# RF-URL: Unsupervised Representation Learning for RF Sensing

Ruiyuan Song, Dongheng Zhang, Zhi Wu, Cong Yu, Chunyang Xie, Shuai Yang, Yang Hu, Yan Chen*
School of Cyber Science and Technology, University of Science and Technology of China
Research Center from Data to Cyberspace, University of Science and Technology of China
Key Lab of Cyberspace Cultural Content Cognition, Communication and Detection, Ministry of Culture and Tourism

## ABSTRACT

The major obstacle for learning-based RF sensing is to obtain a high-quality large-scale annotated dataset. However, unlike visual datasets that can be easily annotated by human workers, RF signal is non-intuitive and non-interpretable, which causes the annotation of RF signals time-consuming and laborious. To resolve the rapacious appetite of annotated data, we propose a novel unsupervised representation learning (URL) framework for RF sensing, RF-URL, to learn a pre-training model on large-scale unannotated RF datasets that can be easily collected. RF-URL utilizes a contrastive framework to mind the gap between signal-processing-based RF sensing and learning-based RF sensing. By constructing positive and negative pairs through different signal processing representations, RF-URL seamlessly integrates the existing RF signal processing algorithms into the learning-based networks. Moreover, the RF-URL is carefully designed to take into account the asymmetric characteristics of different RF signal processing representations. We show that RF-URL is universal to a variety of RF sensing tasks by evaluating RF-URL in three typical RF sensing tasks (human gesture recognition, 3D pose estimation and silhouette generation) based on two general RF devices (WiFi and radar). All experimental results strongly demonstrate that RF-URL takes an important step towards learning-based solutions for large-scale RF sensing applications.

## CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; • **Computing methodologies** → **Machine learning**.

## KEYWORDS

RF sensing, unsupervised representation learning, contrastive learning, pre-training model

**ACM Reference Format:**
Ruiyuan Song, Dongheng Zhang, Zhi Wu, Cong Yu, Chunyang Xie, Shuai Yang, Yang Hu, Yan Chen. 2022. RF-URL: Unsupervised Representation Learning for RF Sensing. In *The 28th Annual International Conference On Mobile Computing And Networking (ACM MobiCom '22), October 17–21, 2022, Sydney, NSW, Australia.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3495243.3560529

*Corresponding author: Yan Chen (eecyan@ustc.edu.cn).


*ACM MobiCom '22, October 17–21, 2022, Sydney, NSW, Australia*

ACM ISBN 978-1-4503-9181-8/22/10…$15.00
https://doi.org/10.1145/3495243.3560529

## 1 INTRODUCTION

The past decade has witnessed much progress in RF sensing. Researchers have utilized advanced signal processing technologies to build explicit models between signal variations and human behaviors, which have enabled applications including vital sign estimation [1] and location tracking [2]. Nevertheless, when the sensing tasks become more complicated, the explicit signal processing models become intractable. Hence, learning-based RF sensing powered by deep neural networks has been an emerging field, which has enabled various applications including gesture recognition [51], human pose estimation [54] and identification [16], etc. While promising results have been achieved under specific conditions, it is still difficult to scale the data-driven system to large-scale RF sensing applications due to the dataset limitation.

**Challenge: Limitation on Annotated RF Datasets.** The data-driven RF sensing methods are typically achieved in a supervised learning manner. However, unlike RGB data which can be annotated manually, annotating RF data is time-consuming and laborious since RF signals are not human interpretable. Moreover, the captured RF signal is highly relevant to the environment for signal propagation, which forces researcher to collect a large-scale annotated dataset with various environments. To resolve such a problem, supervised RF data augmented with other modalities including RGB-cameras [52, 54] and accelerometers [27] has been proposed. However, the overhead introduced by synchronization and calibration between different modalities still limit the real-world deployments of these systems.

**Opportunity: Unsupervised Representation Learning.** The limitation on annotated RF datasets encourages us to exploit unannotated data for model training. URL [3] has attracted much interest in computer vision and natural language processing [14, 23]. Contrastive learning is one kind of URLs that have been growing rapidly recently in computer vision community, which repulses different images (negative pairs) while attracting the same image's different views (positive pairs). In such a way, the general semantic information cross different views corresponding to the same image is retained, while the rest information (noise) is thrown away, resulting in a useful representation. However, prior work has shown that contrastive learning tends to learn shortcut rather than meaningful information for RF signals [32]. To this end, we have noted that the core of contrastive framework lies in building positive and negative pairs to learn their inherent consistencies and discrepancies, while existing methods generally utilize data augmentation to construct the positive and negative pairs, which is designed for visual images but are not able to avoid shortcuts for RF signals.

