi-based solutions have shown great potential in movement velocity estimation [17, 24, 28]. In particular, the ability to leverage existing WiFi networks for human motion sensing makes it a cost-effective and scalable alternative to traditional methods that require dedicated equipment or user-worn sensors. This has spurred significant interest in using WiFi for tracking and estimating the velocity of human movement, particularly in complex environments such as indoor spaces or crowded areas. DFS arises from the change in frequency of the reflected signal due to the relative motion of a target for the WiFi transceiver. Some approaches utilize DFS to model the change in the reflection path length of the moving target, which directly relates to the Doppler velocity [17, 24]. By capturing these Doppler shifts, the velocity of the target can be estimated for the signal propagation characteristics. Furthermore, many studies have built an approximate relationship between Doppler velocity and target velocity, enabling a broad range of sensing tasks, such as human tracking [13, 18], gait recognition [23, 32], and gesture recognition [21, 35]. Unfortunately, although existing research has demonstrated the great potential of DFS for WiFi-based velocity estimation, two significant limitations severely hinder the expansion of these schemes to larger ranges and finer-grained tasks.

These two limitations are as follows:

- **Limitation 1: inherent flaws in current DFS estimation methods.** To estimate DFS accurately, existing works explore FFT-based and path-parameter-based approaches, which have their inherent limitations. Specifically, for *Fast Fourier Transform* (FFT)-based solutions [23, 24], time window and frequency range need to be determined carefully based on sensing task requirements, hindering their implementation and extension in the real world. In addition, the issues of spectrogram leakage [31] and multi-target interference bring unknown errors in the estimated DFS. For path-parameter-based solutions [13, 18], they are affected by the imperfect joint estimation of multi-path parameters, *e.g.*, *Angle of Arrival* (AoA) and *Time of Flight* (ToF). In Sec. 2.2, we present a case study to demonstrate the limitations of FFT-based and path-parameter-based solutions.
- **Limitation 2: nonlinear errors of DFS-based velocity estimation.** DFS-based velocity estimation schemes exhibit nonlinear measurement errors resulting from relative orientation between the moving target and the WiFi transceiver [14]. In addition, the estimation accuracy of the direction is also affected by the moving speed. In Sec. 2.3, we quantify the impact of relative orientation between the moving target and the WiFi transceiver on DFS-based velocity estimation. The preliminary analysis reveals that even if DFS is an unbiased estimate, the velocity estimation at different orientations is quite different. In particular, when the velocity estimation error in some key orientations (*e.g.,* the inflection point of the movement direction) is abnormally large, the estimation accuracy drops seriously.

To address the above limitations, in this paper, we propose *freeDoppler*, a fine-grained Doppler effect learning network for accurate velocity estimation using *Commercial-off-the-shelf* (COTS) WiFi devices. Specifically, *freeDoppler* takes advantage of RF-based deep learning techniques to push the limitations of traditional DFS-based velocity estimation approaches. As shown in Fig. 1, *freeDoppler* enables robust velocity estimation for multiple targets in space, regardless of the relative orientation between the moving target and the WiFi transceiver.

Firstly, to address the velocity estimation error caused by inaccurate DFS estimation in previous works, we design an RF-based *Velocity Estimation Network* (VEN), which effectively learns inherent information about the Doppler effect in CSI sequences and accurately predicts continuous velocities within each time period. Specifically, unlike previous methods that rely on either amplitude or phase information, we use complex-valued CSI in continuous time periods as the input of VEN, to fully capture information related to DFS-based velocity estimation [31], thereby improving its ability to estimate velocity accurately across continuous time periods, even in dynamic environments.

Secondly, to address the interference of multiple targets and relative orientation between the moving target and the WiFi transceiver, an intuitive approach is to collect a sufficient CSI dataset to cover the above conditions, while building such a dataset needs high collection and labeling costs. In our work, we adopt electric field scattering of Maxwell's equations instead of simple ray tracing to describe the impact of multiple targets and relative orientation on the propagation of WiFi signals. Then, based on such physical constraint, we build a physics-informed *CSI Generation Model* (CGM) to generate large-scale and high-quality simulated CSI samples. It is worth noting that to reduce computational complexity and accelerate fitting, the goal of CGM is to generate the intermediate state of complex-valued CSI instead of raw CSI.

Overall, the contributions of this paper are as follows:

- In this paper, we propose *freeDoppler*, a fine-grained deep learning network, that attempts to use the learning-based idea to mitigate the errors of DFS-based velocity estimation methods using WiFi signals.

freeDoppler can achieve accurate velocity estimation, including speed and direction, regardless of the relative orientation of the target to the transceiver.

- We build a velocity learning network VEN to learn important information of WiFi CSI to enhance the Doppler effect, and accurately predict a series of velocities in each time period. We also construct a CSI generation model CGM and use the electric field scattering principle based on Maxwell's equations and the noise addition methods to generate rich intermediate CSI for training our network VEN.
- We implement a prototype for estimating the target's motion velocities and human trajectory tracking. Extensive experiments show that the performance of *freeDoppler* is better than the state-of-the-art methods, and can achieve median errors of 7.98 cm/s for speed estimation, 28° for direction estimation, and 35 cm for human tracking in one or two moving targets, and robust to varieties of trajectories.

The remainder of this paper is organized as follows: Section 2 presents the basic principle of DFS-based velocity estimation, and analyzes the factors that affect velocity measurement from two aspects: the limitations of DFS estimation methods, and the relative orientations between target and WiFi transceiver. In Section 3, we give the overview of *freeDoppler*, and in Section 4, we show the design of each module in detail. Implementation and evaluation are presented in Section 5 and Section 6. In Section 7, we discuss the limitations of this work and future work. Section 8 discusses the related work. We finally conclude our work in Section 9.

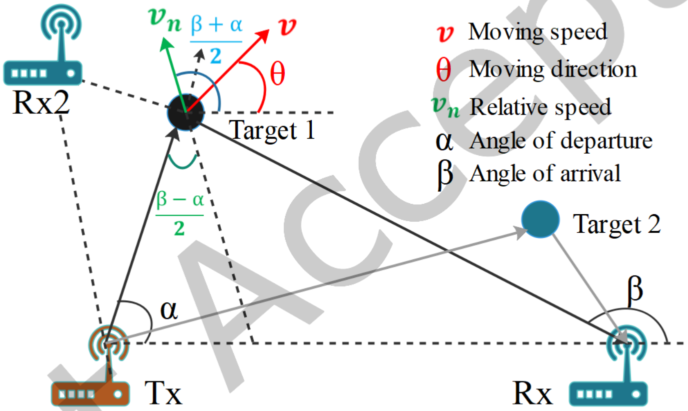


Fig. 2. **Basic principles of target movement velocity estimation.**

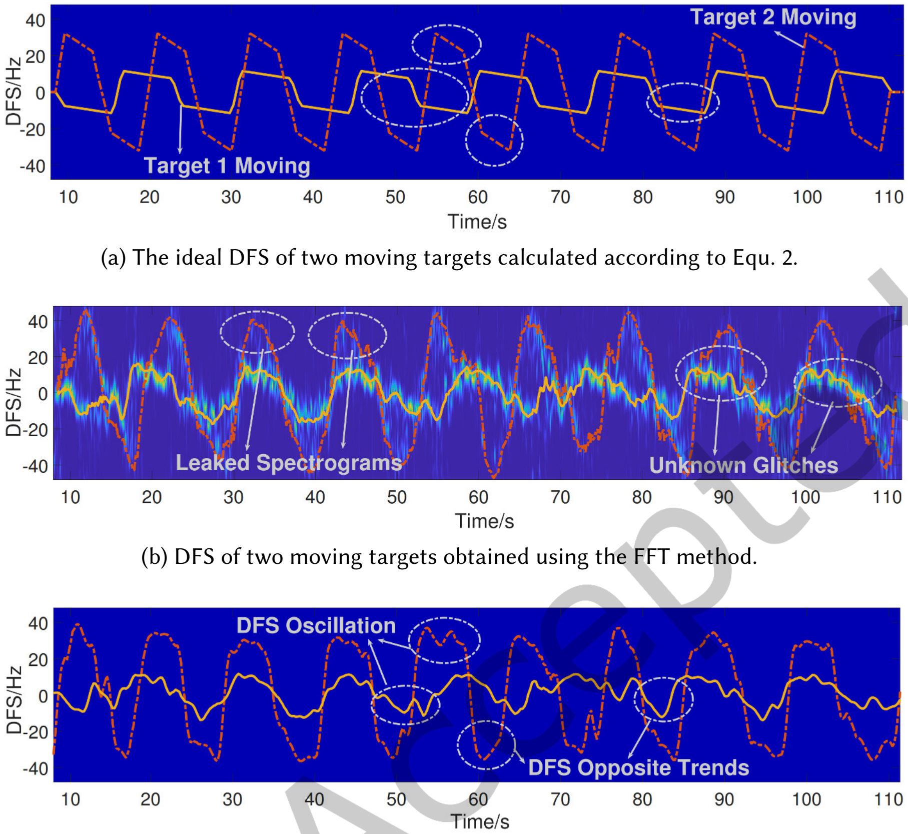
2 Understanding Velocity Estimation Accuracy

2.1 DFS-based Velocity Estimation Using WiFi Tx-Rx

The Doppler effect is the phenomenon that the wave's frequency changes when the wave source moves relative to an observer. For a WiFi sensing system composed of a pair of Tx-Rx as shown in Fig. 2, when the target moves in the environment with velocity $\mathbf{v} = (v, \theta)$ (*i.e.*, speed v and direction θ), the path length of the target's reflected signal changes accordingly ΔL, which introduces a DFS in the carrier frequency of the reflected signal:

$$f_D = \frac{f \cdot v_{\Delta L}}{c} \tag{1}$$

where f is the subcarrier frequency, c is the light speed, and $v_{\Delta L} = \Delta L/\Delta t$ is the speed at which the path length changes. In particular, when the AoD and AoA of the target relative to Tx and Rx are α and β, the above DFS f_D



(a) The ideal DFS of two moving targets calculated according to Equ. 2.

(b) DFS of two moving targets obtained using the FFT method.

(c) DFS of two moving targets obtained using the EM path parameter method.

