InFit: Combination Movement Recognition For Intensive Fitness Assistant Via Wi-Fi

Huichuwu Li, *Student Member, IEEE,* Jiang Xiao, *Member, IEEE,* Wei Wang, *Member, IEEE,* Lu Wang, *Member, IEEE,* Dian Zhang, *Member, IEEE,* and Hai Jin, *Fellow, IEEE*

Abstract—Wi-Fi technology is becoming a promising enabler of device-free fitness tracking to provide reviews and recommendations for effective homely exercise. State-of-the-art Wi-Fi fitness assistants succeed in recognizing the simple meta-movements (e.g., Push-Up and Squat) with discrete and repeatable patterns. Unfortunately, these prior attempts can hardly scale to the combination movements of ever-growing interests in intensive fitness programs. Combination movements are composed of meta-movements that are mutually concatenated or inserted. They have a compound characteristic that inherits from the diversity of combination orders and continuity of meta-movements. The compound characteristic causes substantial training data collection costs and a challenge of combination decomposition that is a prerequisite for providing fine-grained fitness assessment. To this end, we propose *InFit*, a Wi-Fi-based device-free fitness assistant system for combination movements. First, we design a novel data augmentation method, namely *Stitching-based Virtual Sample Generation* (SVSG), to reduce the training data collection costs by generating virtual combination movements. Second, a 2-stage combination movement recognition model is designed to learn temporal dependencies between movements and decompose combination movements. From its outputs, we can tell whether a combination movement is standard. Extensive experimental results show that InFit can achieve an average recognition accuracy of 94%. With zero training samples of combination movements, the average accuracy is 40% higher than the baselines. In addition, SVSG can provide a general enhancement on multiple competing schemes with similar sensing tasks.

Index Terms—Fitness assistant, Combination movement, meta-movement, Virtual sample generation, Wi-Fi.

1 Introduction

At-home workout such as *high-intensity interval training* (HIIT) [1] is becoming a trendy way of keeping healthy and in shape [2], not only for the routine fitness but also in light of the COVID-19 outbreak. In general, a fitness regime with high intensity can bring tangible health benefits, including improving lung capacity, decreasing body fat, and strengthening cardiopulmonary function, etc. Meanwhile, this may lead to a high possibility of injury risks [3]. Hiring a personal trainer is expensive and unavailable at home, especially during an epidemic like COVID-19. Hence, there is an emerging trend of utilizing virtual fitness assistants to improve workout effectiveness.

Referring to state-of-the-art systems [4], the key requirements of a virtual fitness assistant are not only movement recognition and number counting but also fine-grained assessments of workout quality. Thus, users can know whether their movements are standard and how effectively they exercise. Researchers have explored various signal sources to assess fitness movements, especially camera-based [5] and wearable sensor-based [6], [7] solutions. However, camera-based solutions require good lighting conditions and may cause privacy concerns, making them inappropriate for in-home environments. Sensor-based methods are environment-independent but impose wearing burdens on people. Accounting for this, we select to leverage Wi-Fi for fitness movement recognition and assessment since Wi-Fi signals are robust to lighting condition changes and have a low risk of privacy concerns. Moreover, many works [8], [9] have validated that monitor movements precisely in a device-free manner.

In-home fitness movements can be divided into two-folds: meta-movement and combination movement. Current Wi-Fi-based fitness assistant systems [4], [10] focus on the meta-movements, e.g., squat and sit-up. We define the meta-movement as the movements with repeatable and undivided patterns. "Repeatable" means that there are no connecting movements between the repetitions when movement is repeated. Thereby, the start and end postures of a meta-movement are the same. "Undivided" gives a constraint that a meta-movement cannot be decomposed into multiple repeatable movements.

Combination movements are the compositions of meta-movements under two requirements. First, there should be the reasonable logic among adjacent meta-movements. For example, people cannot jump again when they are in the air, or begin a squat during in the mid-time of push-up. Second, combination movements should have repeatable movement patterns. Theoretically, people can design countless combination movements. Hence, combination movements have a compound characteristic that inherits from the continuity of meta-movements and diversity of combination orders. This

Huichuwu Li, Jiang Xiao, and Hai Jin are with the National Engineering Research Center for Big Data Technology and System, Services Computing Technology and System Laboratory and the Cluster and Grid Computing Laboratory, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: {credolee, jiangxiao, hjin}@hust.edu.cn).(Corresponding Author: Jiang Xiao)
Wei Wang is with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: weiwangw@hust.edu.cn).
Lu Wang and Dian Zhang is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China (e-mail: {wanglu, zhangdian}@szu.edu.cn).

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3209656

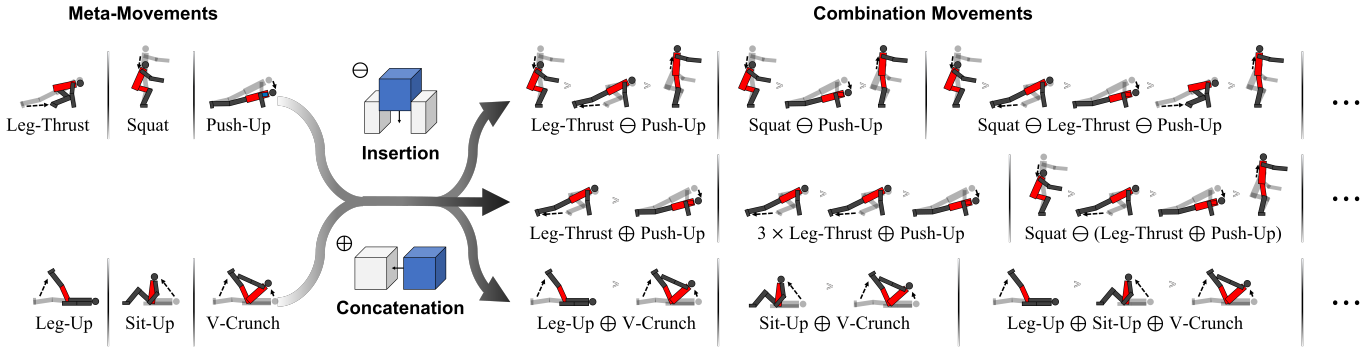


Fig. 1: Combination movements stem from meta-movements through the concatenation and insertion rules.

compound characteristic causes new challenges in training data collection and fitness assessment. Hence, we propose *InFit*, a device-free combination-movement-oriented fitness assistant based on Wi-Fi signal.

The diversity leads to a substantial data collection cost. As shown in Fig. 1, combination movements are composed of different numbers of different meta-movements in a relatively free sequence through insertion ($\ominus$) and concatenation ($\oplus$) rules. The types of combination movements are far more than that of the meta-movements. Besides the diversity of meta-movement types, users can easily design new combination movements by changing the number of meta-movements. For example, users can design a enhanced version of *Leg-Thrust*$\oplus$*Push-Up* by repeating the leg-thrust. Therefore, it is unsustainable to collect the data of all the combination movements for training. To reduce the training data collecting costs of combination movement, we propose a *Stitching-based Virtual Sample Generation* (SVSG). SVSG generates virtual combination movement samples by simulating the composition rules of insertion and concatenation. The virtual samples can augment the temporal correlations between meta-movements within combination movements.

The continuity makes it challenging to decompose combination movements for a fine-grained fitness assessment. The transition states (e.g., pauses) among meta-movements are variable, making existing segmentation methods unsuitable for decomposing combination movements into meta-movement sequences. Thus, it isn't easy to assess the effectiveness of a combination movement since we don't know the completion quality of the meta-movements within it. Fortunately, we observe that every meta-movement consists of two reversed movement states: retraction and extension. Both of them have a common speed change trend: beginning with an ascending speed and ending with a descending speed. Based on these observations, we design a 2-stage combination movement recognition model. It leverages the temporal dependencies of movement speeds to decompose combination movements, making InFit able to provide fine-grained fitness assessments.

The contributions of this work can be summarized as follows:

- To the best of our knowledge, InFit is the first in-depth analysis of combination movements and exploits the unique compound characteristic for recognition under the condition of insufficient combination movement data.
- We propose the SVSG for data augmentation. SVSG combines meta-movements by simulating the combination rules to generate virtual combination movements. Thus, SVSG can provide sufficient data for the recognition model to learn the context information between meta-movements.
- We observe a common speed change rule shared by meta-movements. Based on this rule, we design a 2-stage combination movement recognition model to provide fine-grained movement information for fitness assessments.
- Experimental results show that InFit achieves an average combination motion recognition accuracy of 94%. The recognition accuracy under the condition of zero-knowledge is 40% higher than the state-of-the-arts. Moreover, SVSG can provide a general enhancement even for other schemes designed for similar tasks.

The rest of this paper is organized as follows: In Section 2, we compare InFit with the related work. In Section 3, we introduce the architecture of InFit followed by the details of preprocessing, data augmentation, and movement recognition. Then, we evaluate InFit in Section 4 and give a conclusion in Section 5.

2 Related Work

In this section, we review existing fitness assistants and WiFi-based activity identification systems grouped into three categories: *Vision-based*, *Inertial sensor-based*, and *Wireless-based*.

Vision-based fitness assistant systems leverage image processing techniques to extract movement information from RGB [5] or depth [11], [12] images. They can accurately track human poses, but their performance depends on the brightness conditions and cannot track the occluded targets. Besides, deploying cameras in-home may bring privacy concerns.