**Insight: Minding the Gap Between Signal Processing and Neural Network.** To design an effective contrastive framework for RF signals, we find that the signal processing technologies, which could achieve diverse representations with theoretic signal

models, have been underexplored in the existing learning-based frameworks. For instance, given a series of RF signals, we can derive their Angle of Arrival (AoA)-Time of Flight (ToF) [48, 54], Doppler-Frequency-Spectrum (DFS) [51], etc. These different signal processing representations, together with the raw signal samples, are actually corresponding to the same semantic information, which can naturally form the positive pairs for the contrastive framework. On the other hand, the signal processing representations of different RF signals also naturally form the negative pairs.

In this paper, we introduce a new URL framework, RF-URL, for RF sensing. RF-URL utilizes contrastive learning by introducing different signal processing methods to replace data augmentation which is the common practice for computer vision. In such a way, we can seamlessly integrate the well-developed RF signal processing algorithms into the learning-based RF-URL networks. However, different signal representations have different characteristics (like dimensions, feature, etc), which is referred to "asymmetric characteristics". For instance, DFS is a 2D tensor (Frequency-Time) while AoA-ToF is a 3D tensor (AoA-ToF-Time). Such an asymmetric characteristic would make the contrastive learning framework difficult to converge. To this end, we design a series of modules to transform knowledge between different representations as follows.

- A multi-branch structure is designed to involve different signal processing representations obtained from the well-developed RF signal processing algorithms.
- A translator is utilized as a mediator to embed different signal representations of RF signals into a unified metric space to avoid convergence problem, and a predictor with stop-gradient operation is proposed to improve the performance of RF-URL.
- A memory bank stores all representations of the training dataset and can effectively sample a large-scale negative pairs.

**Contribution:** This paper takes an important step towards learning-based solutions for RF sensing applications by extending URL to solve the appetite for large-scale annotated RF data. The main contributions are summarized as follows:

- The paper introduces a novel URL framework, RF-URL, for RF sensing. To the best of our knowledge, this is the first work to utilize a contrastive framework to mind the gap between signal-processing-based RF sensing and learning-based RF sensing. By learning a general semantic information for various sensing tasks from different signal representations, the proposed RF-URL could enhance the sensing performance in an unsupervised manner.
- The paper presents an architecture for unsupervised RF sensing that leverages a multi-branch design, translator, memory bank and predictor with stop-gradient operation to achieve balance among simplicity, scalability and performance.
- We show that RF-URL is a universal framework to a variety of RF sensing tasks by evaluating it on three basic tasks with two kinds of RF signals (WiFi and radar), including (1) single label prediction task: human gesture recognition with WiFi signals; (2) structured prediction task: 3D pose estimation with millimeter wave radar signals; and (3) dense prediction: human silhouette generation with millimeter wave radar signals. Experimental results show that RF-URL pre-training model improves the performance of all three tasks: 8.98% accuracy improvement for human gesture recognition, 38.23% $l_2$ distance reduction for 3D pose estimation, and 11.33% IoU improvement for human silhouette generation.

## 2 METHOD

RF-URL is a URL framework for RF-based sensing based on contrastive learning. It utilizes different signal representations to construct positive and negative pairs for contrastive learning, which allows us to seamlessly integrate the well-developed RF signal processing techniques with URL framework.

As shown in Figure 1, the architecture of RF-URL mainly contains four components: signal representation, feature extraction with translation, predictor and memory bank sampling. Specifically, RF signal $x^i$ is firstly processed by different signal processing techniques to obtain different representations $(x_1^i, ..., x_n^i)$. Then, for each $x_k^i$, a backbone network $f_{\theta_k}$ is utilized to extract the corresponding feature $y_k^i$. Since different representations of RF signals are with different characteristics and dimensions, e.g., DFS is a 2D tensor (Frequency-Time) and AoA-ToF is a 3D tensor (AoA-ToF-Time), a translator network is adopted to map the RF signal features into a unified metric space with $z_k^i = g_{\theta_k}(y_k^i)$. Then, $(z_1^i, ..., z_n^i)$ together with the representations sampled from memory bank are utilized to construct positive pairs (different signal representations of the same RF signal) and negative pairs (the signal representations of different RF signals) to minimize an InfoNCE loss, which is supposed to be small for positive pair and large for negative pairs. In addition, a shared-weight predictor $h$ is applied on one branch to predict the output of another branch to further improve the representation quality. We will discuss these four components in detail as follows.