Fig. 3. **When using FFT or EM algorithm to obtain the DFS of actual WiFi CSI, due to the inherent limitations of these methods (*e.g.*, spectrum leakage, unknown noise, path parameter correlation), the DFS estimation value may have many errors compared to the ideal value.**

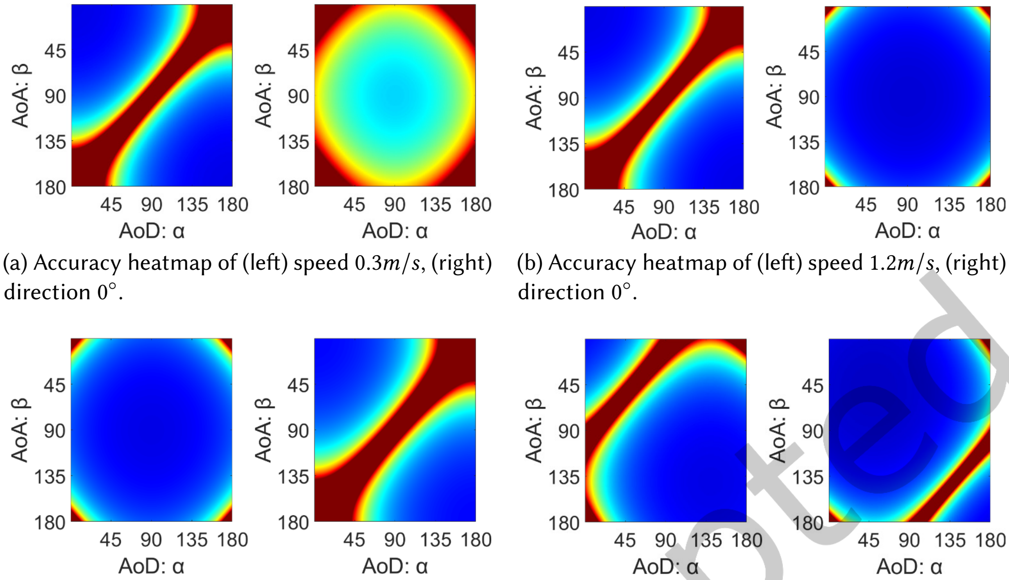
can be expressed as follows [14]:

$$f_D = -\frac{f}{c} \cdot v \cos(\frac{\beta+\alpha}{2} - \theta) \cdot 2\cos(\frac{\beta-\alpha}{2}) \tag{2}$$

where the sign of f_D is positive when the target is close to Tx-Rx, otherwise the sign of f_D is negative. Therefore, when there are two WiFi links in the environment, as shown in Fig. 2, we can simultaneously estimate the speed v and direction θ of the target movement.

2.2 Limitations of DFS Estimation Methods

The current methods of estimating DFS using WiFi signals mainly include FFT-based and path-parameter-based methods. However, both FFT-based and path-parameter-based DFS estimation methods have certain limitations. To verify and analyze the problems of different DFS estimation methods, we collect actual CSI according to the setup shown in Fig. 2, where $d_{los} = 5m$. We recruit two volunteers as the targets, and ask them to walk back and



(a) Accuracy heatmap of (left) speed $0.3m/s$, (right) direction 0°.

(b) Accuracy heatmap of (left) speed $1.2m/s$, (right) direction 0°.

(c) Accuracy heatmap of (left) speed $0.8m/s$, (right) direction 90°.

(d) Accuracy heatmap of (left) speed $1.5m/s$, (right) direction 140°.

Fig. 4. **When the relative orientation (α, β) of the target and Tx-Rx is different, the estimation accuracy of velocity $\mathbf{v} = (v, \theta)$ is different (assuming f_D is an unbiased estimate).**

forth on a path of length L along a preset speed v and direction θ. Fig. 3a shows the true values of DFS calculated according to Equ. 2 for two targets. In particular, we collect CSI when the target is walking, with a sampling rate of $100Hz$, one antenna for Tx and three antennas for Rx. It is worth noting that our test environment is an open area and we set the Tx-Rx height to $1.5m$, and volunteers are also required to strictly control their limbs and pace, thereby avoiding other interference.

Next, we use FFT-based [35] and *Expectation Maximization* (EM)-based [18] methods to estimate DFS respectively, and the results are shown in Fig 3b and Fig. 3c. From Fig. 3b, the DFS amplitude of target 2 is significantly lower than that of target 1, and DFS has unknown glitches, which is due to spectrum leakage [31]. In addition, the result of Fig. 3c indicates that the estimated continuous DFS has oscillation phenomena and even opposite moving trends, which is due to the imperfection of the process of joint estimation of multi-path parameters. It is worth noting that although real human gait involves slight and non-uniform variations in walking speed, the reference curve serves as a physically interpretable benchmark for analyzing estimation performance. The phenomena observed in the results above cannot be fully attributed to normal gait variation. Instead, they reflect the vulnerability of traditional DFS estimators to spectral leakage, noise, and imperfect signal modeling.

In addition to Widar systems, recent methods such as WiTraj [27] and CSI-Difference [12] have also investigated CSI-based trajectory estimation and motion sensing by leveraging Doppler information. In this work, we center our analysis on Widar2.0 and Widar3.0 as representative examples of DFS estimation. These methods embody typical design philosophies and processing pipelines widely adopted in the literature. Our goal is to examine the inherent limitations shared by mainstream DFS estimation approaches and motivate the need for more robust solutions.

2.3 Impact of Relative Orientation in Space

In actual WiFi sensing system deployment, the orientation α and β of the Tx-Rx antenna relative to the target may change between $[0°, 360°]$. To quantify the influence of antenna orientation of Tx-Rx for DFS-based velocity estimation, we first assume that f_D is an unbiased estimate of DFS, and then analyze the gradient changes of speed v and direction θ compared to f_D:

$$\frac{dv}{df_D} = -\frac{c}{f \cdot \cos(\frac{\beta+\alpha}{2} - \theta) \cdot 2\cos(\frac{\beta-\alpha}{2})} \tag{3}$$

$$\frac{d\theta}{df_D} = \frac{c}{v \cdot f \cdot \sin(\frac{\beta+\alpha}{2} - \theta) \cdot 2\cos(\frac{\beta-\alpha}{2})} \tag{4}$$

According to the above formula, assuming that the error of f_D is constant, the estimation error of speed v and direction θ is related to α and β. Fig. 4 shows some simulation results of this correlation, where the closer to red, the larger the estimation error. In particular, for any speed v, the estimation accuracy is worst when $\frac{\beta+\alpha}{2} - \theta = 90°$ or $270°$, and when $\frac{\beta+\alpha}{2} - \theta = 0°$ or $180°$ has the best accuracy. On the contrary, the worst estimation accuracy for direction θ exists when $\frac{\beta+\alpha}{2} - \theta = 0°$ or $180°$, and the best estimation accuracy is at $\frac{\beta+\alpha}{2} - \theta = 90°$ or $270°$. What's more, the average estimation accuracy of direction θ increases with the increase of speed v.

From the above analysis, we can understand that there are significant challenges in DFS-based velocity estimation that remain unresolved. Existing methods fail to adequately address issues related to estimation accuracy and direction dependence. Therefore, developing methods to tackle these issues is crucial for enhancing the effectiveness of WiFi-based velocity estimation systems.

3 Overview

The goal of this paper is that for a given two-dimensional interest zone with range $L_x \times L_y$ and multiple moving targets, *freeDoppler* can accurately estimate the movement velocities (including speed v and direction θ): $\mathbf{V} = \{\mathbf{v_0}, \cdots, \mathbf{v_k}\} = \{(v_0, \theta_0), \cdots, (v_k, \theta_k)\}$ of each target within a time period $[t_0, t_k]$, and further use for WiFi sensing tasks. As shown in Fig. 5, *freeDoppler* consists of four modules: *CSI Generation Model* (CGM), *Velocity Estimation Network* (VEN), CSI processing and trajectory reconstruction.

CGM is used to generate rich simulated CSI. We try to traverse all potential multiple targets' trajectory combinations in the interest zone, and then apply a CGM to each group of trajectories. The simulated CSI generated by CGM is only used for model training.

CSI processing is used to perform band-pass filtering, phase cleaning and amplitude normalization on raw CSI to suppress hardware noise and static path signals, so that the real-world CSI input to VEN is as consistent as possible with the simulated CSI form.

VEN is designed to estimate the velocities for each set of trajectories containing multiple targets. During model training, VEN takes all simulated CSI as input, and outputs the enhanced velocities close to the ideal speeds and directions. During inference, a time sequence of CSI collected from the real-world is fed into VEN, and the velocities are predicted.

Trajectory reconstruction is used to predict the motion trajectories of multiple targets, after we obtain rich velocities predicted by VEN according to real-world CSI.

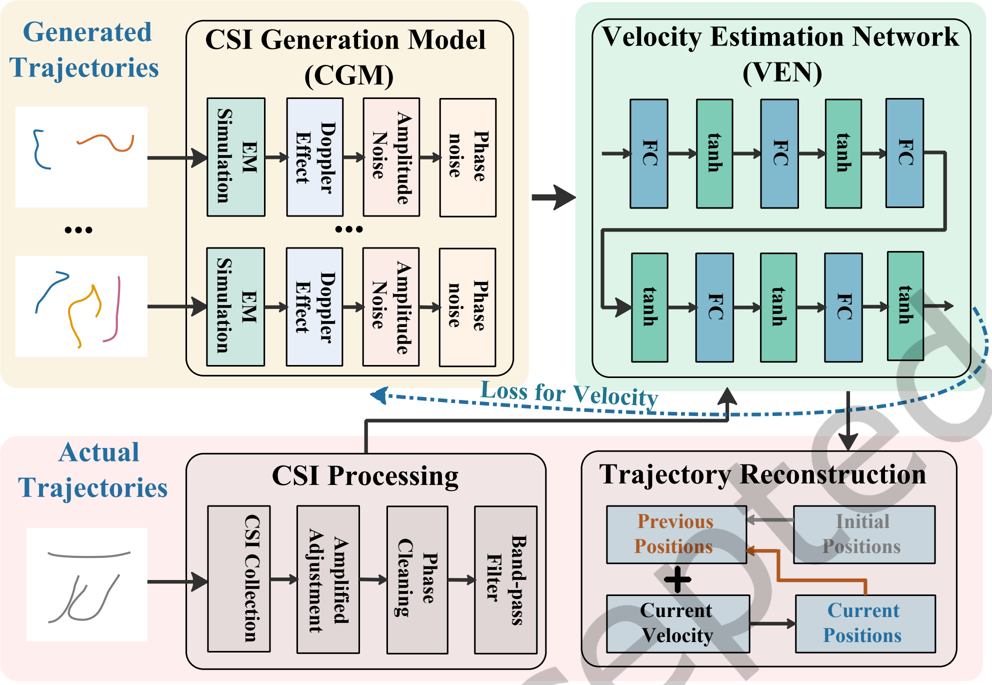


Fig. 5. **Overview of *freeDoppler*: consists of *CSI Generation Model* (CGM), *Velocity Estimation Network* (VEN), CSI processing and trajectory reconstruction.**