Inertial sensor-based systems, e.g., RecoFit [13], FitCoach [6], and MM-Fit [14], attach inertial sensors on human body or fitness equipment to monitor workout activities. They are more robust to environment changes than the vision-based systems, while suffering more tedious data collection processes. Xie et al. [15] assume that derived from some meta-activities with small-angle changes. They

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3209656

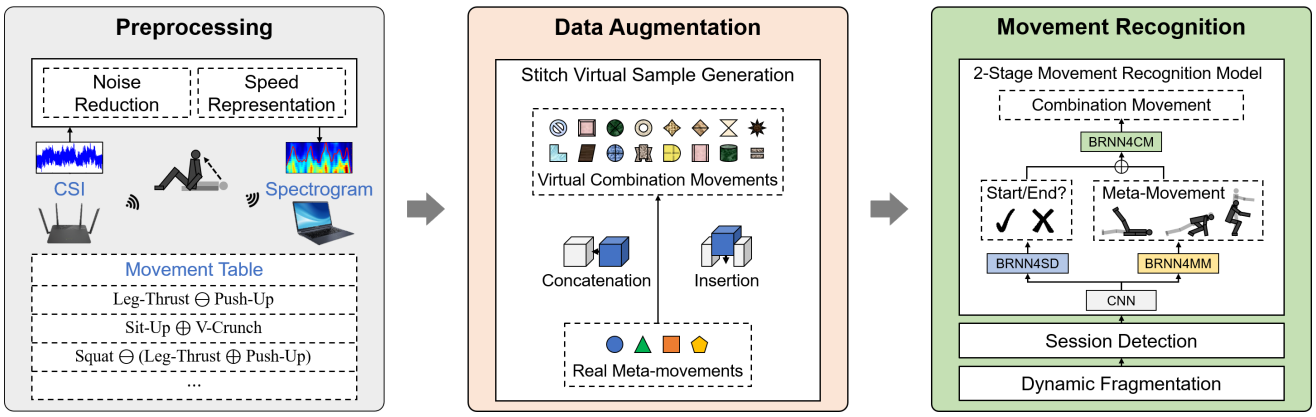


Fig. 2: The architecture of InFit. In the last component, "BRNN4CM", "BRNN4SD", and "BRNN4SM" are the BRNNs for combination movement estimation (4CM), state detection (4SD), and meta-movement recognition (4MM), respectively.

converted movements into meta-activity sequences and recognized their types through a lightweight model trained by a few samples to lower the data overheads. Unfortunately, such a meta-activity-based method is unsuitable for Wi-Fi-based sensing tasks. Due to the narrow bandwidth, it is non-trivial to use commercial Off-the-shelf Wi-Fi devices to track specific body parts.

Wireless-based activity sensing technology shows a promising future due to the non-invasive and penetrating characteristics. Researchers have tried to leverage different wireless signals, e.g., *Radio Frequency Identification* (RFID) [16], ultrasonic signals [4], and Wi-Fi [17], for movement recognition. Existing wireless-based systems can be divided into two categories. The first category is *Detection Before Recognition* where the systems extract movement-related signal segments from the received signal streams according to some clear transition states (i.e., static intervals [8], [10] and preamble gestures [9]). Then, they leverage machine learning methods for recognition. Such systems regard every movement as a meta-movement. As a result, collecting labeled data of diverse combination movements is labor tedious, which limits their practically. The second category is *Recognition Before Detection* where the systems first recognize the type of movements followed by counting the repetitions. Like WiStep [18] and WiRun [19], they extracted the specific walking-induced sinusoid-like patterns from the received Wi-Fi signals and adopted peak detection to count steps. Nevertheless, such peak detection methods is unavailable for combination movements due to the complex movement patterns. DeepSense [20] and EI [21] turned movement recognition into sequence-to-sequence classification problems, making them able to recognize complex movement patterns. However, they cannot provide a fine-grained assessment of exercise quality because of lacking the capabilities to decompose combination movements. To this end, we combine meta-movements to generate virtual combination movements for data augmentation and design a combination-movement-oriented recognition model for fitness assessment.

3 System Design

InFit consists of three components: *Preprocessing*, *Data Augmentation*, and *Movement Recognition* as shown in Fig. 2. In *Preprocessing*, InFit takes two types of data as inputs. One is noisy CSI measurements caused by the multipath effect and the imperfect design of devices. Therefore, InFit adopts a series of operations to extract speed features from the noisy signals. The other type is a predefined movement table recording registered meta-movements and the derived combination movements. InFit leverages this movement table as a guide to generate virtual samples in *Data augmentation*. Finally, we gather the virtual samples and the real ones together to train a 2-stage movement recognition model in *Movement recognition*. This model can identify movement types and provide fine-grained fitness assessment.

3.1 Preprocessing

Human movements can change the associated reflection paths, resulting in Doppler shifts. It allows us to infer the speed pattern of movements according to the frequency shifts. In this section, we introduce how InFit mitigates the noise and transforms the CSI into a more intuitive format.

Noise Reduction. CSI depicts the channel states of propagation links, which are influenced by the effects of scattering, fading, multipath, and hardware imperfection. The received CSI of a *transmitting-receiving antennas pair* (T-R pair) at time t is $H(f,t) = e^{-i\Delta\theta}(\sum_{k=1}^{P_S} \alpha_k e^{-i2\pi f \tau_k} + \sum_{k=1}^{P_D} \alpha_k e^{-i2\pi f \tau_k})$, where f is the carrier frequency, τ_k is the propagation time of path α_k, P_D and P_S are the propagation paths reflected by human body and static things, e.g., wall, floor, and roof. $\Delta\theta$ indicates the linear phase offsets caused by imperfect design of devices [22]. Therefore, our goal is to extract $\sum_{k=1}^{P_D} \alpha_k e^{-i2\pi f \tau_k}$ from the received signals.

InFit removes $\Delta\theta$ through the conjugate multiplication proposed by Qian et al. [23]. We set one receiver equipped with three antennas to receive Wi-Fi signals in this work. The receiving antennas share the same processing circuits and experience similar phase shifts caused by hardware imperfection. Therefore, we can remove the phase shifts by calculating the conjugate multiplication of two antennas' measurements: $H_1(f,t)H_2(f,t)^* = H_{s1}(f,t)H_{s2}^*(f,t) + H_{s1}(f,t)\sum_{d2\in P_{d2}} \boldsymbol{a}_{d2} e^{+i2\pi f \tau_{d2}} + H_{s2}^*(f,t)\sum_{d1\in P_{d1}} \boldsymbol{a}_{d1} e^{-i2\pi f \tau_{d1}} + \sum_{d1\in P_{d1}, d2\in P_{d2}} \boldsymbol{a}_{d1}\boldsymbol{a}_{d2} e^{-i2\pi f(\tau_{d1}-\tau_{d2})}$.

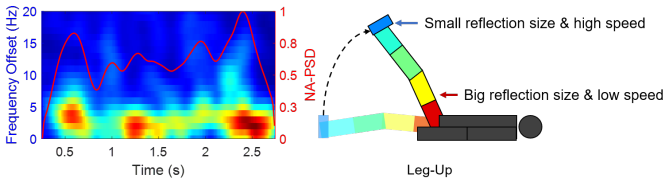


Fig. 3: Spectrogram and NA-PSD of leg-up

The first component, $H_{s1}(f,t)H^*_{s2}(f,t)$, has no frequency shifts, and we can remove it by a high-pass filter. The fourth component indicates the difference between two propagation paths, which negatively influence reproducing movement speed patterns. Fortunately, it can also be filtered out by the high-pass filter. The fourth component related phase shifts is decided by the path difference $d1 - d2$. This difference is approximately equal to the distance between the antennas pair. Therefore, the corresponding frequency shifts are close to zero Hertz. The next step is to let one of the remained two components dominates $H_1(f,t)H_2(f,t)^*$. We select two antennas with the highest amplitude and variance of CSI and calculate their conjugate multiplication. Next, we remove the information of moving directions by calculating $|H_1(f,t)H_2(f,t)^*|$ to simplify the system design. The CSI extraction tool [24] InFit used can report 30 sub-carriers in a propagation channel. Hence, the conjugate multiplication is $|\boldsymbol{h}_1 \circ \boldsymbol{h}_2^*| = \{|h_1(f_1)h_2(f_1)^*|, \ldots, |h_1(f_{30})h_2(f_{30})^*|\}$.

Then, we leverage a band-pass filter to remove the low-frequency noise discussed above and high-frequency noise beyond the upper bound on the motion-induced frequency shifts. Previous studies [25], [26] have validated that motion-induced Doppler shifts can be calculated by $f = \frac{2v}{\lambda}$, where v and λ are movement speed and RF signals' wavelength, respectively. InFit works at channel 36, the center frequency is $5.18GHz$, and λ is about $0.05m$. Since the speeds of most indoor fitness movements are lower than $5m/s$, the pass-band is set to $[2Hz, 200Hz]$.

Finally, InFit leverages *Principal Component Analysis* (PCA) to remove the in-band noise and reduce feature dimensions [25]. The signal streams are cut into chunks by a sliding window. The window size and stride length are both one second. We utilize PCA to calculate the principal components of each chunk and calculate the average value of the principal components from the second to the fifth to reduce the feature dimension.

Speed Representation. InFit transforms the denoised data into more intuitive formats: a spectrogram and a *normalized accumulative power spectral density* (NA-PSD) curve. The former reflects the fine-grained speed information of movement, providing rich features for movement recognition. The latter reflects the coarse-grained speed information of the whole-body movement and simplifies the designs of InFit, i.e., SVSG, dynamic fragmentation, data annotation, and fitness assessment.

InFit calculates the spectrogram of the de-noised CSI by short-time Fourier transform. The sliding window size is set to 512, and the stride length is 16. As shown by the spectrogram on the left in Fig. 3, the left y-axis is the motion-induced frequency offsets, proportional to the movement speed. The temperature of the pixels indicates signal energy, reflecting the size of the reflective surface. Specifically, when the user lifting his leg, his thigh and feet move with a similar angular speed-changing trend. Thigh moves in low speed, causing low-frequency offsets, while the larger surface size can reflect more signal than feet resulting in stronger signal strength. In contrast, the feet move faster, but their smaller reflection areas make the corresponding signal strength weaker.

The red line in the spectrogram is the NA-PSD, an integrated indicator of frequency offsets and signal strengths: $napsd(t) = MINMAX(\sum_{i=1}^{N_f} f_i * r_i(f_i,t))$. N_f is the number of frequencies, f_i is the frequency offset related to moving speed, and $r_i(f_i,t)$ is the signal power associated with the size of the reflection surface. Hence, the NA-PSD is dominated by the body parts with larger sizes and faster speeds.

3.2 Data Augmentation

SVSG generates virtual samples by simulating the composition rules of *concatenation* and *insertion*. To make the virtual samples as real as possible, the problems of "when to stitch" and "how to stitch" need to be solved.

"When to stitch" means that meta-movements need to be reasonably connected. For example, a person cannot do a push-up when jumping in the air. Combination movements can derived from meta-movements through two ways:

- *Concatenation* operation stitches the meta-movements in a chain rule. We determine a random place on the last quarter of the NA-PSD curve of the previous meta-movement, and its NA-PSD value is v. Then, we scan the NA-PSD curve of the latter meta-movement. The first place that reaches the NA-PSD value v is the connection place.
- *Insertion* operation constructs virtual samples by inserting meta-movements in other ones. The place to insert is named "final position," [10] which divides a meta-movement into two reversed movement states: retraction and extension. We observe that these movement states have a common speed change trend: ascending at the beginning and descending before entering the next part. Hence, the final position has a minimal value. Based on this observation, we determine the insertion place by finding the local minimum speed close to the central place on the NA-PSD curve.