### 2.1 Signal Representation

In the field of RF sensing, various signal processing techniques have been developed to obtain the representations of the RF signals, channel state information (CSI), DFS, AoA-ToF, etc. However, since each of these signal representations only provides a certain perspective of the RF signals, directly utilizing these representations for learning-based RF sensing may introduce inductive bias and lead to unsatisfied solution [19]. For example, DFS mainly embodies the velocity information, due to which the inductive bias might enforce the network to over-weight the feature of DFS while ignoring other information, leading to over-fitting and low generalization performance.

RF-URL aims to exploit general semantic information for various sensing tasks from different signal representations. It is achieved by the contrastive learning theory through attracting different signal representations of the same RF signals while repelling others.

### 2.2 Feature Extraction and Translation

**Backbone Encoder:** The different signal representations of RF signals have different dimensions and characteristics, e.g., DFS is a 2D tensor (Frequency-Time) and AoA-ToF is a 3D tensor (AoA-ToF-Time). Thus, a customized multi-branch backbone network is adopt to process different signal representations.

**Translator:** The different signal representations may cause convergence issue of the RF-URL. Thus, a small MLP neural network, named as **translator**, is adopted as a mediator to transform the different representations of RF signals into a unified latent space

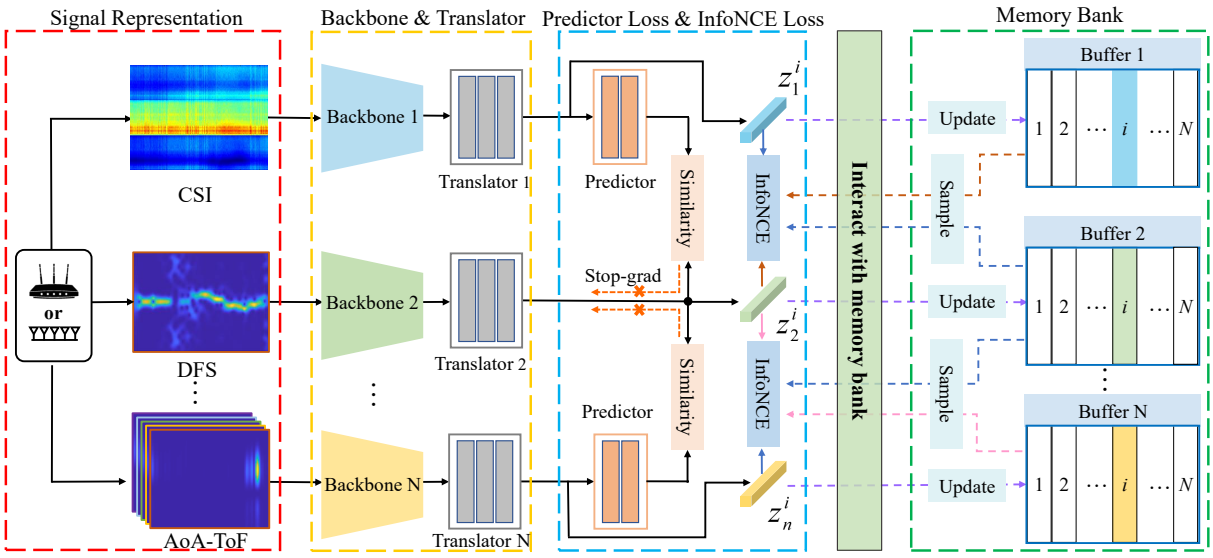


**Figure 1: An illustration of RF-URL pre-training mode for RF sensing.**

with $z_k^i = g_{\theta_k}(y_k^i)$. As shown in Figure. 2, we illustrate the training processing of RF-URL with/without translator, from which we can see that translator plays a crucial role on enforcing convergence. Overall, backbone is adopted to extract features of different signal

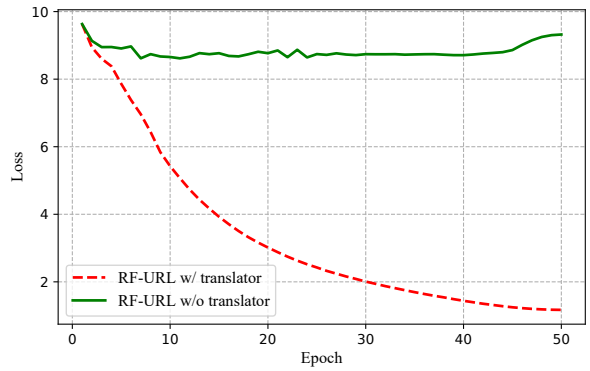


**Figure 2: Training RF-URL with/without translator.**

representations generated by different RF signal processing algorithms. Translator is utilized as a mediator to embed different signal representations of RF signals into a unified metric space to avoid convergence problem.