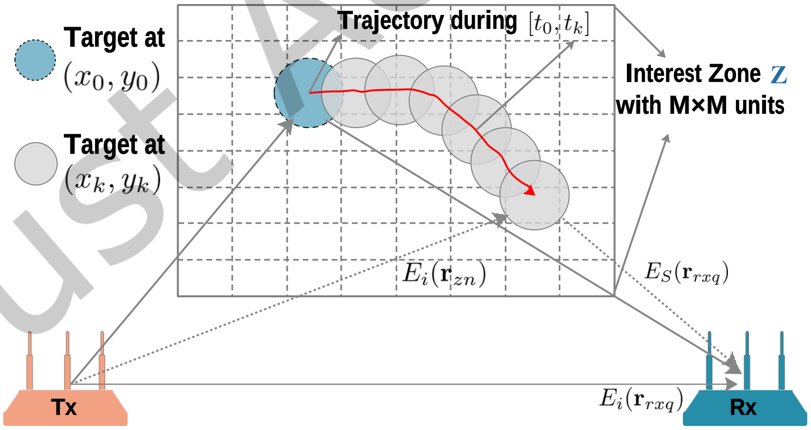


Fig. 6. **Electric field scattering model of *transverse magnetic* (TM) waves propagating in a two-dimensional interest zone. The total electric field E_t at the Rx antennas is the superposition of the incident field E_i excited by the Tx antennas and the scattered field E_s of the target.**

4 System Design

4.1 CSI Generation Model

Traditionally, there are two ways to estimate DFS using WiFi signals: performing FFT on temporal CSI or path parameter estimation considering DFS. However, these solutions have limitations. FFT-based methods are affected by time resolution, frequency resolution, and spectrogram leakage, while path-parameter-based methods are limited by multipath decomposition models and parameter estimation algorithms. In addition, according to the DFS-based velocity estimation principle, the velocity estimation accuracy is affected by the relative orientation of the moving target and WiFi TX-Rx. Even though DFS is an unbiased estimate, when the moving target is located at different relative orientations, the estimated speed and direction exhibit nonlinear errors. In order to obtain the accurate target moving velocities in any time period, we must consider solving the above problems. Fortunately, leveraging learning-based methods is expected to push many traditional limitations in wireless sensing. Therefore, in this paper, we tend to use rich CSI data to learn the DFS-based velocity model and predict velocities.

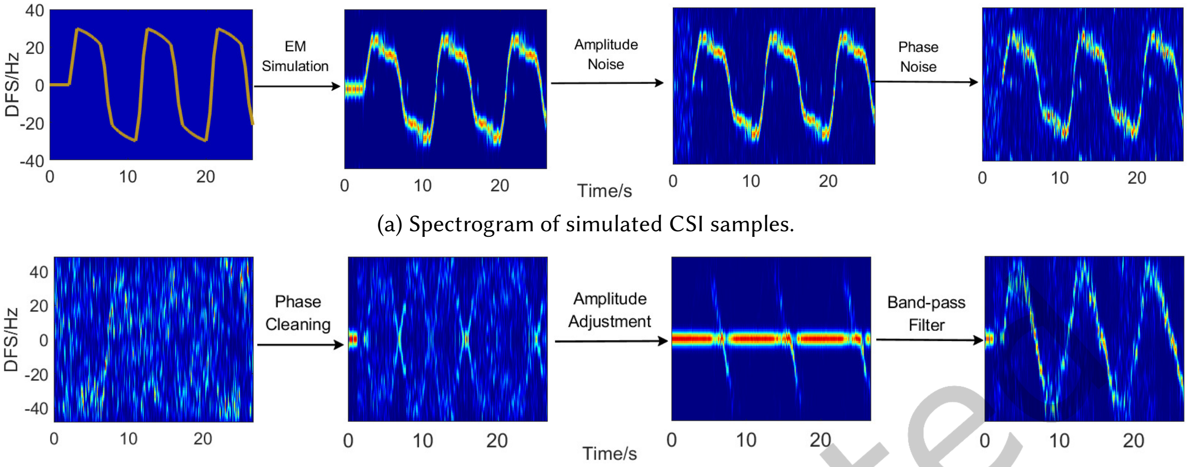
To learn an accurate network model, we need to cover all possible CSI as much as possible. Compared with the time-consuming and labor-intensive real-world CSI collection method, *Radio Frequency* (RF) data generation based on environmental physical models has recently shown great potential [3, 5, 34]. Unlike the ray tracing model, which equates the target to a point, we prefer conditions closer to the real target, such as size. We already know that electromagnetic waves propagating in space satisfy Maxwell's equations [7]. Therefore, we try to build a scattering model that can fully describe signal propagation starting from Maxwell's equations. Similar to [20], we consider the propagation of *Transverse Magnetic* (TM) waves in two dimensions. This situation is common, as shown in Fig. 6, where the electric field is perpendicular to the direction of wave propagation, and the interest zone to be sensed is parallel to the direction of wave propagation. According to the principle of electric field superposition [7], the total field can be regarded as the superposition of the incident field and the scattered field, and the scattered field describes the characteristics of the reflective targets in the environment. Next, we elaborate on how we estimate the scattered field in the environment.

For a WiFi transceiver composed of Tx-Rx, as shown in Fig. 6, the positions of P transmit antennas are $\{\mathbf{r}_{tx1}, \cdots, \mathbf{r}_{txP}\}$, and the positions of Q receive antennas are $\{\mathbf{r}_{rx1}, \cdots, \mathbf{r}_{rxQ}\}$. We assume that the target moves within the interest zone and divides the interest zone into $M \times M$ units, and the positions of the units are $\{\mathbf{r}_{z1}, \cdots, \mathbf{r}_{zM^2}\}$. According to [20], the scattering field at Rx is:

$$
\begin{aligned}
&\mathbf{E}_s = \mathbf{G}_S \cdot \mathbf{G}_Z^{-1} \cdot (\mathbf{E}_t - \mathbf{E}_i) \\
&\mathbf{G}_S(q, n) = k_0^2 \iint_{Z_n} G\left(\mathbf{r}_{rxq}, \mathbf{r}\right) d\mathbf{r} \\
&\mathbf{G}_Z(m, n) = k_0^2 \iint_{Z_n} G\left(\mathbf{r}_{zm}, \mathbf{r}\right) d\mathbf{r} \\
&s.t.\ \ q \in [1, Q], n \in [1, M^2], m \in [1, M^2]
\end{aligned}
\tag{5}
$$

where $\mathbf{G}_S$ and $\mathbf{G}_Z$ are $Q \times M^2$ and $M^2 \times M^2$ dimensions *Green's Coefficient Matrix*, and k_0 is the wave number of the air and Z_n is the n-th unit of the interest zone. In addition, the total electric field $\mathbf{E}_t = E_t(\mathbf{r}_{zi})$ and the incident field $\mathbf{E}_i = E_i(\mathbf{r}_{zi})$ are both M^2-dimensional vectors, the scattered field $\mathbf{E}_s = E_s(\mathbf{r}_{rxi})$ is a Q-dimensional vector. $\mathbf{E}_t$ and $\mathbf{E}_i$ can be obtained by the method of moments. When the target is located at a different position (x_k, y_k), *i.e.*, the units in the interest zone that are stimulated are different, so the $\mathbf{E}_s$ obtained is different.

However, through the scattering field $\mathbf{E}_s$ to characterize moving targets in different positions, two problems should be solved. Firstly, the scattering field $\mathbf{E}_s$ is also related to the static environment and the built-in properties of the targets. Secondly, the influencing factors of real-world CSI include not only dynamic reflection paths



(a) Spectrogram of simulated CSI samples.

(b) Spectrogram of real-world CSI samples.

Fig. 7. **Spectrogram comparison of simulated CSI samples and real-world CSI samples.**

but also static paths and hardware noise. Fortunately, in our problem, the properties of different objects can be approximately constant (such as a human torso). In addition, we can apply a series of processing to real-world CSI to suppress the interference of static paths and hardware noise, the detailed process is shown in Section 4.2. Therefore, we can use $\mathbf{E}_s$ to generate intermediate state CSI that is only related to dynamic reflection paths instead of raw CSI.

To simulate the data used for DFS-based velocity estimation, we generate intermediate state CSI in the continuous time period $[t_0, \cdots, t_k]$ for the target's motion trajectory $\{(x_0, y_0), \cdots, (x_k, y_k)\}$:

$$\mathbf{H}_I = \{\mathbf{E}_s(x_0, y_0), \mathbf{E}_s(x_1, y_1), \cdots, \mathbf{E}_s(x_k, y_k)\} \tag{6}$$

In our problem, the target moves with speed range $[0, 1.5m/s]$ and direction range $[0, 2\pi]$, so we iterate through the combinations of speed and direction at equal time intervals Δt to generate all trajectories within the interest zone. Then, we obtain the DFS caused by moving the target according to Equ. 2. The intermediate state CSI is as follows:

$$\mathbf{H}_D = \sum_{i=0}^{k} \mathbf{H}_I(x_i, y_i) e^{j2\pi f_{D_i}/F_s} \tag{7}$$

where f_{D_i} is the DFS when the target in position (x_i, y_i), F_s is the sample rate. Together, Equ. 5 – Equ. 7 simulate how motion-induced DFS influences the scattered field and are projected onto the received CSI via multipath propagation.

Furthermore, to ensure the quality of simulated CSI, we must also consider the complexity of actual scenarios, such as the presence of multiple targets within the interest zone, the amplitude and phase noise of WiFi signals, and random channels. In this paper, we assume that there are 1 or 2 moving targets in the interest zone. Then we normalize the amplitude to the range (0,1). For the amplitude noise, we add a Gaussian distribution obeying (0,1), while the phase noise is a uniform distribution obeying $[0, 2\pi]$. Therefore, the generated intermediate state CSI becomes:

$$\mathbf{H}_G = \sum_{l=0}^{L} \mathbf{H}_D^l + \mathcal{N} \tag{8}$$

where L is the number of targets, $\mathcal{N}$ is the noise. In particular, we generate $\mathbf{H}_G(q, f_s)$ for each Rx q antenna and each subcarrier f_s. As shown in Fig. 7, we can get the generated simulated CSI.

Therefore, CGM simulates the scattering of EM waves using a formulation derived from Maxwell's equations and Green's function. It captures the phase shifts and amplitude modulations introduced by moving scatterers under various motion and orientation settings. Compared with traditional path-based analytical CSI modeling, CGM retains physical interpretability while providing diverse, noise-free intermediate-state CSI samples with accurate Doppler characteristics. This makes it ideal for generating large-scale training data for learning-based systems, without relying on heuristic assumptions or manually labeled data.

4.2 CSI Processing

The raw CSI collected by the WiFi *Network Interface Card* (NIC) contains static paths, dynamic reflection paths, and hardware noise. To estimate the velocity of target motion, we need to suppress the influences of hardware noise and static paths. Furthermore, it is almost impractical to generate simulated CSI containing variable and unknown hardware noise and static paths, so *freeDoppler* considers processing raw CSI and does not consider these influences when generating CSI. We know that the CSI matrix $\hat{\mathbf{H}}$ collected from the network card contains multiple antennas and multiple subcarrier information. In this paper, we consider the number of Tx antenna to be one, the number of Rx antennas to be three, and the number of subcarriers to be N_s. In general, CSI can be expressed as the superposition of responses from each path [17]:

$$\begin{aligned} \hat{\mathbf{H}}(n,f,t) &= (\mathbf{H}_s(n,f) + \mathbf{H}_d(n,f,t)) \cdot e^{j\Delta(n,f,t)}, \\ \mathbf{H}_d(n,f,t) &= \sum_{l=0}^{L} \gamma_l(t) e^{j2\pi \int_{-\infty}^{t} f_{D_l}(u)du} \end{aligned} \tag{9}$$

where n is antenna index, f is subcarrier frequency, t is time index, $\mathbf{H}_s$ is the CSI of static paths and $\mathbf{H}_d$ is the CSI of dynamic paths, γ_l and f_{D_l} is the attenuation and DFS of moving target l, Δ is the unknown phase error.