"How to stitch" makes a virtual sample more reliable and looks like a real one. The challenge is to ensure movement consistency and continuity. The meta-movements used by SVSG were collected in different environmental and physical conditions. If we directly stitch them together, the generated virtual samples will lack consistency due to their different signal states, such as the energy level. Therefore, SVSG adjusts the signal energy to the same level to rebuild the spatial continuity. Since the body movements are often at low-speed, SVSG calculates the average energy of the adjacent samples' low-frequency components. Then it uses the equation, $S_l = \frac{\rho_l}{\rho_p} \times S_p$, to balance the two samples' energy level. S_p and S_l indicates the previous and the later samples. ρ_p and ρ_l are the average energy of the low-frequency

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3209656

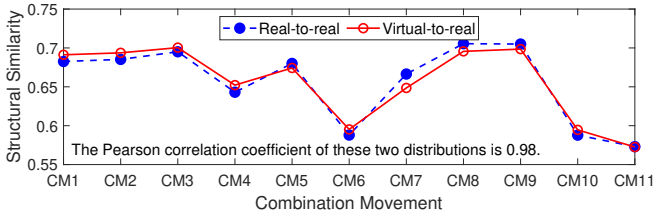


Fig. 4: Similarity between real and virtual samples

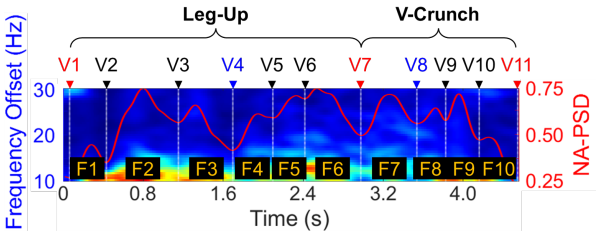


Fig. 5: A dynamic fragmentation example of CM1

components of S_p and S_l. Accounting for the movement continuity, SVSG should compensate the inevitable transitional movements to improve reliability. For example, SVSG needs to simulate the movements from standing to a plank position when generating a virtual sample of Jump⊕Push-Up. Fortunately, the inevitable transitional movements can be simulated by some meta-movements, e.g., the jump and push-up are connected by a squat and a leg-thrust.

To validate the reliability of SVSG, we first introduce the structural similarity (SSIM) [27] to quantify the similarity between movement samples. Then, we conducted an experiment to evaluate the overall performance of SVSG from a statistical view.

Since the samples are in the format of spectrogram which can be regarded as gray-scale images, we introduce leveraging the image-oriented metric, SSIM, to calculate the similarity between samples. SSIM is a widely used metric that models the similarity of images as the combination of brightness, contrast, and structural information. $SSIM(x,y) = \frac{(2\mu_x\mu_y+c_1)(2\mu_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\mu_x^2+\mu_y^2+c_2)}$, where x and y are the spectrogram of two samples, μ and σ indicate the mean value and standard deviation. The value of SSIM ranges from -1 to 1. When x and y are the same, $SSIM(x,y)$ is equal to 1.

An empirical experiment was conducted to evaluate the overall performance of SVSG. The key idea is that we will get two close SSIM values if comparing two similar samples (x_1 and x_1) to another one. Therefore, we calculated two SSIM distributions: *virtual-to-real* D_{v2r} and *real-to-real* D_{r2r}. D_{v2r} should be similar to D_{r2r} if the data generated by SVSG is reliable.

Specifically, we obtained D_{v2r} by repeating the following steps 1000 times: (i) generating a virtual sample according to the movement table randomly, (ii) selecting a random sample of the same type in the real dataset, and (iii) computing the SSIM value of them. Then, we replaced the first step with selecting a random real-sample and calculated D_{r2r} in a similar way.

The similarity distributions of eleven combination movements are shown in Fig. 4. The distribution of virtual-to-real is similar to the real-to-real's distribution. The Pearson correlation coefficient of these two distributions is up to 0.98, indicating that the virtual samples generated by SVSG are close to the real ones. In addition, we observed that CM6, CM10, and CM11 have the lowest similarity among the combination movements. The reason is due to the instability of physical movements. These combination movements consist of more than three meta-movements, which brings more difficulties to people keeping a stable performance.

3.3 Movement Recognition

This section introduces how InFit recognizes combination movements. First, it leverages a dynamic fragmentation method to convert the spectrogram into "images" sequences of the same size. Then, InFit trains a simple neural network for session detection, separating fitness sessions from other unknown activities. Finally, the detected session will be sent to train the 2-stage deep neural network for combination movement recognition.

3.3.1 *Dynamic Fragmentation*

InFit designs a speed-based structuring method named dynamic fragmentation with no ambiguity in the transition states. Traditional methods, like DeepSense [20] and EI [21], divide the data into size fixed pieces by a predefined sliding window and stride length. A fragment at the transition state may have the features of the two adjacent meta-movements. As a result, it creates a blurred boundary between two meta-movements, which influences the assessment quality. Thereby, InFit proposes dynamic fragmentation to remove this ambiguity based on the speed change rule of the transition states.

As prior works [4], [10] found that each meta-movement consisted of a retraction part and an extension part. For example, when doing leg-up, a man begins with an abdominal contraction to lift his legs to the final position (vertical to the ground) in the retraction part. Then, the legs fall to their initial places with abdominal relaxation in the extension part. We observe that all the movement states have the same speed trend: beginning with an ascending speed and ending with a descending speed. This trend also exists in the distorted meta-movements within combination movements.Theoretically, there are two peaks and three valleys on the NA-PSD curve of a meta-movement. The peaks divide a meta-movement into two retraction and extension states. Then, we can easily find the transition times among meta-movements according to the sequence of movement states.

In fact, a meta-movement usually matches more than two segments due to the instability of human motions. Fig. 5 demonstrates the dynamic fragmentation result of a CM1, the concatenation of one leg-up and one v-crunch. The valleys marked as Vi are the temporal boundaries of the fragments Fi. $V1$ and $V11$ indicate the initial positions of leg-up and v-crunch, $V4$ and $V8$ indicate their final positions. $V7$ indicates the transition state between these two meta-movements. The NA-PSD curve still reflects the speed trend that begins with an ascending speed and ends with a descending speed in each part, though

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3209656

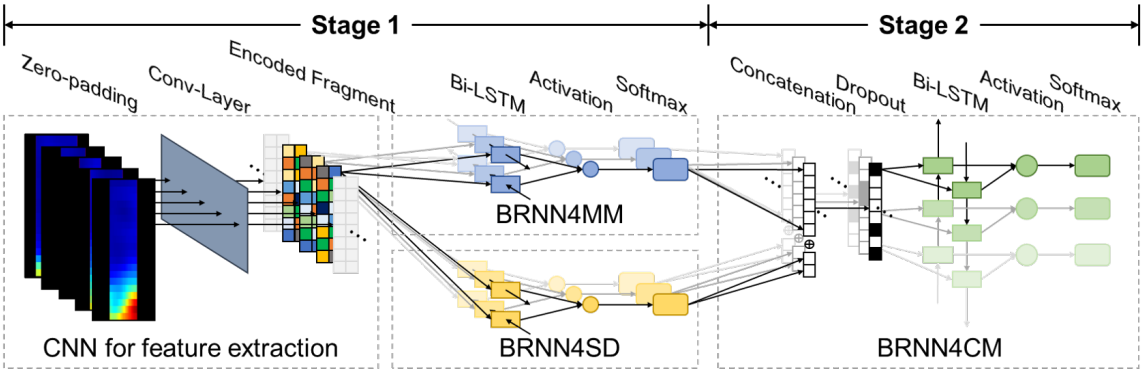


Fig. 6: 2-stage Combination Movement Recognition Model. The first stage's tasks are meta-movement classification and their states detection. The second stage is combination movement classification.

movement instability causes some fluctuations. Although dynamic fragmentation cannot ensure the ideal case that divides every meta-movement into two fragments, it can still remove the ambiguity at the transition states. The movement instability brings another problem that the length of the segment sequence is variable. It is difficult to identify the movement states based on some simple rules or models. Therefore, we leverage a recurrent neural network to learn the temporal dependencies among fragments and classify them into retraction and extension classes.

The last problem before feeding data to the recognition models is how to convert the segments into a structured size. As the recognition model utilizes CNN to extract segments' features automatically, the segments should be structured. We made a statistic on the duration time of the fragments. The result shows that 95% of the fragments are shorter than $0.25s$. Therefore, we scale the segments to $0.25s$ by interpolation and downsampling.

3.3.2 Session Detection

Session detection aims at separating fitness sessions from other unknown activities. Prior works [4], [10] realize according to the number of repetitions patterns. Specifically, they utilize a predefined sliding window with size of $8s$ to calculate the auto-correlation curves and count the peaks on the curves to obtain the repetition number. If the repetition number is more than 3, a fitness session exists.

However, it may not work to recognize the fitness sessions of combination movements. It is difficult to determine the sliding window size due to the high dynamic characteristic. For example, the duration of three continuous leg-ups is about $6s$, while the duration of of three combination movements (*Leg-Up* $\oplus$ *V-Crunch* lasts over $12s$. Hence, the prior works can detect meta-movement sessions but fail to detect the combination movement sessions. To overcome this problem, we consider session detection as a sequence-to-sequence classification task without predefined sliding windows. We stack one layer CNN with one layer BRNN to learn spatiotemporal features among the segments and classify them into three classes: unknown movements, fitness sessions, and static states out of the sessions. The session detection model has a similar structure to the first stage of the 2-stage combination movement recognition model, and we train it individually.

3.3.3 2-stage Combination Movement Recognition Model

Fig. 6 shows the structure of the 2-stage combination movement recognition model. In the first stage, the model leverages CNN for feature extraction and is stacked with two BRNN for meta-movement classification (BRNN4MM) and state detection (BRNN4SD). Finally, the outputs of BRNN4MM and BRNN4SD will be concatenated as the input of the BRNN for combination movement classification (BRNN4CM) in the second stage.

CNN for feature extraction. InFit leverages a convolutional layer to extract the features of the fragments automatically. The convolutional layer has 4 kernels, of which the size is 1×1, and the stride is set to 3×1. A batch norm layer is used to normalize the mean and variance of the kernels outputs. Then, a max-pooling operation is used to improve the stability of extracted features and reduce the dimensions. The size of the pooling window is 2, and the stride length is set to 2. The outputs of the max-pooling layer will be flattened and fed in the next BRNN.

BRNN4MM. It is difficult to identify a fragment to which meta-movement belongs only based on the features extracted from CNN. A segment reflects the short-term ($0.25s$) moving speeds of the body parts. Different movements may have some similar segments. Therefore, InFit needs to combine the temporal dependencies between the fragments with their speed features for fragment identification.