## 2.3 Predictor and Loss Function

**Predictor** is a small shared-weight neural network $h$ with stop-grad (stop gradient) operation, which is applied on one branch to predict the output of another branch to further improve the representation quality. Stop-grad could avoid a direct interaction between two branches, which prevent training collapsing. The loss is calculated as

$$\mathcal{L}_P^i = \frac{1}{2(n-1)} \sum_{k=1}^{n-1} \left[ \mathcal{D}(h(z_k^i), sg(z_{k+1}^i)) + \mathcal{D}(sg(z_k^i), h(z_{k+1}^i)) \right], \quad (1)$$

where $\mathcal{D}(.)$ indicates a distance metric and $sg(.)$ stands for stop gradient operation.

RF-URL is a contrastive-learning-based framework that learns features of RF signals from the positive and negative pairs created through different signal processing methods. A contrastive loss [22] is low when $z_k$ is similar to $z_{k+1}$ for positive pairs $\{x_k^i, x_{k+1}^i\}$ and dissimilar for negative pairs $\{x_k^i, x_{k+1}^j\}(i \neq j)$. With similarity function $s(\mathbf{u}, \mathbf{v}) = \mathbf{u}^\top \mathbf{v} / \|\mathbf{u}\| \|\mathbf{v}\|$, a contrastive loss InfoNCE [40] with $K$ negative pairs $\{x_k^i, x_{k+1}^j\}_{j=1}^K$ is written as

$$\mathcal{L}_c^i(z_k^i, z_{k+1}^i) = -\log \frac{\exp(s(z_k^i, z_{k+1}^i)/t)}{\exp(s(z_k^i, z_{k+1}^i)/t) + \sum_{j=1}^K \exp(s(z_k^i, z_{k+1}^j)/t)}, \quad (2)$$

where $t$ denotes temperature parameter [44]. However, $\mathcal{L}_c^i(z_k^i, z_{k+1}^i)$ defines an asymmetric loss by fixing $x_k$, i.e., $\mathcal{L}_c^i(z_{k+1}^i, z_k^i) \neq \mathcal{L}_c^i(z_k^i, z_{k+1}^i)$. Similarly, we define $\mathcal{L}_c^i(z_{k+1}^i, z_k^i)$ by fixing $x_{k+1}$ and obtain a symmetrical contrastive loss as follows

$$\mathcal{L}_c^i = \frac{1}{2(n-1)} \sum_{k=1}^{n-1} \left[ \mathcal{L}_c^i(z_k^i, z_{k+1}^i) + \mathcal{L}_c^i(z_{k+1}^i, z_k^i) \right]. \quad (3)$$

The final loss function for RF-URL is

$$\mathcal{L}_{RF-URL} = \sum_i \mathcal{L}_c^i + \lambda \sum_i \mathcal{L}_P^i, \quad (4)$$

where $\lambda$ is the scale factor. It is noticed that both $\mathcal{L}_c^i$ and $\mathcal{L}_P^i$ only calculate the distances between consecutive processed signals ($z_k$ and $z_{k+1}$) rather than all pairs. Compared with directly calculating distances between all pairs, calculating distance between consecutive signals could maintain the consistency between all pairs while reducing the computational complexity.

## 2.4 Memory Bank

Memory bank stores all representations of the training dataset [44]. Therefore, we can effectively sample a large-scale negative

samples. As shown in Figure. 1, the memory bank mainly contains two operations: sampling representations from buffers to calculate contrastive loss $\mathcal{L}_c^i$ and updating the representations in buffers.