To suppress these influences, we process raw CSI as shown in Fig. 7b. First, to clean phase error, we use the principle of similar phase noise of multiple antennas, select one antenna as the reference antenna n_0, and multiply the CSI of other antennas with the conjugate of the reference antenna:

$$\mathbf{H}(n_m, f, t) = \hat{\mathbf{H}}(n_m, f, t) \cdot \hat{\mathbf{H}}^*(n_0, f, t), n_m = n_1, n_2 \tag{10}$$

Second, to eliminate the by-product terms caused by conjugate multiplication, *i.e.*, the two products of the static path of one antenna and the moving path of the other antenna [13], we adjust the CSI amplitude:

$$\begin{cases} |\mathbf{H}(n_1, f, t)| = \left|\hat{\mathbf{H}}(n_1, f, t)\right| - \gamma_1 \\ |\mathbf{H}(n_2, f, t)| = \left|\hat{\mathbf{H}}(n_2, f, t)\right| + \gamma_2 \end{cases} \tag{11}$$

where γ_1 and γ_2 are two constants. In this way, by-product terms are greatly suppressed, while terms containing the correct DFS are greatly enhanced. Third, we apply a band-pass filter to eliminate static paths, and we normalize the amplitude to the range (0,1). From Fig. 7, it can be seen that the simulated generated CSI and the processed real CSI are similar.

4.3 Velocity Estimation Network

To obtain accurate continuous velocities under any motion trajectory, we use a learning-based scheme and use the generated simulated CSI to train the VEN of *freeDoppler*. VEN processes continuous complex-valued CSI sequences, fully leveraging the amplitude and phase information inherent in the data. Unlike traditional deep learning models that rely only on the magnitude of CSI for velocity estimation, VEN uses both magnitude and phase, which is crucial for capturing Doppler-induced variations. To achieve this, VEN employs a structured

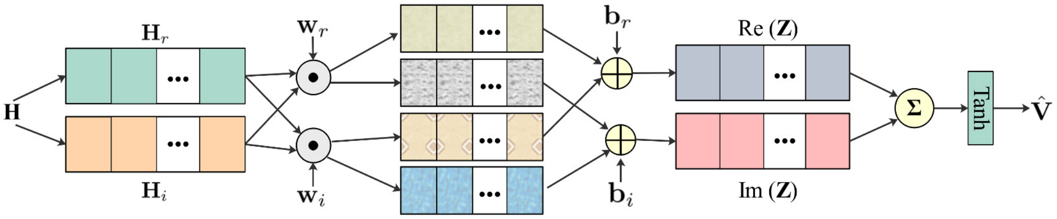


Fig. 8. **Complex-valued neurons of VEN: These neurons process both the real and imaginary parts of the input CSI data, enabling the network to capture DFS in both amplitude and phase. The complex weights and biases allow the network to effectively learn from the full complex-valued input, improving velocity estimation accuracy.**

architecture of fully connected (FC) layers that separate and process the real and imaginary parts of the CSI input, ensuring that the network effectively captures the critical information for accurate velocity estimation. As shown in Fig. 5, VEN consists of five *Fully Connected* (FC) layers and *Hyperbolic Tangent* (Tanh) activation function. During the model training process, the input of VEN is the simulated CSI **H** generated by CGM from any trajectory combination and the ground-truth velocities **V**, and the output is a series of predicted velocities $\hat{\mathbf{V}}$. To handle the complex-valued **H**, we represent **H** as follows:

$$\mathbf{H} = \mathbf{H_r} + i * \mathbf{H_i} \tag{12}$$

where $\mathbf{H_r}$ and $\mathbf{H_i}$ are the real part and imaginary part of complex-valued **H**.

Then, we design complex-valued FC neurons as shown in Fig. 8, and the learned representations are as follows:

$$\begin{aligned} \mathbf{Z} &= \mathbf{W} * \mathbf{H} + \mathbf{b} \\ &= (\mathbf{W_r} + i\mathbf{W_i}) * (\mathbf{H_r} + i\mathbf{H_i}) + (\mathbf{b_r} + i\mathbf{b_i}) \\ &= (\mathbf{W_r}\mathbf{H_r} - \mathbf{W_i}\mathbf{H_i} + \mathbf{b_r}) + i(\mathbf{W_r}\mathbf{H_i} + \mathbf{W_i}\mathbf{H_r} + \mathbf{b_i}) \end{aligned} \tag{13}$$

where $\mathbf{W} = \mathbf{W_r} + i\mathbf{W_i}$ and $\mathbf{b} = \mathbf{b_r} + i\mathbf{b_i}$ are the complex-valued weight and bias. Next, **Z** is input to a Tanh activation function:

$$\hat{\mathbf{V}} = \mathrm{Tanh}(\mathbf{Z}; \Theta_{tanh}) \tag{14}$$

where Θ_{tanh} is the parameter set of Tanh. In particular, we use L_2 as the loss function:

$$\begin{aligned} \mathcal{L}_v &= ||\hat{\mathbf{V}} - \mathbf{V}||_2 \\ &= ||\hat{\mathbf{v}}_r - \mathbf{v}_r||_2 + ||\hat{\mathbf{v}}_i - \mathbf{v}_i||_2 \end{aligned} \tag{15}$$

where $\mathbf{v}_r$ and $\mathbf{v}_i$ are the real and imaginary parts of the velocities ground-truth obtained by synthesizing the speed **v** and direction θ into complexed-value, and $\hat{\mathbf{v}}_r$ and $\hat{\mathbf{v}}_i$ are the real and imaginary parts of the velocities estimation. Finally, we minimize $\mathcal{L}_v$ during model training, and feed the processed real-world CSI to VEN during inference to obtain accurate velocity sequences. The training of VEN utilizes high-quality simulated CSI generated by CGM, allowing the network to learn nuanced relationships between CSI variations and target motion patterns. By incorporating complex-valued representations directly into its structure, VEN avoids the information loss associated with traditional amplitude-only or phase-only approaches. The output of VEN includes both speed and direction components for each target, calculated with high precision across varying orientations and trajectories.

In addition, according to Equ. 2, to capture the complete target motion information and obtain the speed v and direction θ, we need to use at least two pairs of WiFi transceivers and collect CSI to estimate two DFS f_{D_1} and f_{D_2} respectively. Therefore, in this paper, we set up two pairs of mutually perpendicular WiFi transceivers to collect raw CSI $\mathbf{H}_{rx1}$ and $\mathbf{H}_{rx2}$ from the real-world, and generate simulated CSI $\mathbf{H}_{G,rx1}$ and $\mathbf{H}_{G,rx2}$ based on CGM.

During the training VEN and inference process, we input the intermediate CSI of two pairs of WiFi transceivers $[\mathbf{H}_{rx1}, \mathbf{H}_{rx2}]$ and $[\mathbf{H}_{G,rx1}, \mathbf{H}_{G,rx2}]$ into VEN to obtain the estimated velocity sequence.

Therefore, unlike conventional Doppler estimation methods that rely on FFT analysis or path parameter estimation, which are sensitive to noise and rely on fixed assumptions, VEN learns to extract velocity-related features directly from raw CSI. It uses complex-valued inputs that preserve both amplitude and phase, enabling it to capture Doppler shifts more robustly under dynamic multipath conditions. This design overcomes the limitations of prior learning-based approaches that only use CSI magnitude and struggle with orientation-induced variance.

4.4 Trajectory Reconstruction

After accurately estimating the velocity sequence $\mathbf{V} = \{(v_0, \theta_0), \cdots, (v_k, \theta_k)\}$, we can use it for further trajectory reconstruction. First, to improve the speed prediction to obtain a more accurate reconstructed trajectory, we optimize the predicted velocity sequence based on two observations, *i.e.*, the rate of change in speed (v) between consecutive velocity estimates should not exceed a predefined threshold, Δv, to avoid unrealistic jumps in speed; the rate of change in direction (θ) should also remain within a defined limit, $\Delta\theta$, to prevent erratic directional shifts. These constraints are incorporated into the optimization process to refine the velocity sequence:

$$\min \sum_{i=1}^{k} (\hat{v}_k - \hat{v}_{k-1})^2, \quad s.t. \quad |\hat{v}_k - \hat{v}_{k-1}| \leq \Delta v \tag{16}$$

$$\min \sum_{i=1}^{k} (\hat{\theta}_k - \hat{\theta}_{k-1})^2, \quad s.t. \quad |\hat{\theta}_k - \hat{\theta}_{k-1}| \leq \Delta\theta \tag{17}$$

By optimizing these constraints, we reduce the likelihood of unrealistic velocity fluctuations, ensuring a smoother and more consistent reconstructed trajectory.

Second, we recover the trajectory from velocity only, so we need an initial position (x_0, y_0). This assumption is reasonable, and there are actually many ways to calibrate this initial position. For example, we can use landmarks as initial positions. Or similar to Widar [17], we can iteratively search the entire tracking area with first coarse resolution and then fine resolution, and use the position with the smallest trajectory fitting error as the initial position. Finally, once the initial position (x_0, y_0) is known, we can fit the target motion trajectory according to the following:

$$\hat{x}_k = x_0 + \sum_{i=1}^{k} \hat{v}_i \cos(\hat{\theta}_i)\Delta t \tag{18}$$

$$\hat{y}_k = y_0 + \sum_{i=1}^{k} \hat{v}_i \sin(\hat{\theta}_i)\Delta t \tag{19}$$

where Δt is the time interval of two consecutive velocities. The motion trajectory $(\hat{x}_k, \hat{y}_k)$ is iteratively updated based on the predicted velocities and directions. This method ensures that the reconstructed trajectory closely follows the estimated motion dynamics, with the temporal integration of velocity allowing for a continuous path to be formed. Together, Equ. 16 – Equ. 19 form a complete pipeline: the raw velocity predictions are first smoothed to ensure temporal consistency, and then integrated from a known starting position to reconstruct the 2D trajectory.

5 Implementation

Hardware. We build the real-world CSI collection platform using three COTS *Industrial Personal Computers* (IPCs), each equipped with an Intel 5300 WiFi NIC. One IPC is used as Tx equipped with one antenna, and the remaining two IPCs as Rx equipped with three antennas. The distance between the three receiving antennas is 5.9 cm. The operating system of IPC is Ubuntu 20.04, and we install PicoScenes to collect CSI readings [8]. The NIC is set up to operate on a channel with a center frequency of 5.19 GHz and a bandwidth of 40MHz. We set all Rx to monitor mode, and inject Tx to broadcast at a rate of 100 packets per second. *freeDoppler* subsequent processing is on an computer with Intel Core i7-10700 CPU 4.80GHz and RAM 48GHz.