InFit adopts BRNN to learn the temporal dependencies between fragments. Compared with the conventional RNN only using the previous context, BRNN leverages both the previous and future contexts to improve recognition performance. The BRNN has only one BRNN layer with 128 *Bidirectional Long Short-Term Memory* (BiLSTM) nodes. The output of BRNN4MM is a probabilistic matrix with size of $N \times M$, where M is meta-movement class number plus static interval, and N is the fragment sequence length. We set cross-entropy loss to train BRNN4MM as $L_M = \sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} \log \bar{y}_{ij} + \Omega_M$. Ω_M penalizes the estimations when they are overconfident [28], and its formulation is $\sum_{i=1}^{N}\sum_{j=1}^{M} (\log \bar{y}_{ij} + \log (1 - \bar{y}_{ij}))$.

BRNN4SD. Except for recognizing the type of meta-movement, detecting the start and end time of a meta-movement is necessary for combination movement decom-

TABLE 1: MOVEMENT TABLE

Movement Class	Composition	Movement Class	Composition	Movement Class	Composition
MM1	Leg-Up	MM7	Jump	CM6	Squat $\oplus$ Jump
MM2	Sit-Up	CM1	Leg-Up $\oplus$ V-Crunch	CM7	Squat $\ominus$ (Leg-Thrust $\ominus$ Push-Up) $\oplus$ Jump
MM3	V-Crunch	CM2	Sit-Up $\oplus$ V-Crunch	CM8	Leg-Up $\oplus$ Leg-Up $\oplus$ V-Crunch
MM4	Squat	CM3	Leg-Up $\oplus$ Sit-Up $\oplus$ V-Crunch	CM9	Leg-Up $\oplus$ Leg-Up $\oplus$ Leg-Up $\oplus$ V-Crunch
MM5	Leg-Thrust	CM4	Squat $\ominus$ Leg-Thrust	CM10	Jump $\oplus$ Jump $\oplus$ Squat
MM6	Push-Up	CM5	Squat $\ominus$ (Leg-Thrust $\ominus$ Push-Up)	CM11	Jump $\oplus$ Jump $\oplus$ Jump $\oplus$ Squat

position. To successfully decompose combination movements can help InFit provide fine-grain analysis of fitness sessions and tell users how are their exercise quality. However, it is non-trivial to detect the start or end time of the meta-movements within combination movements due to the compound characteristic.

Traditional methods detect the movement according to a specific movement state, such as a pause or a predefined movement. It is not work in combination movement recognition because the transition states between meta-movements are not as straightforward as static pauses. Hence, we need to find a more general rule describing the transition states. Fortunately, meta-movements usually consist of two reversed movement states: retraction and extension [4], [10], and we find a common speed change trend in these states: ascending at the beginning and descending before entering the next part.

These observations can help us decompose the combination movement without clear transition states. We regard state detection as a sequential-to-sequential classification problem and classify the segments into three classes, e.g., static pauses between combination movements, retractions, and extensions. Similar to the BRNN4MM, a BRNN is stacked on the CNN. The BRNN has one BiLSTM layer with 128 nodes. During the training phase, the loss function is $L_S = \sum_{i=1}^{N}\sum_{j=1}^{S} y_{ij}\log\bar{y}_{ij} + \Omega_S$. S, N, and Ω_S are the number of target classes, the length of segment sequences, and the penalty avoiding overfitting.

BRNN4CM. In theory, we can easily infer the combination movement if knowing the sequences of meta-movement types and their states. Therefore, the outputs of BRNN4MM and BRNN4SD will be concatenated to form the inputs of BRNN4CM (the inputs are probabilistic matrices with size of $(M+S)\times N$). However, we find some noises in the input sequence due to the errors of BRNN4MM and BRNN4CM. Taking the meta-movement V-crunch as example, volunteers may occasionally lift their torsos a quarter second later than the legs, and then, the movement will finished as a V-Crunch. In this case, some fragments at the beginning might be mistaken for Leg-Up. We regard these mistaken fragments as the noises in the outputs of BRNN4MM and BRNN4SD.

Hence, we add a dropout layer imitating the noise in the inputs to improve the robustness of BRNN4CM. The dropout rate is set to 0.5. Then, a BiLSTM layer with 100 hidden nodes is set to recognize the combination movements. The loss function of BRNN4CM is $L_C = \sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log\bar{y}_{ij} + \Omega_C$, where C is the class number of combination movements plus static interval.

Finally, we fine-tune BRNN4MM, BRNN4SD, and BRNN4CM together. The total loss is $L_T = \alpha L_M + \beta L_S + \gamma L_C$, where α, β, and γ are predefined hyperparameters to adjust the contributions. Empirically, we set them to 1 in this work.

4 EVALUATIONS

This section evaluates InFit from multiple aspects. It first introduces the experimental setup and three evaluation metrics. Then, InFit is evaluated from two aspects: micro and macro. The micro-evaluation part evaluates InFit's overall performance on different movements and the effectiveness of SVSG, and the macro-evaluation part discusses the robustness. Finally, we describe how InFit assesses the exercise quality.

4.1 Experimental Setup

This section first clarifies the settings of hardware and operation system, environments, movements, data collection, volunteers. Then, the workflow of training and inference is illustrated to help understand the evaluations on the subtasks of session detection, meta-movement classification, state detection, and combination movement classification. At last, we describe the baselines selected for comparison.

Hardware and operation system. We used a ThinkCentre E75s as a transmitter. It used an Intel 5300 NIC linked with one 6dBi omnidirectional antenna to send Wi-Fi signals. A laptop Thinkpad X200 equipped with an Intel 5300 NIC was used to work as the receiver, and it had three antennas. The operating system was Ubuntu 14.04. We used the tool proposed by Halperin et al. [24] to extract CSI measurements. InFit worked on Channel 36 at 5GHz, and the sampling rate is $1000Hz$. We leveraged a laptop with an NVIDIA GeForce RTX3060 Laptop GPU to train the *Deep Neural Network* (DNN)-based models.

Environment. We conducted experiments in two different environments. The first place is a spacious exhibition hall of which the size is $15\times 10m^2$. Since people rarely use the exhibition hall, the layout of the furniture is stable during data collection. The second place is a dormitory room which size is about $7\times 3m^2$. There are many objects, causing a serious multipath effect. Moreover, only one student in the dormitory room was selected as a volunteer. His roommates had neither been involved in the experiment nor been asked to restrain their behavior from keeping a good environment for our experiment. As a result, the layout of the dormitory room was dynamic.

Movements. We set 18 target movements, including 7 meta-movements and 11 combination movements, as shown in Table 1. The combination movements derive from the meta-movements in different composition ways. CM1, CM2,

CM3, and CM6 derive from the meta-movements by concatenation. CM4 and CM5 are the insertion results of multiple meta-movements. CM7 consists of MM4, MM5, MM6, and MM7 in a fusion way. To observe whether InFit will mistake continued repetitions as a single one, we set CM8 and CM9. In addition, we set CM10 and CM11 to evaluate the influence of composition order.

TABLE 2: FITNESS LEVEL OF USERS

Fitness Level	User ID
Master	U2, U7, U8, U15, U16, U18
Normal	U1, U5, U9, U10, U11, U12, U13, U14, U19
Novice	U3, U4, U6, U17, U20

Volunteers. We invited 20 volunteers to help us evaluate the impact of user diversity. Their heights ranged from $165cm$ to $180cm$, and the weights ranged from $60kg$ to $80kg$. The volunteers can be classified into three groups at different fitness levels, as listed in Table 2. The masters took at least one week to practice the target movements. The normal level volunteers are only familiar with some meta-movements and have workout habits in daily life. The novice ones have no workout habits and feel difficult to accomplish some combination movements.

Data Collection. Every volunteer was asked to provide a "cycle" of data at each time. A data collection cycle contains 18 fitness sessions, including 7 meta-movement sessions and 11 combination movement sessions. At the beginning of a fitness session, a volunteer moved randomly and did random activities such as stretching and playing on a smartphone for 10 to 30 seconds. Then, the volunteer moved to the target place and waited about 10 seconds. Next, the volunteer repeated a target movement 10 times. After that, the volunteer kept static for about 10 seconds, followed by other daily activities lasting 10 seconds.

Each session has four label sequences related to the four sub-tasks. *Labels of session detection* classify the fragments into three classes: unknown movements, static states, and fitness activities. *Labels of meta-movement classification* categorize the fragments into eight classes: seven meta-movement and intervals among movements. *Labels of state detection* divide the fragments into three classes: extension, retraction, and intervals among movements. *Labels of combination movement classification* classify the fragments into nineteen classes: eighteen target movements and intervals among movements. These label sequences have the same length as the number of fragments generated by the dynamic fragmentation.

We utilized a camera to record movements in parallel with the wireless devices. The start and end times of meta-movements, including those within combination movements, were marked manually according to the videos. Then, we used a script to automatically generate the label sequences for session detection, meta-movement classification, and combination movement classification according to the movement tables. Next, we leveraged NA-PSD to determine the final position of meta-movements like the way to determine the insertion place in Section 3.2. The part before the final position is labeled retraction, and the latter one is the extension.

Workflow of training and inference. In the training phase, we first trained a session detection model that can separate fitness sessions from daily activities. Then, we filtered the unknown movements out and built a new training dataset augmented by SVSG. Finally, we train the 2-stage combination movement recognition model in three steps: a) Training BRNN4MM and BRNN4SD in the first stage, respectively, b) Training the BRNN4CM in the second stage, and c) Fine-tuning the whole model.

In the inference phase, InFit first leveraged the session detection model to separate fitness sessions from other unknown activities. Then, InFit leveraged the 2-stage combination movement recognition model for meta-movement class sequence, state class sequence, and combination movement class sequence. From the outputs, InFit can know what type the movements are in a fitness session, how many times the user repeats, and how long each repetition lasts. Finally, InFit made a further step to assess the exercise quality of the session.

Baseline. We selected the work of Guo *et al.* [10] (FA), DeepSense [20], and EI [21] as baselines:

FA was a Wi-Fi-based fitness assistant designed for meta-movement recognition. This system used a thresholding-based method for segmentation. Then, it manually extracted features from the segments and designed a two-layer DNN for movement recognition.

DeepSense regarded activity recognition as a sequence-to-sequence classification problem. It cut the received signals into frames with a fixed size. Then, DeepSense adopted an autoencoder to compress those frames and a CNN to extract the discriminative features of each frame. Finally, an LSTM was applied to learn the temporal dependencies between frames for activity recognition.