Software. We use the Linux shell script to collect CSI from the real-world, and the entire signal processing is performed on MATLAB R2020b. The code generated by CGM and simulated CSI is Julia 1.9.3 [1]. We use Python 3.12 and the PyTorch library for model implementation, model training, and model inference. VEN uses randomly generated simulated CSI for offline training, and a total of 1000 epochs are used to train the VEN. During training, the Adam optimizer is used with a learning rate of 0.001.

Baseline methods. In this paper, we use Widar 2.0 [18] and Widar 3.0 [35] as baselines, which are two different schemes for estimating DFS using WiFi signals. Specifically, Widar 2.0 uses the EM algorithm to jointly estimate the path parameters including DFS, and Widar 3.0 applies time-frequency analysis to obtain DFS profiles on CSI. It is noted that Widar 2.0 and Widar 3.0 perform velocity estimation based on Euq. 2, as they claimed in their work. In addition, to obtain the trajectory, Widar 2.0 also obtains AoA, ToF, and attenuation, while Widar 3.0 uses DFS and the target initial position.

Evaluation metrics. We employ three metrics to comprehensively evaluate the performance of *freeDoppler* in velocity estimation and trajectory reconstruction:

- *Relative Speed Error (%)*: Defined as the average percentage error between the estimated and ground-truth speeds, this metric reflects the system's accuracy in capturing Doppler shifts under different motion patterns.
- *Direction Error (°)*: Calculated as the average absolute angular difference between predicted and true motion directions. This metric evaluates the precision of direction estimation, which is crucial for accurate trajectory inference.
- *Location Error (cm)*: Measured as the average Euclidean distance between the reconstructed and ground-truth positions over time. This end-to-end metric quantifies the effectiveness of the system in real-world motion tracking applications.

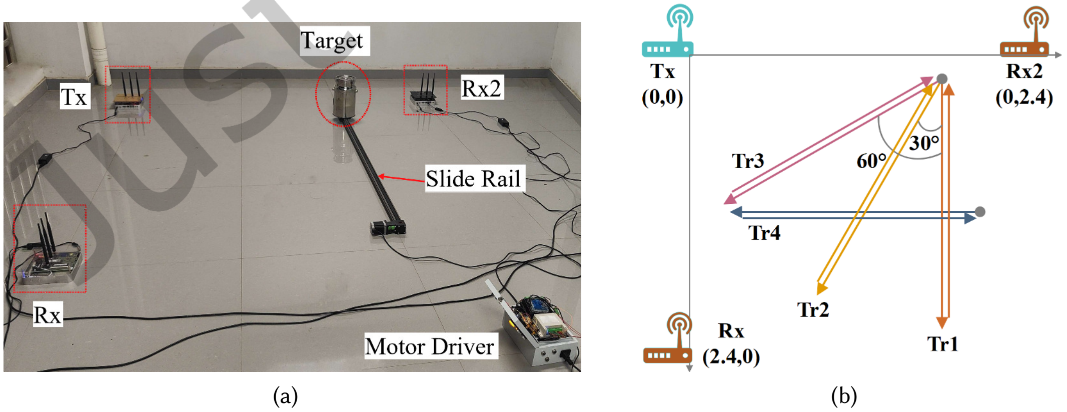


Fig. 9. **Experimental setup for evaluating velocity estimation accuracy:** (a) Environment 1. (b) Four trajectories with three speeds.

6 Evaluation

6.1 Experimental Setup and Dataset

Experimental setup for evaluating velocity estimation. We first evaluate the velocity estimation performance of *freeDoppler* using a controlled setup shown in Fig. 9. The experiment is conducted in a 2.4m × 2.4m indoor area with a single Tx and two Rxs. The two Rxs are placed 2.4 m apart along a horizontal line, and the Tx is positioned 2.4m in front of their center, forming an approximately orthogonal cross-layout to improve angular diversity. Each Rx is equipped with 3 antennas, and CSI is sampled at a rate of 100Hz using 30 subcarriers per antenna pair.

The target is a cylindrical metal object (18cm diameter) mounted on a motorized slide rail, which enables programmable motion with precise control of direction and velocity. We configure the target to move back and forth five times along four different trajectories (Fig. 9b) at speeds of 26cm/s, 38cm/s, and 50cm/s, yielding 24 combinations of motion parameters. This setup ensures accurate alignment between the physical movement and recorded CSI, allowing fine-grained analysis of the Doppler signatures.

For training purposes, simulated CSI is generated over the same 2.4m × 2.4m space, discretized into a 240 × 240 spatial grid. The target is modeled as a reflective object, and its motion along randomly perturbed trajectories is simulated using CGM. We generate over 5,000 synthetic samples covering a wide range of speeds and angles, ensuring VEN is trained with diverse and physically grounded CSI-Doppler mappings.

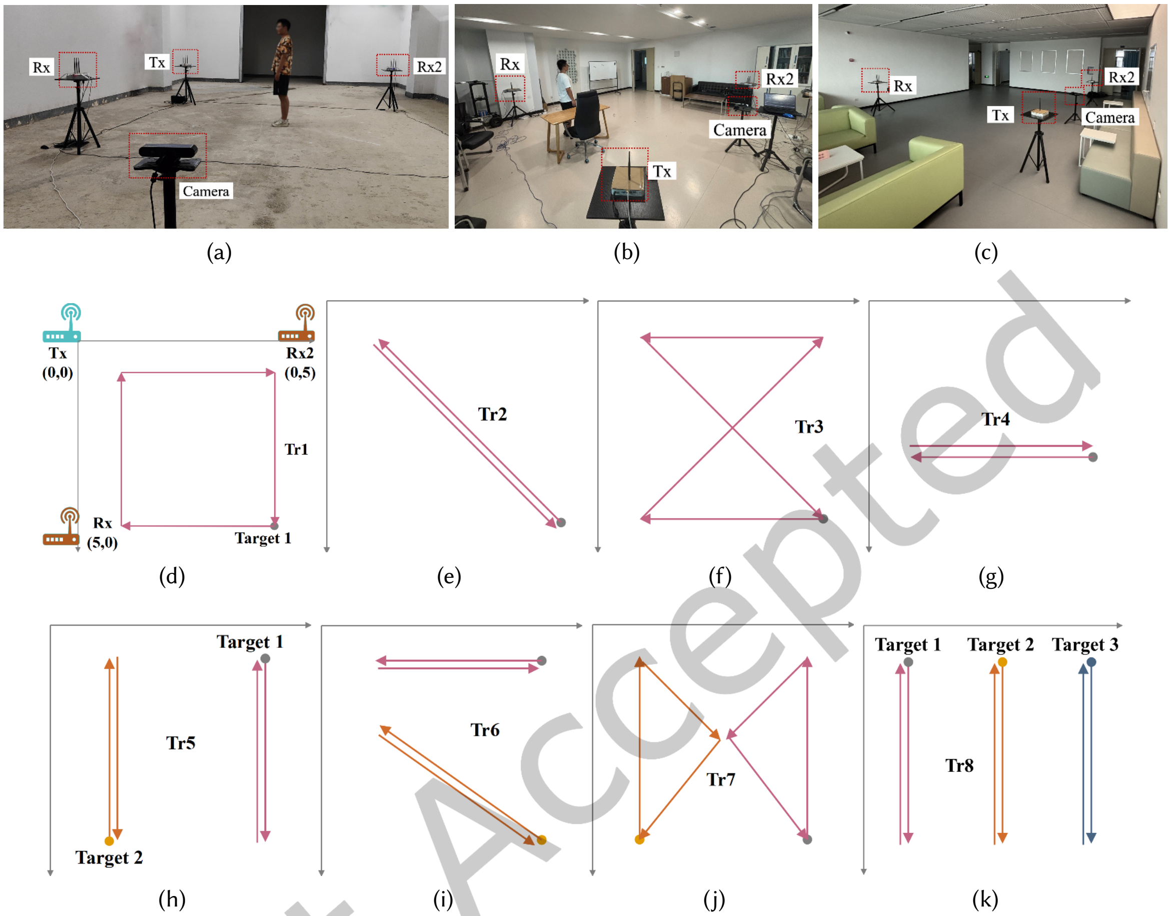


Fig. 10. **Experimental setup for evaluating human tracking accuracy:** (a)-(c) Environment 2-4 (env 2, env 3, env 4). (d)-(g) Single target with four trajectories and two speeds. (h)-(j) Two targets with three trajectories and two speeds. (k) Three targets with one trajectory and three speeds.

Experimental setup for evaluating human tracking. To evaluate human trajectory tracking, we deploy a single Tx and two Rxs in three different 5m × 5m indoor environments as shown in Fig. 10a - Fig. 10c, with a height of 1.2m and the same angular layout as the velocity experiment. The targets are three individuals of different heights to simulate inter-subject variation. Each subject is instructed to walk along predefined trajectories shown in Fig. 10d through Fig. 10k, including both straight and turning paths. Each trajectory is repeated 5–10 times at different walking speeds (slow, normal, brisk), covering a speed range of approximately 0.5–1.5m/s. The Rxs continuously collect CSI during all trials at a 100Hz packet rate.

To obtain accurate ground-truth trajectories for comparison, we use the Astra Pro Plus depth camera, which captures synchronized video streams and depth information of the moving targets. Using the YOLOv8-pose model [9], we estimate the position of individuals from the video streams. The camera's intrinsic parameters and depth data are then used to calculate the 3D coordinates of the movements, providing highly accurate ground-truth tracking trajectories.

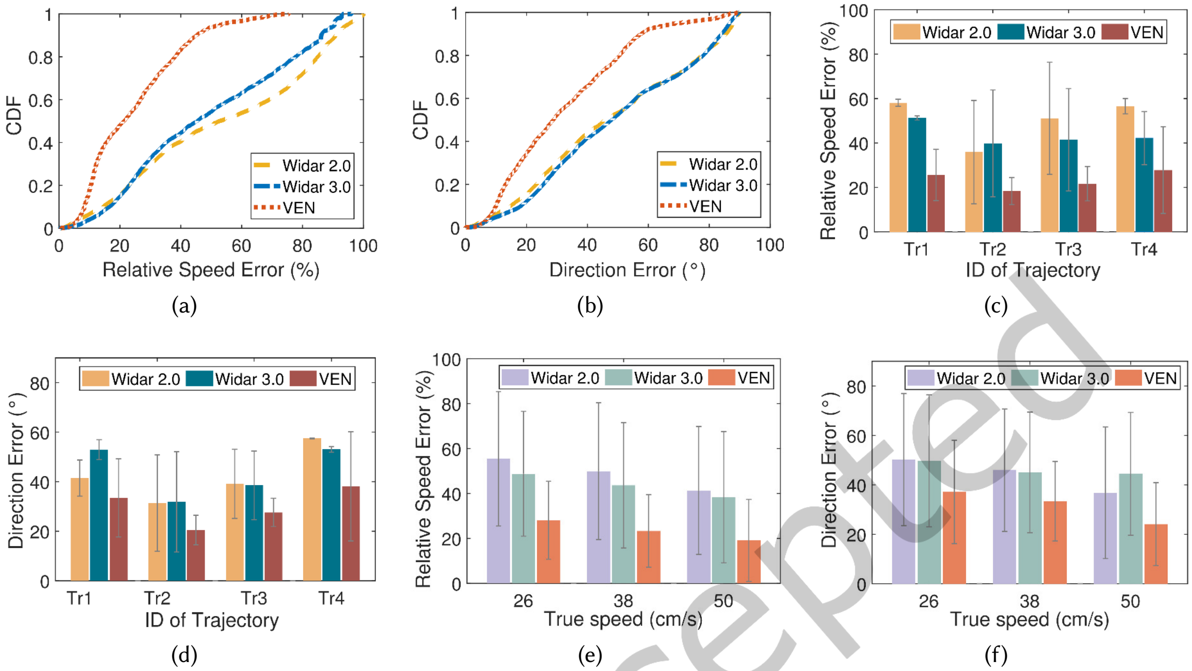


Fig. 11. **Accuracy of velocity estimation at different conditions:** (a) Overall speed estimation error. (b) Overall direction estimation error. (c) Speed and (d) direction estimation errors of different trajectories. (e) Speed and (f) direction estimation errors of different true speeds.

Simulated CSI for the human tracking setting is generated using CGM over a 5m × 5m space divided into 50 × 50 grid units (10cm resolution). Human targets are modeled as cylinders with a 40cm diameter. Their motion patterns are derived from real walking trajectories, incorporating natural variation in speed and direction transitions. This simulated data supports VEN training under more complex conditions and augments real data for trajectory reconstruction evaluation.

6.2 Accuracy of Velocity Estimation

Overall performance. To evaluate the overall performance of VEN on velocity estimation, we compare it against two baseline models, Widar 2.0 and Widar 3.0, both of which use DFS for velocity estimation. For each model, we estimate the speed and direction of the moving target and analyze the results. The performance metrics include relative speed estimation error and direction estimation error, as shown in Fig.11a and Fig.11b. The results indicate that VEN outperforms both baselines across all evaluated metrics. VEN achieves a median relative speed estimation error of 21% (*i.e.*, 7.98 cm/s for speed estimation), significantly lower than the 51% error of Widar 2.0 and the 44% error of Widar 3.0. Similarly, VEN's median direction estimation error is 28°, outperforming the 46° and 48° errors of Widar 2.0 and Widar 3.0, respectively. VEN's improved performance can be attributed to its ability to leverage both the amplitude and phase information of CSI, providing more comprehensive learning of DFS. Additionally, the CGM-generated simulated CSI enhances the training process by covering a broader range of motion scenarios, enabling VEN to generalize better to unseen trajectories. Overall, these findings validate the effectiveness of VEN for fine-grained velocity estimation and highlight its advantages over traditional methods like Widar 2.0 and Widar 3.0.

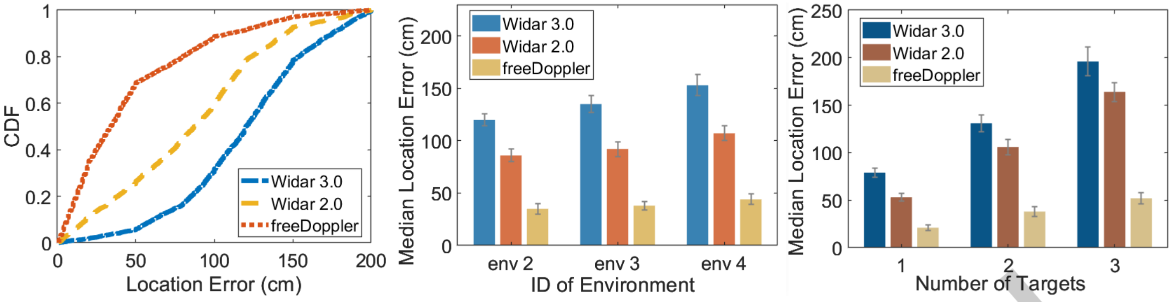
Impact of different trajectories. As introduced in Sec.2.3, when the target moves in different orientations relative to the transceivers, the accuracy of DFS-based velocity estimation exhibits nonlinear measurement errors due to variations in Doppler frequency shifts. To evaluate VEN's robustness under such conditions, we conducted experiments where the target moved along four different predefined trajectories (Tr1, Tr2, Tr3, and Tr4). The results are shown in Fig.11c and Fig. 11d. Each trajectory introduces distinct challenges for velocity estimation. For example, Tr1 and Tr4 involve motion paths that are less favorable for capturing Doppler variations due to reduced angular coverage relative to the transceivers. In contrast, Tr2 and Tr3 are more favorable, as both Rx1 and Rx2 can simultaneously capture richer motion information from different orientations. The results indicate that VEN consistently achieves lower relative speed estimation errors and direction estimation errors across all four trajectories compared to Widar 2.0 and Widar 3.0. Specifically, the errors for Tr2 and Tr3 are noticeably smaller, demonstrating VEN's ability to leverage multi-receiver inputs effectively to improve estimation accuracy. For the more challenging trajectories Tr1 and Tr4, VEN still outperforms the baselines, highlighting its robustness and adaptability to less optimal sensing conditions.

Impact of different speeds. In addition to trajectory variations, we also evaluated VEN's performance under different target movement speeds, as mentioned in Section 2. The relationship between speed and direction in DFS-based velocity estimation is interdependent, making it essential to assess VEN's robustness across a range of speeds. We tested target speeds of 26 cm/s, 38 cm/s, and 50 cm/s, and the results are shown in Fig.11e and Fig.11f. The results demonstrate that VEN consistently achieves lower relative speed estimation errors and direction estimation errors compared to Widar 2.0 and Widar 3.0 across all tested speeds. Interestingly, a notable trend emerges: as the target speed increases, VEN's velocity estimation errors decrease. For example, at 50 cm/s, the estimation error is significantly smaller than at 26 cm/s. This improvement can be attributed to the higher motion frequencies captured by the transceivers at faster speeds, which enhances the Doppler shift information and reduces ambiguity in velocity estimation. This observation aligns with our analysis in Sec. 2. Moreover, VEN's ability to handle lower speeds effectively also highlights its versatility, as slower motion often results in weaker Doppler variations that are harder to distinguish. Even under these challenging conditions, VEN maintains superior performance compared to Widar 2.0 and Widar 3.0, demonstrating its robustness and reliability across a wide range of speeds.

6.3 Accuracy of Human Tracking

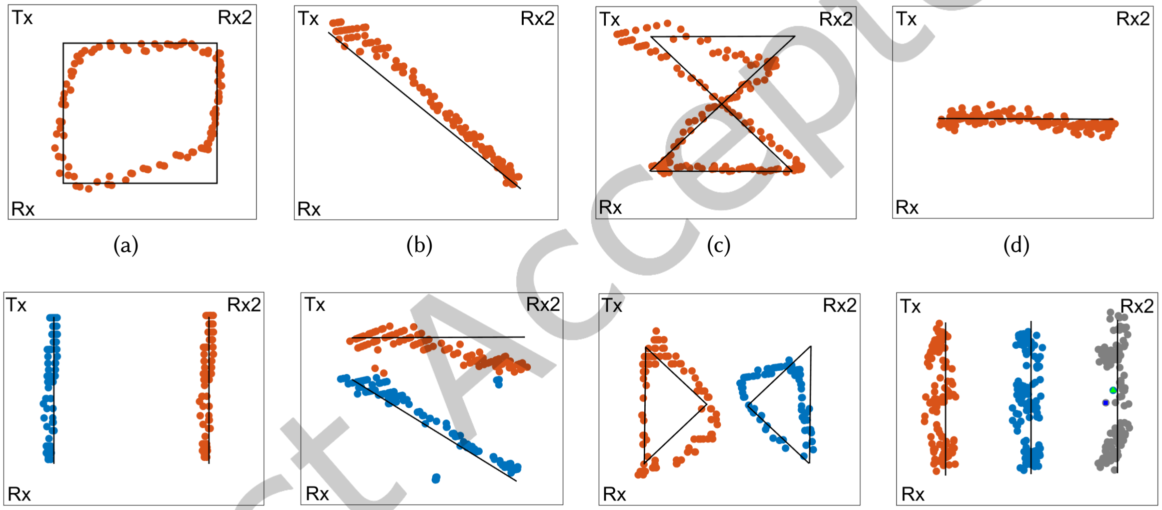
Overall performance. To evaluate the overall performance of *freeDoppler* for human tracking, we preprocess all collected CSI data to remove noise and irrelevant interference before applying Widar 2.0, Widar 3.0, and *freeDoppler* for trajectory estimation. The estimated trajectories are then compared to the ground-truth 3D trajectories obtained from the depth camera, with results presented in Fig. 12a. The results demonstrate that *freeDoppler* achieves superior tracking accuracy compared to the baseline methods. Specifically, *freeDoppler* records a median location estimation error of 35 cm, which is significantly lower than the 86 cm error of Widar 2.0 and the 120 cm error of Widar 3.0. This improvement can be attributed to the enhanced capability of *freeDoppler* to utilize both the amplitude and phase information from the CSI, as well as its robust training on diverse simulated scenarios.

Impact of different environments. We evaluate *freeDoppler* in three distinct indoor environments as shown in Fig. 10a-Fig. 10c. The result is shown in Fig. 12b, despite the increased complexity and variability of these settings, *freeDoppler* consistently achieves a median location error of less than 50 cm, indicating robust tracking performance. Moreover, in all three environments, the median location errors are significantly lower than those



(a) (b) (c)

Fig. 12. **Accuracy of human tracking:** (a) Overall human tracking error. (b) Human tracking error in different environments. (c) Human tracking error in different numbers of targets.