EI also converted the received signals into sequences of frames. For an environment-independent system, it leveraged a CNN for activity recognition and a domain discriminator to recognize the environment. In this paper, we used the activity recognizer (the CNN) for comparison to observe the recognition performance only with learning the spatial features.

4.2 Evaluation Metrics

We evaluate the performance of InFit by *F1-Score* and *counting score* and discuss the effectiveness of data augmentation by the *gain of svsg*:

F1-Score. We use F1-Score to evaluate the accuracy of movement recognition: $F = 2 \times \frac{precision \times recall}{precision + recall}$. $precision$ is $\frac{N_T}{N_T + N_T'}$, indicating the confidence that a fragment can be recognized correctly. N_T is the number of fragments being recognized correctly. N_T' is the number of fragments belonging to other classes but mistaken as the target class. $recall = \frac{N_T}{N}$ is the probability of a fragment that can be recognized correctly. N is the total number of the fragment belonging to the target class.

Counting Score. Fitness assistance needs to count the number of fitness movements accurately. We can obtain the movement rate, which is the repetitions per minute, by calculating the quotient of the movement numbers and the related duration time. Then, it is possible to estimate the real power consumption according to movement

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3209656

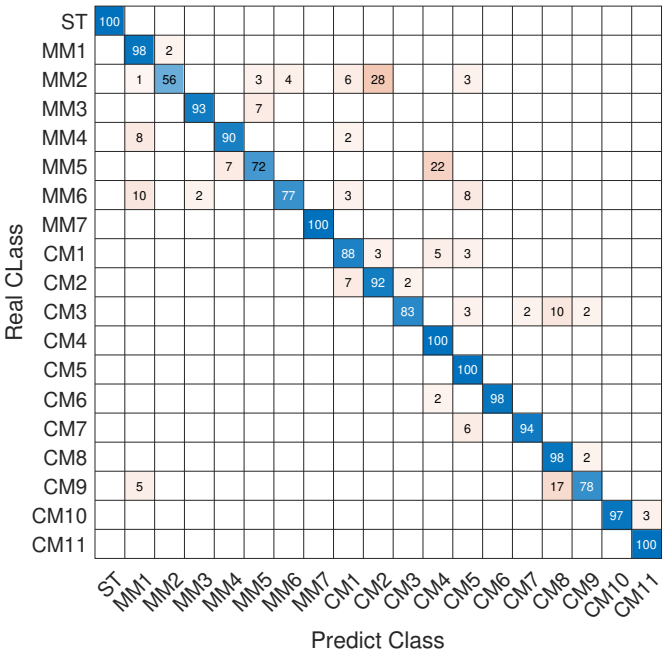


Fig. 7: Confusion matrix of movement recognition

types and rates. To evaluate the performance of movement counting, we define the counting score calculated as $C = 1 - (\frac{1}{N_s}\sum_{i=1}^{N_s}|\bar{n}_i - n_i|)$. N_s is the number of samples, n_i is the true movement number of the i-th sample, and $\bar{n}_i$ is the number of detected movements. $\frac{1}{N_s}\sum_{i=1}^{N_s}|\bar{n}_i - n_i|$ indicates the probability of the occurrence of counting errors. If the counting error is 0, C has the max value of 1. If the counting error occurred too much, C will be close to 0 and even be a negative number.

Gain of SVSG. This work utilizes virtual samples to emphasize the performance of InFit. The question is what conditions must the number of virtual samples meet to have a positive impact on InFit. Too few samples make a slight contribution to InFit, while too many virtual samples can make InFit overfitting. Hence, we quantify SVSG-related enhancement through the gain of SVSG to help us figure out the optimal numbers of virtual samples. For a certain movement, let F_S represents the F1-Score of the condition that SVSG augmented the dataset, and F represents the condition without data augmentation. The gain (G) of SVSG is $G = (F_S - F)/F$.

4.3 Micro-evaluation

In this section, we first evaluate the overall performance on different movements. Then, we discuss the generalization ability of SVSG by evaluating the enhancement on the sub-tasks and the baselines.

4.3.1 Overall Performance

The first user was asked to provide 10 cycles in the dormitory room. Four cycles of data were selected to construct the training set, and four cycles of virtual combination movements were generated for data augmentation. Next, we selected three cycles of real data for cross-validation and the remained data for testing.

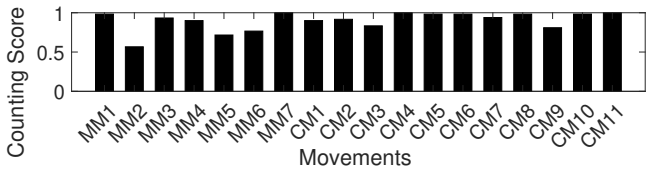


Fig. 8: Counting score of all movements

Fig. 7 illustrates the confusion matrix of the 18 target movements. InFit has an average accuracy above 90% on movement recognition. Specifically, the average accuracy of combination movement recognition is 94%, higher than the average accuracy of meta-movement 84%. It indicates that the BRNN4CM is robust to the inferences error of BRNN4MM and BRNN4SD. MM2 has a probability of 28% to be mistaken as CM2, and 22% of MM5s are mistaken as CM4. The reason may be due the same features shared by them since CM4 consists of MM4 and MM5.

CM9 is the concatenation of three MM1 and one MM3, and it also the concatenation of a MM1 and CM8. Therefore, InFit mistakes 17% CM9 as CM8 and 5% as MM1. It turns out to be that the continued repetitions may be recognize a single one. We can introduce other methods like peak detection to improve the accuracy of recognizing such combination movements with repetitions. The accuracies of CM6, CM10, and CM11 are close to 100%. It indicates that InFit can successfully distinguish the combination movements consisting of the same meta-movements in different orders.

Fig. 8 shows the counting score of different movements. The counting score distribution is similar to the recognition accuracy distribution since InFit doesn't count the movement when it is mistaken. This counting score can be improved through some simple calibration method. For example, in a short period, the repetitions belong to the same movement. Therefore, we can introduce a time window to help count the number.

4.3.2 SVSG's Performance on Sub-tasks

To evaluate the improvement brought by SVSG, we compare the performance on four sub-tasks under the conditions with different sizes of training sets. The third user was asked to provide ten cycles of data collected in the exhibition hall. The dataset was randomly divided into three parts for training, cross-validation, and testing in a ratio of 4:3:4. We generated four cycles of virtual combination movements from the training data to build different training sets. The training sets can be described as $X \times Cycle_{Real} + Y \times Cycle_{Virtual}$. Both X and Y are in the set of $0, 1, 2, 3, 4$.

As shown in Fig. 9, each column shows the F1-Score, counting score, and gain of SVSG of a sub-tasks. For sub-tasks of the session and state detection, the F1-Score can stay above 0.75 even without the enhancement of SVSG. Under the enhancement of SVSG, the F1-Score can exceed 0.9, and the counting score will ascend to 0.9. Surprisingly, InFit achieves comparable performance on these sub-tasks only with the help of virtual samples.

The results of meta-movement classification and combination movement classification have a similar trend. Their performance increases with the growth of data volume.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3209656

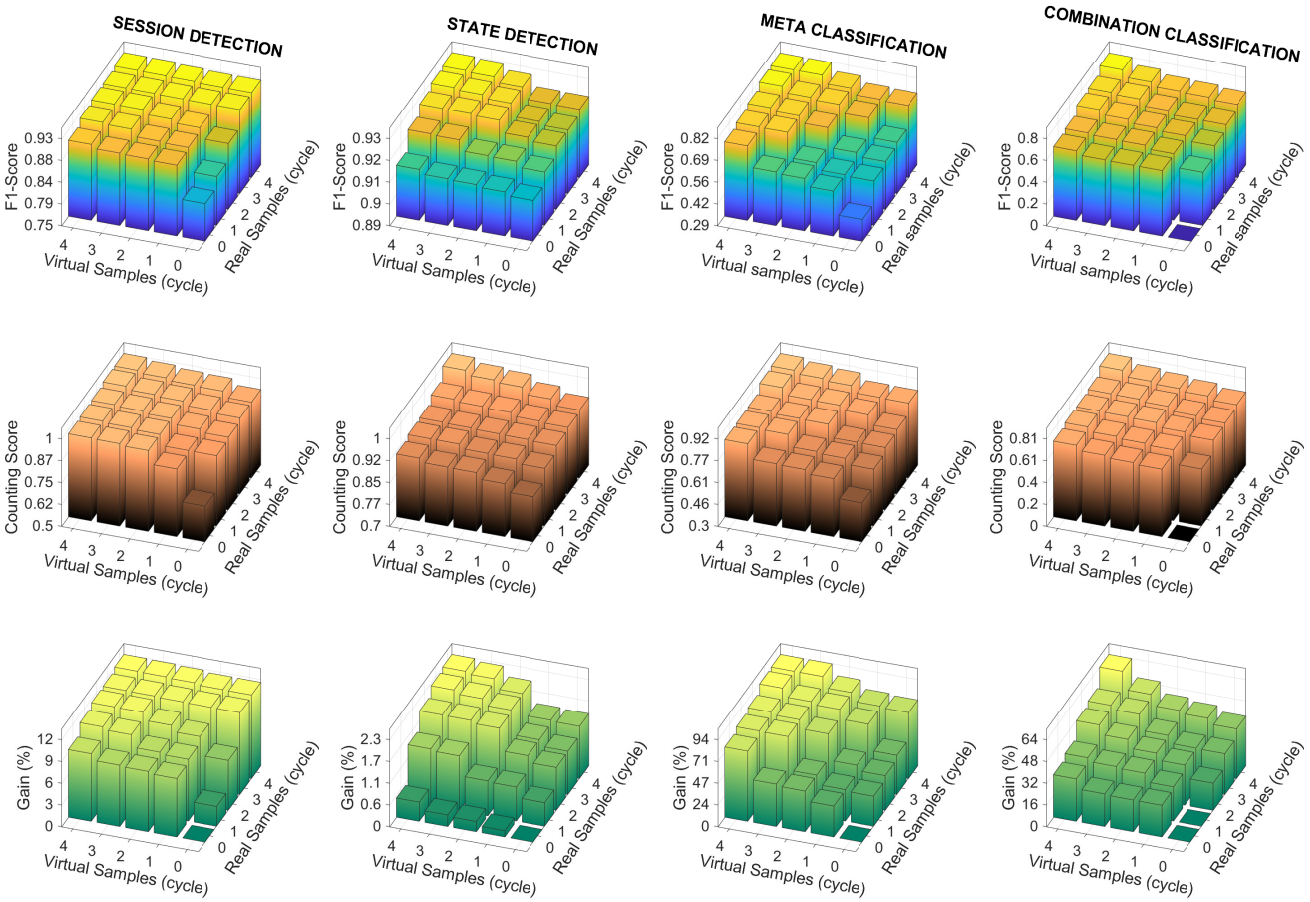
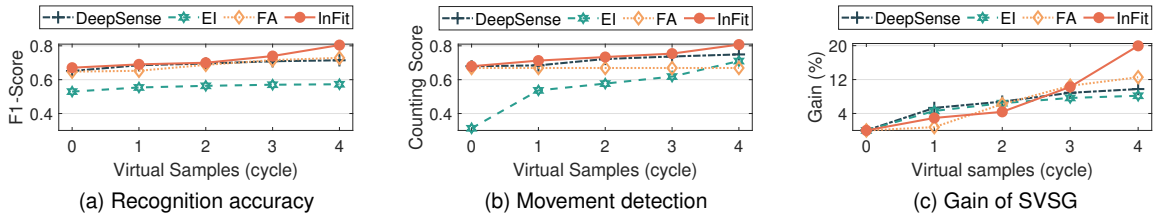


Fig. 9: Enhancement on the sub-tasks by SVSG. "Meta-movement Classification" and "Combination movement Classification" are written in a shorter format due to the limit of drawing space.