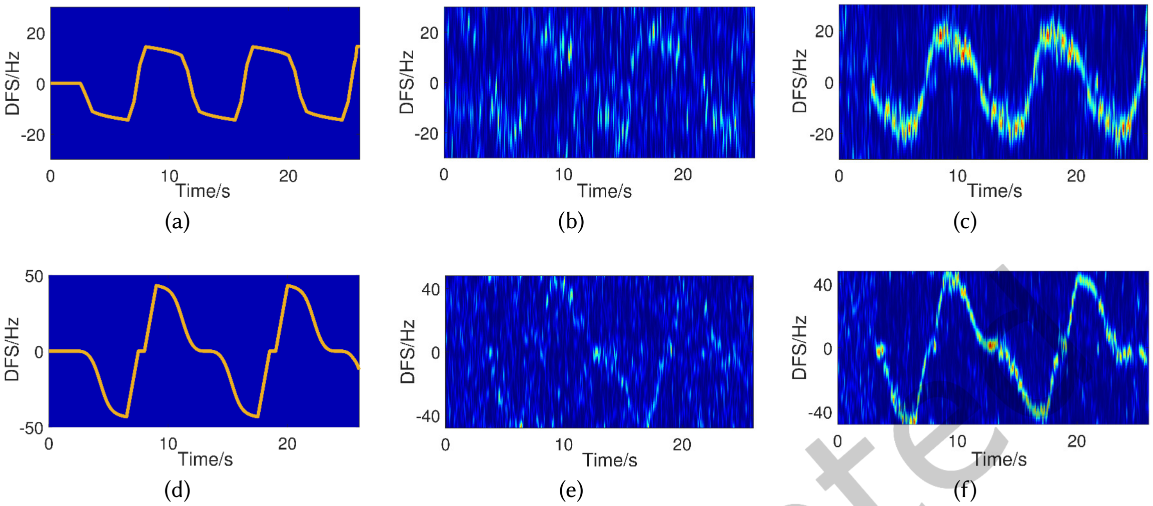
(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 13. **Human tracking of different trajectories:** (a)-(d) Human tracking of the single target with four different trajectories, (e)-(g) human tracking of two targets with three different trajectories, (h) human tracking of three targets with one trajectory, where the black lines represent the ground truth, the orange points represent the predicted trajectories of target 1, the blue points represent the predicted trajectories of target 2, and the gray points represent the predicted trajectories of target 3.

of Widar2.0 and Widar3.0. These findings confirm that *freeDoppler* maintains high accuracy even in diverse real-world scenarios, making it highly suitable for deployment across a wide range of indoor applications.

Impact of different targets' numbers. We also evaluate *freeDoppler*'s performance with varying numbers of moving targets, including single-target, two-target, and three-target. The result is shown in Fig. 12c. For all cases, the median location errors of *freeDoppler* are significantly lower than those of Widar2.0 and Widar3.0. For single-target, the accuracy is particularly high, demonstrating the system's ability to precisely follow even



(a) (b) (c)

(d) (e) (f)

Fig. 14. **Visualization comparison of DFS spectrogram simulation results:** (a) and (d) are the ground truth. (b) and (e) are the simulation results using ray tracing. (c) and (f) are the simulation results using CGM.

complex motion paths. However, in the two-target and three-target cases, *freeDoppler* shows a slight reduction in accuracy due to increased interference between targets. For example, when targets move closely together or cross paths, distinguishing their trajectories becomes more challenging. It is worth mentioning that *freeDoppler* needs to have prior knowledge of the number of targets, which may limit the wider applications of *freeDoppler*. We consider exploring more accurate velocity sensing of unknown multiple targets in future work.

Impact of different trajectories. To further evaluate the robustness of *freeDoppler*, we analyze its performance when the target moves along different trajectories, as shown in Fig.13a and Fig.13g. The trajectories include a mix of straight lines, curves, and oscillatory paths to reflect various real-world movement patterns. Despite the differences in target motion trajectories, *freeDoppler* accurately tracks human movements, with errors remaining relatively low across all tested scenarios.

6.4 DFS Spectrogram Generation Visualization

We also evaluate the performance of the CGM model in generating intermediate-state CSI, with a particular focus on the DFS spectrogram. The simulation setup consists of a transmitter and two receivers to capture the reflected signals. The targets move with predefined speeds and directions, and their scattering properties are modeled in the CGM based on Maxwell's equations. For the ray tracing model, the signal propagation paths are calculated assuming simplified reflections without considering complex interactions. Both models generate DFS spectrograms at consecutive time intervals and use the same environment and target parameters as input.

To benchmark its performance, we compare it with the traditional ray tracing model. For this evaluation, we set up a 5m×5m 2D region of interest with specific target trajectories and generate DFS spectrograms using both models. The simulations were repeated with different target speeds (0.4m/s, 0.8m/s, 1.0m/s, 1.3m/s), orientations (30°, 60°, 90°, 120°), and initial positions ([1,1], [3,0], [4,5], [4,2]) to ensure a comprehensive comparison. Two representative simulation results are shown in Fig. 14. As can be seen, the DFS spectrograms generated by CGM clearly show smoother and more continuous Doppler frequency changes over time. In contrast, spectrograms generated by ray tracing models often exhibit abrupt transitions and artifacts, especially in scenarios involving multipath effects or rapid direction changes. By visually comparing the results, it can be observed that the spectrograms consistently generated by CGM better reflect the characteristics of real CSI data collected under similar conditions. This includes accurately capturing the main Doppler shifts as well as subtle changes caused by target relative motion and environmental interactions. The results highlight the reliability of CGM and its ability to generate high-quality DFS spectrograms suitable for training downstream models.

To quantitatively evaluate the similarity between the CGM-generated spectrograms and the measured spectrograms, we adopt the Pearson correlation coefficient as the metric. For each pair of simulated and measured spectrograms, we compute the frame-wise Pearson correlation coefficient r_t along the frequency dimension and take the average over all frames as the similarity score for that sample:

$$\bar{r} = \frac{1}{T}\sum_{t=1}^{T} r_t.$$

To obtain robust statistical summaries, we calculate $\bar{r}$ for multiple experimental configurations (different speeds, directions, and initial positions) and perform averaging and confidence interval estimation in the Fisher z-space before transforming back to r for reporting the final mean and 95% confidence interval. For comparison between CGM and the ray-tracing baseline, we use a paired test (paired t-test in Fisher z-space) to assess the statistical significance of their differences under each configuration. Results show that CGM consistently achieves significantly higher spectrogram similarity than the baseline: CGM: mean $\bar{r} = 0.85$ [0.82, 0.88]; ray-tracing: mean $\bar{r} = 0.62$ [0.57, 0.67]; paired t-test $p < 0.001$, statistically confirming the superior fidelity and coverage of CGM. In addition, we acknowledge the limitations of relying on a single similarity metric. In future work, we will investigate more rigorous and diversified fidelity and diversity measures (e.g., spectrogram-level structural similarity metrics, trajectory matching scores), and explore adversarial training or domain-adaptation techniques further to narrow the gap between simulated and measured CSI.

7 Discussions and Future Work

Adaptation to complex scenarios. Currently, *freeDoppler* is mainly evaluated in controlled environments with a limited number of moving targets and predictable motion patterns. However, there are also many complex scenarios in real-world applications, such as smart homes and markets. The ability of *freeDoppler* to accurately track multiple targets simultaneously may be affected by the increase in the number of obstacles and unpredictable target behavior. In addition, the dynamic interactions between targets pose additional challenges for accurate velocity estimation. To address these issues, future work would explore the integration of reinforcement learning (RL) frameworks, which would enable the system to adapt to changing environmental conditions. Reinforcement learning algorithms can help the model learn to deal with new, unseen obstacles and interference, making it more robust in complex environments. In addition, adversarial training techniques would be used to simulate complex noise patterns and unexpected environmental changes, enhancing the system's ability to cope with real-world changes.

Computational complexity and real-time feasibility. The computational efficiency of *freeDoppler* is essential for practical deployment, especially in real-time or resource-constrained environments. The system comprises two core components: CGM and VEN. CGM, designed for offline data generation, involves physics-based simulations rooted in Maxwell's equations and electromagnetic scattering theory. While computationally intensive, CGM operates exclusively during the training phase and does not affect online inference. In contrast, VEN is designed for runtime velocity and direction estimation. Its compact architecture allows efficient inference, making it suitable for integration into real-time pipelines. Nevertheless, the feasibility of real-time performance depends on system scale, target density, and hardware capabilities. To further enhance deployment efficiency, future work will explore lightweight model optimization (e.g., quantization or pruning), as well as edge computing and distributed inference frameworks that can reduce latency and support high-throughput applications in dynamic environments.

Transceiver layout and receiver configuration. In real-world environments, the positions of the transmitter and receivers may be constrained by spatial layout, furniture, or human movement patterns, making it difficult to always maintain optimal placement. The performance of velocity estimation is sensitive to the geometric layout of transceivers, as it determines the diversity and quality of observable Doppler shifts. In particular, layouts with orthogonal Tx-Rx pairs (e.g., forming a cross shape) are expected to provide better angular coverage and enable more accurate direction estimation, while narrow-angle configurations may lead to degraded performance due to limited directional resolution. Moreover, the number of receivers plays a crucial role in the richness of spatial information. A larger number of Rx nodes can potentially offer more diverse signal perspectives, improving the system's ability to distinguish complex motion patterns and to handle occlusion or multipath effects in NLoS environments. While our current implementation uses two receivers, future extensions could explore adaptive receiver configurations, including three or more Rx units or dynamic reconfiguration mechanisms, to further improve robustness and scalability in larger or cluttered spaces.

8 Related Work

8.1 Doppler Frequency Shift for WiFi Sensing

Due to the fine-grained representation of target motion characteristics, DFS has recently been widely used in WiFi-based sensing systems. For example, by estimating the DFS of the human torsos as they move through space, and combining it with other information (*e.g.*, AoA, ToF), some device-free localization systems can achieve accurate human tracking [13, 18, 30]. Some researchers use CSI to calculate the Doppler spectrograms when the user moves in the area of interest, which can be used for human activity recognition and fall detection [4, 22, 24, 25]. Gait recognition systems distinguish different humans based on differences in limb movement patterns and speeds of different users during natural walking, which can be extracted from Doppler spectrograms [23, 32, 33]. In addition, DFS is also used in gesture recognition tasks. By capturing DFS during the movement of the subject's arms or fingers, some studies can accurately classify a variety of different gestures [16, 19, 21, 26, 35].

8.2 FFT-based DFS Estimation

In previous WiFi-based sensing arts, there are mainly two ways to estimate DFS using CSI. The first method is FFT-based [16, 19, 21, 23, 24, 26, 33, 35]. These methods usually apply *Short-Time Fourier Transform* (STFT) to the amplitude or phase of CSI to generate spectrograms over the time and Doppler frequency domains. For example, WiFiU [23] performs STFT on the *Principal Component Analysis* (PCA) component of the CSI amplitude to generate spectrogram, and applies spectrogram enhancement techniques to reduce noise in the spectrogram, thereby extracting human gait features. WiDance [19] defines the sign of DFS based on whether the target is moving closer to or farther from the transceiver, thereby associating DFS with the direction of movement and classifying eight activities of the dancer's legs. Wi-PIGR [33] collects CSI from two pairs of orthogonal transceivers, generates sub-spectrograms from CSI amplitudes in segments, and then aligns and fuses multiple independent sub-spectrograms to obtain a complete path-independent gait spectrogram. Widar3.0 [35] applies time-frequency analysis to CSI to obtain the DFS profiles, further extracts the velocity distributions of gestures in body coordinates, and serves as a unique indicator of gestures.