(a) Recognition accuracy (b) Movement detection (c) Gain of SVSG

Fig. 10: Baseline comparison and SVSG's generality

With the increase of virtual samples, the recognition accuracy and counting score increased more than 0.4. The gain of SVSG shows that these tasks benefit from the virtual samples much more than the other two sub-tasks. The reason is that virtual combination movements aim at providing sufficient knowledge about how meta-movements compose to combination movements, which makes limited contributions to session detection and state detection.

In summary, SVSG can enhance InFit in all the sensing tasks, and it reveals an opportunity for zero-effort combination movement recognition.

4.3.3 SVSG's Performance on the Baselines

In this section, we compared InFit with the baselines under the conditions with different volumes of virtual samples. We reused the datasets mentioned in the previous section. The cycle number of virtual sample was set ranging from 0 to 4. The sliding window of DeepSense and EI was set to $0.25s$ with no overlap.

As shown in Fig. 10a, the recognition accuracies of the four systems are proportional to the virtual sample cycles. InFit outperforms Deepsense, EI, and FA by 0.13, 0.4, and 0.1 in movement recognition when the virtual sample cycle is 4. Since CNN cannot learn the temporal dependencies between frames, EI has the lowest combination movement recognition accuracy among the baselines. The ascending trend also exists in movement detection, as shown in Fig. 10b. The detection accuracy EI has been significantly improved thanks to the sufficient information about the difference between moving and static states provided by the virtual samples. We should notice that SVSG does not contribute to FA on movement detection because FA uses

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3209656

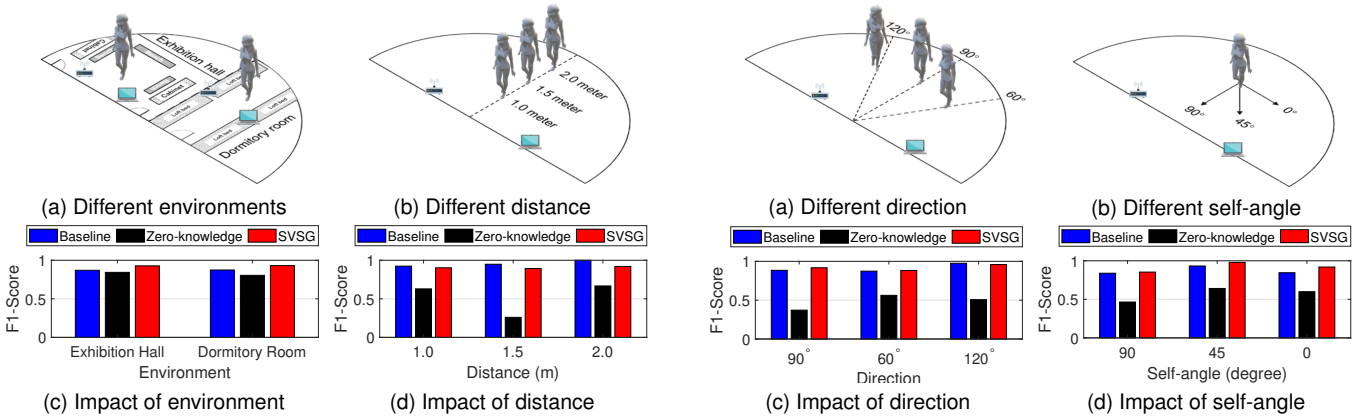


Fig. 11: Performance on different environments and distance

Fig. 12: Performance on different directions and self-angles

a simple threshold-based segmentation method. Fig. 10c shows that three systems have at least 0.08 improvement when there are four cycles of virtual samples. In summary, SVSG provides a general enhancement of combination movement recognition on all systems, and InFit gains the most benefits.

4.4 Macro-evaluation

This section first evaluates the robustness of InFit in different conditions, including the changes of environments, distances and directions to the devices, self-angle, and user diversity. In these experiments, a collection cycle includes the data of five movements, including MM1, MM2, MM3, CM1, and CM2.

In each experiment, we compared the results of three tests for evaluation. The first test was named "baseline" that we trained and tested local models under different conditions. The second test was named "zero-knowledge" observing the local models' performance in different conditions. The "SVSG" test was to observe the cross-condition performance when the local models were enhanced by SVSG. Every local model was fine-tuned by four cycles of virtual samples derived from the meta-movements in different conditions. The results of baseline, zero-knowledge, and SVSG are represented by the blue, black, and red bars in Fig. 11, respectively.

4.4.1 Impact of Environment Changes

Theoretically, InFit should have stable performance in different environments since it has filtered out the environmental noise. We asked a volunteer to provide 10 cycles of data in each environment as shown in Fig. 11a. Four cycles were used for training, three cycles were used for cross-validation, and the remained four cycles were testing data. As shown in Fig. 11c, the results of three test achieve an average accuracy higher than 0.8 in both conditions. It means that the preprocessing method has greatly reduced the impact of different environments.

4.4.2 Impact of Distance Changes

According to the free space propagation mode, like Friis mode, the received signal strength is inversely proportional to the propagation distance. When the distance increases, the power of the received signals will decrease due to signal attenuation, resulting in a descending trend of recognition accuracy. To evaluate the impact of different distances between the user and devices, we conducted three tests on three different distance conditions. As shown in Fig. 11b, the locations were vertical to the connection line between the transmitter and receiver. The distances between the locations to the connection line range from $1.0m$ and $2.0m$.

Fig. 11d illustrates the recognition accuracies under different distances. We observe that the baseline test has an average accuracy higher than 0.9 and an ascending trend within 2.0m. This ascending trend does not conflict with common sense that the recognition accuracy is inversely proportional to the distance due to signal attenuation. Within a certain distance, the strength of received motion-induced signals is strong enough, and signal attenuation has little influence. While for the systems that recognize movement based on the velocity features extracted from the motion-induced *Doppler Frequency Shift* (DFS), the DFS-based velocity estimation accuracy will have stronger influence. The more accurate the velocity estimation is, the better the recognition performance is. Theoretically, the velocity estimation accuracy is proportional to the distance. We refer the reader to the work proposed by Niu et al. [29] for a detailed analysis.

Compared with the baseline test, the result of zero-knowledge test are below 0.6. It means that InFit is sensitive to distance changes. Fortunately, the accuracy can be improved if the training set includes a small volume of the data collected under other conditions according to the result of SVSG test.

4.4.3 Impact of Direction Changes

Fig. 12a illustrates the different conditions of direction, including $60°$, $90°$, and $120°$. These three places are $1m$ away from the mid-point on the connection line between the transceivers. The volunteer at each place was asked to provide ten cycles of data and face the midpoint when exercising. Also, the datasets was divided into three sets for training, cross-validation, and testing.

The results of three tests are shown in Fig. 12c. The accuracies of the baseline test stay all above 0.85. It means that the motion-induce signals are distinguishable when the

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3209656

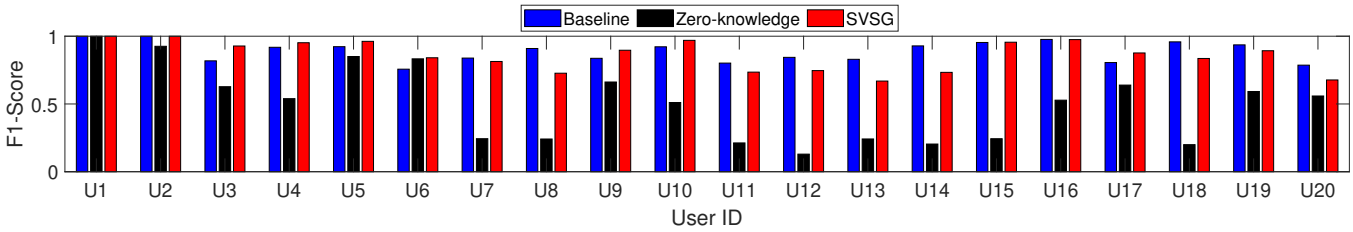


Fig. 13: Impact of user diversity

direction changes since the signals have similar SNR. However, the movements will induce different signal patterns. As shown by the result of zero-knowledge test, the accuracy may decline more than 0.4 compared with baseline. This degradation can be avoided if the user can provide a small volume of meta-movements in new directions.

4.4.4 Impact of Self-angle Changes

We further evaluated the impact of the self-angle. When people change self-angles, the same movement may cause different frequency offsets. As shown in Fig. 12b, the volunteer was asked to stay at the given place. There were three self-angle conditions, including $90°$, $45°$, and $0°$. The arrows represent the direction where the user faces.

As shown by the blue bars in Fig. 12d, the baseline has a stable accuracy higher than 0.8, indicating distinguishable signal patterns of different movements. According to the result of the zero-knowledge test, the self-angle impacts the motion-induced signal patterns. Also, InFit can be more robust to self-angle change by learning the knowledge under different conditions.

4.4.5 Impact of User Diversity

To evaluate the impact of user diversity, we conducted three tests on 20 volunteers. Every volunteer was asked to provide 10 cycles of data collected in the exhibition hall. Training data, cross-validation data, and testing data were in a ratio of 4:3:4. As shown in Fig. 13, for the result of the zero-knowledge test, InFit's performance will descend if directly apply it to recognize different users' movements. Fortunately, InFit can avoid this degradation by leveraging SVSG to augment training datasets. It learned the personal characteristics from the virtual samples derived from the meta-movements of other persons.

Besides, we notice that the average accuracy of the baseline tests is 0.87. The standard deviation is 0.08, which means that the accuracy varies a lot given different users. The reason underlies user diversity caused by movement proficiency. Accordingly, we calculated the average accuracies of the user at different fitness levels listed in Table 2. The average accuracies of the master, normal, and novice groups are 0.92, 0.87, and 0.77, respectively. Persons with rich fitness experience could be easier to keep every movement in a standard rhythm and intensity. Thus, the training and test datasets proposed by the masters have closer feature distributions compared with the datasets of the other two groups. In general, InFit can quickly learn the representative features of experienced users' movements and achieve a good recognition performance. In contrast, it requires more training data for the users with unstable movement patterns. Improving the robustness of user diversity is beyond the scope of this study and is left for future work.

4.4.6 System Overhead

Storage overhead: InFit is the first wireless-based fitness assistant system designed for combination movements. Its SVSG method can significantly reduce the storage overheads of combination movements. For the on-site application scenarios, InFit should have lower or comparable overheads compared to the state-of-the-art on meta-motion recognition tasks and show greater advantages on combination motion recognition tasks. We selected FA [10] as a comparison to evaluate InFit's storage costs on meta-movement recognition tasks. To recognize the 7 meta-movements described in Table 1, InFit and FA need 22MB and 7MB of memory to store their training data. Although InFit has a higher storage overhead, it is lightweight enough for *commercial-off-the-shelf* (COTS) intelligent devices such as smartphones and smart speakers. Then, we added the 11 combination movements to the recognition task. InFit's storage overhead only increases 3MB. This result implies that InFit's storage cost does not change significantly as long as the type of the meta action does not increase when the number of combination movements is increased.

Training time: We repeated the training phase 5 times to measure the average training time of the DNN-based models described in Section 4.1: the session detection model and the 2-stage combination movement recognition model. In each training phase, we set the training epoch to 50, and the learning rate was initialized to 0.003. For only the meta-movement recognition tasks, the session detection model needed 4 seconds for training. The 2-stage combination movement recognition model required 6.8 seconds, the sum training time of three components: BRNN4MM (2.6 seconds), BRNN4SD (2.7 seconds), and BRNN4CM (1.5 seconds). InFit took about 11 seconds to train its DNN-based models, much quicker than FA's training time, 77 seconds. The reason is that InFit divides the recognition task into multiple sub-tasks, which can limit the search space of parameters and reduce the computation complexity. After adding the combination movements, InFit took 11 and 20 seconds to train the two DNN-based models, indicating that InFit can be updated frequently to overcome accuracy reduction brought by the changes in user habits.

Runtime latency: InFit should show the assessment at least in soft real-time such that users can adjust their posture in time. We measured the average runtime latency of getting a report after finishing a fitness session to evaluate the potential of applying InFit to real-time application scenarios.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3209656

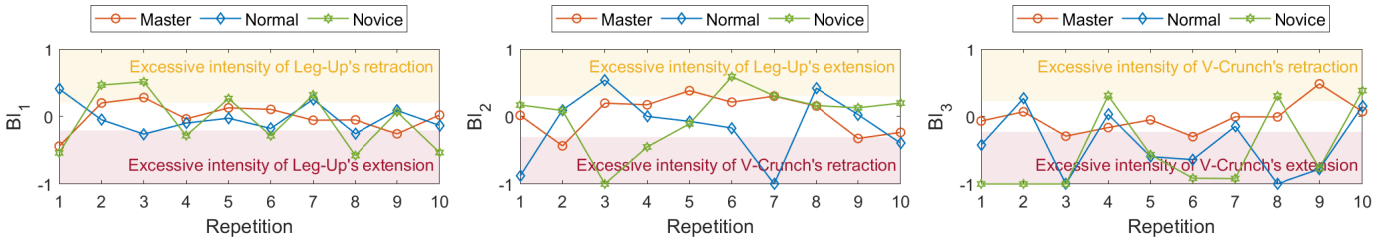


Fig. 14: Intensity of 10 CM1 repetitions across three volunteers

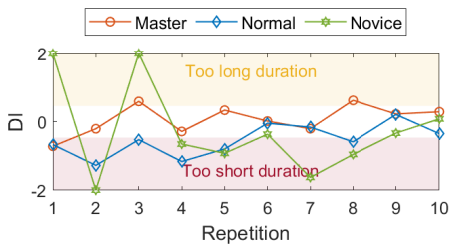


Fig. 15: Duration of 10 CM1 Repetitions

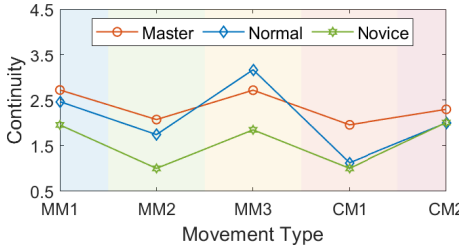


Fig. 16: Continuity of different sessions

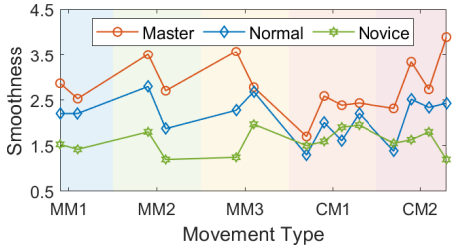


Fig. 17: Smoothness of different sessions

Specifically, we repeated the following steps 10 times: i) selected 10 random samples per movement, ii) intercepted 60 seconds of each sample, iii) test the latency of the intercepted samples, and iv) record the latency. Given a session lasting 60 seconds, the average latency is 4.077 seconds. After finishing a fitness session, a user usually takes a rest lasting more than 10 seconds, much longer than the runtime latency, demonstrating InFit's potential to meet the soft real-time requirement.

4.5 Fitness Assessment

Fitness assessment can tell how the completion quality of a fitness session is. Thus, users can adjust their movements based on the feedback of InFit. High completion quality means a low risk of injury and high exercise efficiency. This section demonstrates how InFit describes the completion quality. Specifically, according to the prior works [4], [10], [16], we first introduce four metrics to assess a fitness session from two perspectives, e.g., *local effects* and *global effects*. Then, we compare three volunteers at different levels to show the assessment results.

4.5.1 Local Effects

Local effects, including *intensity* and *duration*, describe the exercise quality of each repetition. *Intensity* reflects the energy expended of a repetition. Prior works define the *balance indicator* (BI) to reflect the intensity. Let I_r and I_e be the energy of retraction and extension, $BI = \alpha - I_e/I_r$, where α is the standard value of I_e/I_r. Thus, if BI is negative, a meta-movement may have an excessive intensity of retraction. Or, the extension part may have an excessive intensity when BI is positive. However, we cannot directly introduce this indicator to describe the intensity of combination movements since it has many movement states. In order to describe combination movements' intensity, we leverage a BI vector $\boldsymbol{BI} = (BI_1, BI_2, \ldots, BI_{m-1}$, where m is the number of movement states within a combination movement and $BI_i = \alpha_i - I_{i+1}/I_i$, indicating the energy balance between two adjacent movement states. I_i is the integral of the corresponding NA-PSD curve. *Duration* reflects the time spent on each repetition. If a repetition has a shorter duration than the standard, the user may have performed a fierce action, increasing the risk of injury. On the other hand, too long duration may be caused by inefficient exercise. We introduce a *duration indicator* (DI) $DI = d - \tau$, where d is the time spent on a repetition, and τ is the standard value.

4.5.2 Global Effects

Global effects, including *continuity* and *smoothness*, describe the overall quality of a fitness session. *Continuity* describes the consistency of the intervals between repetitions in a fitness session. For efficient training, the exercise should have reasonable movement rhythm control. To evaluate the quality of rhythm control, we adopt the kurtosis of intervals' duration as a metric. Let $\boldsymbol{R} = \{r_i\}_{i=1}^{k}$ be the set of intervals' duration within a fitness session. The continuity is $CI = Kurt(\{r_i\}_{i=1}^{k}) = \frac{\sum_{i=1}^{k}(r_i-\mu)^4}{(\sum_{i=1}^{k}(r_i-\mu)^2)^2} - 3 = \frac{\mu^4}{\sigma^4} - 3$, where μ and σ are the mean and standard deviation of $\boldsymbol{R}$. The larger C is, the better continuity a fitness session has. *Smoothness* reflects the consistency of the repetitions within a fitness session. The user should try to keep the repetitions' intensity as similar as possible. Identical to the continuity indicator, we adopt kurtosis to describe the smoothness. Suppose a session has n repetitions, and each repetition has m movement states. Thus, a repetition's energy expended can be represented by $(I_1, I_2, \ldots, I_m)$. We calculate the smoothness of this session as $SI = (Kurt(\boldsymbol{I_1}), Kurt(\boldsymbol{I_2}), \ldots, Kurt(\boldsymbol{I_m}))$, where $\boldsymbol{I_i} = \{I_(i,j)\}_{j=1}^{n}$. A higher value of $Kurt(\boldsymbol{I_i})$ means a better smoothness of the combination movement's corresponding part in this session.

4.5.3 Effect Evaluation

To demonstrate the performance of effect evaluation, we asked three volunteers, e.g., U2 (master), U5 (normal), and U6 (novice), at different levels to provide data of five

movements: MM1, MM2, MM3, CM1, and CM2. The master has taken days to practice the five movements before this experiment. The volunteer at the normal level only has experience in the meta-movements. The novice one has no habit of daily workouts. For each movement, a volunteer repeated ten times. We show the exercise quality from local and global aspects.

Fig. 14 illustrates the intensity of 10 CM1 repetitions across three volunteers. The colored areas represent the excessive intensity, and the white space around zero indicates the standard value. The master (shown by the red lines) has the most stable and standard intensity. The volunteer at the normal level (shown by the blue lines) has a good performance of BI_1. However, it lost stability in BI_2 and BI_3 due to the lack of experience in combination movements. The novice user has the poorest performance because he can not control his muscles well. As shown in Fig. 15, the volunteers at master and normal levels have a relative standard duration similar to the trend of intensity. The novice user usually fails to maintain a stable duration.

Next, we compared the continuity and smoothness of the volunteers. Fig. 16 illustrates the continuity, and we utilize colored areas to separate sessions of different movements. The master shown by the red line has the best performance. The volunteer at the normal level (the blue line) has a medium performance but a better continuity on MM3 than the master. It is because the normal user has experience in the meta-movements, making him able to maintain a similar rhythm to the master even better than him. The novice one has the poorest performance compared with the above two volunteers. This trend also exists in the smoothness, as shown in Fig. 17.