8.3 Path-Parameters-based DFS Estimation

Another way to estimate DFS is path-parameters-based [13, 18, 29, 30]. These methods model the multipath propagation channel between Tx-Rx as consisting of DFS and multiple other path parameters (*e.g.*, AoA), and thus estimate the DFS of each path. For example, IndoTrack [13] captures CSI from multiple antennas, uses the

MUSIC algorithm to process CSI phase information to estimate DFS, and combines DFS with AoA to achieve accurate device-free human tracking. Widar2.0 [18] uses multi-dimensional parameters including AoA, ToF, DFS, and attenuation to model the propagation of reflection paths, and effectively solves the parameters of each path based on the EM algorithm, and uses the results for human tracking. mD-Track [30] further introduces AoD as one of the path parameters and also solves them based on the EM algorithm.

9 Conclusion

This paper presents the design and implementation of *freeDoppler* a fine-grained Doppler effect learning network for accurate and robust velocity estimation using WiFi devices, regardless of the relative orientation of the targets, even for multiple targets. *freeDoppler* has two main technical contributions. First, it includes a velocity learning network VEN to learn important information about WiFi CSI to enhance the Doppler effect. VEN uses complex-valued CSI in continuous time segments as input, and accurately predicts a series of velocities in each time segment. Second, it also considers the high cost of actually collecting CSI data and constructs a CSI generation model CGM to generate large-scale simulated CSI samples. CGM adopts the electric field scattering model of Maxwell's equations and the noise addition method to improve the quality of simulated CSI, and considers the generation of intermediate state CSI to reduce complexity and accelerate fitting. Our real-world evaluation shows that *freeDoppler* outperforms state-of-the-art methods with a median error of 7.98 cm/s for velocity estimation, 28° for direction estimation, and 35 cm for human tracking with one or two moving targets, and is robust to various trajectories.

Acknowledgments

This work is partially supported by Hebei University High-level Talents Scientific Research Start-up Project (Grant No. 521100225229), the Young Scientists Fund of the National Natural Science Foundation of China (Grant No.62502214), the Basic Research Program (Natural Science Foundation) of Jiangsu (Grant No. BK20250727), the "Pioneer" and "Leading Goose" R&D Program of Zhejiang (Grant No. 2023C01029), the Jiangsu Province Higher Education Basic Science (Natural Science) Research Project (Grant No. 25KJB520026), and the Startup Foundation for Introducing Talent of NUIST (Grant No. 2025r068).

References

[1] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. 2017. Julia: A fresh approach to numerical computing. *SIAM review* 59, 1 (2017), 65–98.

[2] Robert Bodor, Bennett Jackson, and Nikolaos Papanikolopoulos. 2003. Vision-based human tracking and activity recognition. In *Proc. of the 11th Mediterranean Conf. on Control and Automation*, Vol. 1. 1–6.

[3] Hong Cai, Belal Korany, Chitra R Karanam, and Yasamin Mostofi. 2020. Teaching rf to sense without rf training measurements. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 4 (2020), 1–22.

[4] J Cardenas, Carlos A Gutierrez, and Ruth Aguilar-Ponce. 2020. Effects of antenna orientation in fall detection systems based on wifi signals. In *2020 IEEE Latin-American Conference on Communications (LATINCOM)*. IEEE, 1–6.

[5] Xingyu Chen and Xinyu Zhang. 2023. RF Genesis: Zero-shot generalization of mmwave sensing through simulation-based data synthesis and generative diffusion models. In *Proceedings of the 21st ACM Conference on Embedded Networked Sensor Systems*. 28–42.

[6] Redwan Dahmouche, Omar Ait-Aider, Nicolas Andreff, and Youcef Mezouar. 2008. High-speed pose and velocity measurement from vision. In *2008 IEEE international conference on robotics and automation*. IEEE, 107–112.

[7] David J Griffiths. 2023. *Introduction to electrodynamics*. Cambridge University Press.

[8] Zhiping Jiang, Tom H Luan, Xincheng Ren, Dongtao Lv, Han Hao, Jing Wang, Kun Zhao, Wei Xi, Yueshen Xu, and Rui Li. 2021. Eliminating the barriers: Demystifying wi-fi baseband design and introducing the picoscenes wi-fi sensing platform. *IEEE Internet of Things Journal* 9, 6 (2021), 4476–4496.

[9] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. 2023. Ultralytics YOLO. https://github.com/ultralytics/ultralytics

[10] Dominik Kellner, Michael Barjenbruch, Jens Klappstein, Jürgen Dickmann, and Klaus Dietmayer. 2015. Tracking of extended objects with high-resolution Doppler radar. *IEEE Transactions on Intelligent Transportation Systems* 17, 5 (2015), 1341–1353.

[11] Shengjie Li, Xiang Li, Qin Lv, Guiyu Tian, and Daqing Zhang. 2018. WiFit: Ubiquitous bodyweight exercise monitoring with commodity wi-fi devices. In *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. IEEE, 530–537.

[12] Wenwei Li, Ruiyang Gao, Jie Xiong, Jiarun Zhou, Leye Wang, Xingjian Mao, Enze Yi, and Daqing Zhang. 2024. WiFi-CSI difference paradigm: Achieving efficient doppler speed estimation for passive tracking. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 8, 2 (2024), 1–29.

[13] Xiang Li, Daqing Zhang, Qin Lv, Jie Xiong, Shengjie Li, Yue Zhang, and Hong Mei. 2017. IndoTrack: Device-free indoor human tracking with commodity Wi-Fi. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 1–22.

[14] Kai Niu, Xuanzhi Wang, Fusang Zhang, Rong Zheng, Zhiyun Yao, and Daqing Zhang. 2022. Rethinking Doppler effect for accurate velocity estimation with commodity WiFi devices. *IEEE Journal on Selected Areas in Communications* 40, 7 (2022), 2164–2178.

[15] Sharon Nuzik, Robert Lamb, Ann VanSant, and Susanne Hirt. 1986. Sit-to-stand movement pattern: a kinematic study. *Physical therapy* 66, 11 (1986), 1708–1713.

[16] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. 2013. Whole-home gesture recognition using wireless signals. In *Proceedings of the 19th annual international conference on Mobile computing & networking*. 27–38.

[17] Kun Qian, Chenshu Wu, Zheng Yang, Yunhao Liu, and Kyle Jamieson. 2017. Widar: Decimeter-level passive tracking via velocity monitoring with commodity Wi-Fi. In *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. 1–10.

[18] Kun Qian, Chenshu Wu, Yi Zhang, Guidong Zhang, Zheng Yang, and Yunhao Liu. 2018. Widar2. 0: Passive human tracking with a single Wi-Fi link. In *Proceedings of the 16th annual international conference on mobile systems, applications, and services*. 350–361.

[19] Kun Qian, Chenshu Wu, Zimu Zhou, Yue Zheng, Zheng Yang, and Yunhao Liu. 2017. Inferring motion direction using commodity wi-fi for interactive exergames. In *Proceedings of the 2017 CHI conference on human factors in computing systems*. 1961–1972.

[20] Fei Shang, Panlong Yang, Dawei Yan, Sijia Zhang, and Xiang-Yang Li. 2024. LiquImager: Fine-grained Liquid Identification and Container Imaging System with COTS WiFi Devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 8, 1 (2024), 1–29.

[21] Raghav H Venkatnarayan, Griffin Page, and Muhammad Shahzad. 2018. Multi-user gesture recognition using WiFi. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. 401–413.

[22] Hao Wang, Daqing Zhang, Yasha Wang, Junyi Ma, Yuxiang Wang, and Shengjie Li. 2016. RT-Fall: A real-time and contactless fall detection system with commodity WiFi devices. *IEEE Transactions on Mobile Computing* 16, 2 (2016), 511–526.

[23] Wei Wang, Alex X Liu, and Muhammad Shahzad. 2016. Gait recognition using wifi signals. In *Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing*. 363–373.

[24] Wei Wang, Alex X Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. 2015. Understanding and modeling of wifi signal based human activity recognition. In *Proceedings of the 21st annual international conference on mobile computing and networking*. 65–76.

[25] Wei Wang, Alex X Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. 2017. Device-free human activity recognition using commercial WiFi devices. *IEEE Journal on Selected Areas in Communications* 35, 5 (2017), 1118–1131.

[26] Dan Wu, Ruiyang Gao, Youwei Zeng, Jinyi Liu, Leye Wang, Tao Gu, and Daqing Zhang. 2020. FingerDraw: Sub-wavelength level finger motion tracking with WiFi signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 1–27.

[27] Dan Wu, Youwei Zeng, Ruiyang Gao, Shenjie Li, Yang Li, Rahul C Shah, Hong Lu, and Daqing Zhang. 2021. WiTraj: Robust indoor motion tracking with WiFi signals. *IEEE Transactions on Mobile Computing* 22, 5 (2021), 3062–3078.

[28] Dan Wu, Daqing Zhang, Chenren Xu, Yasha Wang, and Hao Wang. 2016. WiDir: Walking direction estimation using wireless signals. In *Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing*. 351–362.

[29] Yaxiong Xie, Jie Xiong, Mo Li, and Kyle Jamieson. 2016. xD-track: Leveraging multi-dimensional information for passive wi-fi tracking. In *Proceedings of the 3rd Workshop on Hot Topics in Wireless*. 39–43.

[30] Yaxiong Xie, Jie Xiong, Mo Li, and Kyle Jamieson. 2019. mD-Track: Leveraging multi-dimensionality for passive indoor Wi-Fi tracking. In *The 25th Annual International Conference on Mobile Computing and Networking*. 1–16.

[31] Zheng Yang, Yi Zhang, Kun Qian, and Chenshu Wu. 2023. SLNet: A Spectrogram Learning Neural Network for Deep Wireless Sensing. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. 1221–1236.

[32] Lei Zhang, Cong Wang, Maode Ma, and Daqing Zhang. 2019. WiDIGR: Direction-independent gait recognition system using commercial Wi-Fi devices. *IEEE Internet of Things Journal* 7, 2 (2019), 1178–1191.

[33] Lei Zhang, Cong Wang, and Daqing Zhang. 2021. Wi-PIGR: Path independent gait recognition with commodity Wi-Fi. *IEEE Transactions on Mobile Computing* 21, 9 (2021), 3414–3427.

[34] Xiaopeng Zhao, Zhenlin An, Qingrui Pan, and Lei Yang. 2023. NeRF2: Neural Radio-Frequency Radiance Fields. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. 1–15.

[35] Yue Zheng, Yi Zhang, Kun Qian, Guidong Zhang, Yunhao Liu, Chenshu Wu, and Zheng Yang. 2019. Zero-effort cross-domain gesture recognition with Wi-Fi. In *Proceedings of the 17th annual international conference on mobile systems, applications, and services*. 313–325.

[36] Huiyang Zhou, Pengfei Cao, and Shujing Chen. 2016. A novel waveform design for multi-target detection in automotive FMCW radar. In *2016 IEEE Radar Conference (RadarConf)*. IEEE, 1–5.

Received 1 March 2025; revised 10 August 2025; accepted 13 October 2025