Users can intuitively know their exercise quality through these figures. According to the local effect, users can see whether they are in good condition and which parts of the combination movements need more practice. Then, from the figures of global effects, they can know whether the session is in a stable rhythm and whether the repetitions are performed well.

5 Conclusion

InFit is the first attempt to uncover and address the combination gesture recognition with low training costs in the modern high intensive fitness programs.

A virtual sample generation method, namely, SVSG, is proposed to reduce the requirement of data collection. SVSG generates virtual combination movements by stitching meta-movements together according to the composition rules, e.g., concatenation and insertion. We propose the 2-stage combination movement recognition model. It divides the combination movement recognition task into multiple sub-tasks to realize fine-grained fitness assessment. Experiments show that InFit achieves the average combination movement recognition accuracy of 94%. The recognition accuracy is 40% higher than the state-of-the-arts under the condition of zero-knowledge. Moreover, SVSG can enhance all the sub-tasks of InFit and provide a general enhancement to other schemes designed for similar tasks.

InFit is an initial exploration, and there are many interesting questions left for our future work. For example, how to improve the robustness of InFit Can we generate virtual samples to imitate different positions? Maybe we can remove the requirement of data collection with the help of other techniques, such as videos and game engines.

References

[1] Y. Feito, K. Heinrich, S. Butcher, and W. Poston, "High-intensity functional training (hift): Definition and research implications for improved fitness," *Sports*, vol. 6, no. 3, p. 76, 2018.

[2] W. R. Thompson, "Worldwide survey of fitness trends for 2018: the crep edition," *ACSM's Health & Fitness Journal*, vol. 21, no. 6, pp. 10–19, 2017.

[3] T. J. Gabbett and N. Domrow, "Relationships between training load, injury, and fitness in sub-elite collision sport athletes," *Journal of sports sciences*, vol. 25, no. 13, pp. 1507–1519, 2007.

[4] Y. Xie, F. Li, Y. Wu, and Y. Wang, "HearFit: Fitness Monitoring on Smart Speakers via Active Acoustic Sensing," in *Proceedings of the 40th Annual IEEE International Conference on Computer Communications (INFOCOM)*. IEEE, 2021.

[5] J. Mankoff and C. Harrison, "GymCam: Detecting, Recognizing and Tracking Simultaneous Exercises in Unconstrained Scenes," *ACM Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, vol. 2, no. 4, 2018.

[6] X. Guo, J. Liu, and Y. Chen, "FitCoach: Virtual fitness coach empowered by wearable mobile devices," in *Proceedings of the 36th Annual IEEE International Conference on Computer Communications (INFOCOM)*. IEEE, 2017.

[7] M. Radhakrishnan, D. Rathnayake, O. K. Han, I. Hwang, and A. Misra, "Erica: enabling real-time mistake detection & corrective feedback for free-weights exercises," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems (SenSys)*. ACM, 2020, pp. 558–571.

[8] F. Zhang, K. Niu, J. Xiong, B. Jin, T. Gu, Y. Jiang, and D. Zhang, "Towards a diffraction-based sensing approach on human activity recognition," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, vol. 3, no. 1, p. 33, 2019.

[9] N. Yu, W. Wang, A. X. Liu, and L. Kong, "Qgesture: Quantifying gesture distance and direction with wifi signals," *ACM Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, vol. 2, no. 1, p. 51, 2018.

[10] X. Guo, J. Liu, C. Shi, H. Liu, Y. Chen, and M. C. Chuah, "Device-free personalized fitness assistant using wifi," *ACM Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, vol. 2, no. 4, p. 165, 2018.

[11] E. Velloso, A. Bulling, and H. Gellersen, "MotionMA: Motion modelling and analysis by demonstration," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pp. 1309–1318, 2013.

[12] C. Chen, S. Member, R. Jafari, S. Member, and N. Kehtarnavaz, "Improving Human Action Recognition Using Fusion of Depth Camera and Inertial Sensors," *IEEE Transactions on Human-Machine Systems (THMS)*, vol. 45, no. 1, pp. 51–61, 2015.

[13] D. Morris, T. S. Saponas, A. Guillory, and I. Kelner, "RecoFit: Using a Wearable Sensor to Find, Recognize, and Count Repetitive Exercises," in *Proceedings of the SIGCHI Conference on Human Factos in Computing Systems (CHI)*, 2014, pp. 3225–3234.

[14] D. Stromback, S. Huang, and V. Radu, "Mm-fit Multimodal deep learning for automatic exercise logging across sensing devices," *ACM Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, vol. 4, no. 4, 2020.

[15] L. Xie, X. Dong, W. Wang, and D. Huang, "Meta-activity recognition: A wearable approach for logic cognition-based activity sensing," in *Proceedings of the 36th Annual IEEE International Conference on Computer Communications (INFOCOM)*, 2017.

[16] H. Ding, L. Shangguan, Z. Yang, J. Han, Z. Zhou, P. Yang, W. Xi, and J. Zhao, "FEMO : A Platform for Free-weight Exercise Monitoring with RFIDs," in *Proceedings of the 13th Conference on Embedded Networked Sensor Systems (SenSys)*, 2015, pp. 141–154.

[17] N. Xiao, P. Yang, Y. Yan, H. Zhou, X.-Y. Li, and H. Du, "Motion-fi+: Recognizing and counting repetitive motions with wireless backscattering," *IEEE Transactions on Mobile Computing (TMC)*, 2020.

[18] Y. Xu, W. Yang, J. Wang, X. Zhou, H. Li, and L. Huang, "WiStep: Device-free Step Counting with WiFi Signals," *ACM Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, vol. 1, no. 4, pp. 1–23, 2018.

[19] L. Zhang, M. Liu, L. Lu, and L. Gong, "Wi-Run: Multi-runner step estimation using commodity Wi-Fi," in *Proceedings of the 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2018, pp. 1–9.

[20] H. Zou, Y. Zhou, J. Yang, H. Jiang, L. Xie, and C. J. Spanos, "Deepsense: Device-free human activity recognition via autoencoder long-term recurrent convolutional network," in *Proceedings of the 2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[21] W. Jiang, C. Miao, F. Ma, S. Yao, Y. Wang, Y. Yuan, H. Xue, C. Song, X. Ma, D. Koutsonikolas *et al.*, "Towards environment independent device free human activity recognition," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2018, pp. 289–304.

[22] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "SpotFi: Decimeter level localization using WiFi," in *Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM)*, vol. 45. ACM, 2015.

[23] K. Qian, C. Wu, Z. Zhou, Y. Zheng, Z. Yang, and Y. Liu, "Inferring motion direction using commodity Wi-Fi for interactive exergames," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, vol. 2017-May, 2017, pp. 1961–1972.

[24] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11 n traces with channel state information," *ACM SIGCOMM Computer Communication Review (SIGCOMM)*, vol. 41, no. 1, pp. 53–53, 2011.

[25] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of wifi signal based human activity recognition," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM, 2015, pp. 65–76.

[26] N. J. Willis, *Bistatic radar*. SciTech Publishing, 2005, vol. 2.

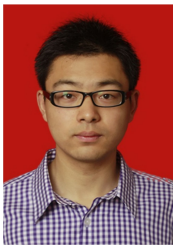
[27] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing (TIP)*, vol. 13, no. 4, pp. 600–612, 2004.

[28] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016, pp. 2818–2826.

[29] K. Niu, X. Wang, F. Zhang, R. Zheng, Z. Yao, and D. Zhang, "Rethinking Doppler Effect for Accurate Velocity Estimation with Commodity WiFi Devices," *IEEE Journal on Selected Areas in Communications (JSAC)*, pp. 1–16, 2022.

Huichuwu Li received a B.S degree in the school of Computer Science and Technology from Wuhan University of Science and Technology (WUST), Wuhan, China, in 2013. He is currently pursuing a Ph.D. degree in computer science and technology from Huazhong University of Science and Technology (HUST), Wuhan, China. His current research interests include wireless communications, indoor localization, scene analysis, and smart sensing.

Jiang Xiao is currently an associate professor in the School of Computer Science and Technology at Huazhong University of Science and Technology (HUST), Wuhan, China. Jiang received the BSc degree from HUST in 2009 and the Ph.D. degree from Hong Kong University of Science and Technology (HKUST) in 2014. She has been engaged in research on blockchain, distributed computing, big data analysis and management, and wireless indoor localization. Awards include Hubei Dawnlight Program 2018, CCF-Intel Young Faculty Research Program 2017, and Best Paper Awards from IEEE ICPADS/GLOBECOM.



Wei Wang received the Ph.D. degree from the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology. He is currently a Professor with the School of Electronic Information and Communications, Huazhong University of Science and Technology. His research interests include PHY/MAC design and mobile computing in wireless systems. He served on TPC of INFOCOM and GBLOBECOM. He served as an Editor for International Journal of Comparative Sociology, China Communications, and Guest Editor for Wireless Communications and Mobile Computing and the IEEE Communications Society, IEEE Multimedia Communications Technical Committee, and IEEE COMMUNICATIONS.

Lu Wang received the B.S. degree in communication engineering from Nankai University in 2009 and the Ph.D. degree in computer science and engineering from the Hong Kong University of Science and Technology in 2013. She is currently an Assistant Professor with the College of Computer Science and Software Engineering, Shenzhen University. Her research interests focus on wireless communications and mobile computing.

Dian Zhang received the PhD degree in computer science and engineering from the Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2010. After that, she worked as a research assistant professor at the Fok Ying Tung Graduate School, HKUST. From 2012 to 2019, she worked as an associate professor at Shenzhen university. She is currently an associate professor at Computer and Software Engineering, Shenzhen University. Her research interests include big data analytics and mobile computing. She is a member of the IEEE.

Hai Jin received the PhD degree in computer engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1994. He is a Cheung Kung scholars chair professor of computer science and engineering with the Huazhong University of Science and Technology (HUST), China. In 1996, he was awarded a German Academic Exchange Service fellowship to visit the Technical University of Chemnitz in Germany. He worked at The University of Hong Kong between 1998 and 2000, and as a visiting scholar with the University of Southern California between 1999 and 2000. He was awarded the Excellent Youth Award from the National Science Foundation of China, in 2001. He is the chief scientist of China Grid, the largest grid computing project in China, and the chief scientists of the National 973 Basic Research Program Project of Virtualization Technology of Computing System, and Cloud Security. He has coauthored 22 books and published more than 700 research papers. His research interests include computer architecture, virtualization technology, cluster computing and cloud computing, peer-to-peer computing, network storage, and network security.