**Sampling:** We adopt a cross sampling strategy. For representation $z_k^i$, a mini-batch of samples $\{z_{k+1}^j\}_{j=1}^K$ are randomly sampled from buffer $k+1$ to form the negative pairs $\{(z_k^i, z_{k+1}^j)\}_{j=1}^K$ and positive pair $(z_k^i, z_{k+1}^i)$. The same operation is executed for representation $z_{k+1}^i$ to obtain negative pairs $\{(z_{k+1}^i, z_k^j)\}_{j=1}^K$ and positive pair $(z_{k+1}^i, z_k^i)$. Then, the contrastive loss $\mathcal{L}_c^i$ in Eqn.(3) can be calculated to update parameters of neural networks by back-propagation algorithm.

**Update buffer:** The representations in memory bank could not be updated by back-propagation process, and a dynamically update strategy is adopted to update the parameters of memory bank with a momentum mechanism

$$z^{new} \leftarrow m \cdot z + (1-m) z^{old}, \tag{5}$$

where $m \in (0, 1)$ is a momentum coefficient, $z$ is the output of the translator, and $z^{old}$ comes from the memory bank.

## 3 RF SENSING TASKS

As shown in Figure. 3, in this paper, we demonstrate the universality of RF-URL framework through three different RF sensing tasks including human gesture recognition with WiFi signals, 3D pose estimation with millimeter wave radar signals, and human silhouette generation with millimeter wave radar signals.

- **Human gesture recognition** is a *single label prediction* task which utilizes a classifier after the pre-trained backbone of RF-URL to classify different gestures.
- **3D pose estimation** is a *structured prediction* task that estimates human skeletons by adding a regression module after the pre-trained backbone of RF-URL.
- **Human silhouette generation** is a *dense prediction* task that generates a semantic segmentation of human by adding a decoder module to the back-end of the pre-trained backbone of RF-URL.

Since almost all RF sensing tasks can be seen as a combination of above three tasks, and two most widely used RF signals are WiFi and radar signals, with the above three RF sensing tasks, it is sufficient to demonstrate the universality of the RF-URL framework.

### 3.1 Signal Processing

While there are many different signal processing algorithms which can be utilized to produce different signal representations, without loss of generality, we mainly utilize AoA-ToF and DFS in this paper. Specifically, as shown in Figure. 3, for human gesture recognition, we adopt two different signal processing algorithms (AoA-ToF and DFS) to produce two signal representations as the input of neural network. For human 3D pose estimation and silhouette generation, two perpendicular radars have been deployed for data collection, which have captured human information from two different views. In such a case, we can simply adopt one signal processing algorithm (AoA-ToF) to naturally generate two signal representations from the captured data of two radars.

*3.1.1 AoA-ToF.* Considering the signal transmitted and reflected from AoA $\theta$ and ToF $\tau$, the relative phase shift of this signal on adjacent antennas is $\Phi(\theta) = exp\{-j2\pi f_k \frac{d\cos\theta}{c}\}$, where $d$, $f_k$ and $c$ denote the space interval between two adjacent antennas, signal frequency and the speed of light. The phase shift on adjacent frequencies is $\Phi(\tau) = exp\{-j2\pi\Delta f\tau\}$, where $\Delta f$ denotes the difference between adjacent frequencies. By compensating the phase shift and adding the signals on different antennas and frequencies, the signals from AoA $\theta$ and ToF $\tau$ would superimpose coherently while the signals from other locations would be suppressed. Hence, the signals from that AoA-ToF could be separated, and the extracted signal [48, 49] can be expressed as

$$P(\theta, \tau) = \sum_{m=0}^{M} \sum_{k=0}^{K} s_{m,k} e^{j2\pi f_k \frac{md\cos\theta}{c}} e^{j2\pi k\Delta f\tau}, \tag{6}$$

where $s_{m,k}$ denotes the received signal, $m$ and $k$ denote the index of receiver antenna and signal frequency.

Since typical frequency modulated continuous wave (FMCW) radars could perform signal transceiving over large bandwidth with multiple input multiple output (MIMO) antenna array, AoA-ToF representations from radar could achieve much higher spatial resolution with larger data size compared with that of WiFi. By contrast, WiFi devices do not need to perform frequency sweeping, which leads to higher frame rate compared with radar. The output of the algorithm is a matrix of dimension $G_1 \times G_2$. For FMCW radar, $G_1 = 160$ and $G_2 = 200$. For WiFi, $G_1 = 96$ and $G_2 = 96$, which is smaller due to its lower spatial resolution.

*3.1.2 DFS.* Following the literature [51], the signal processing pipeline of DFS includes three steps: **(1)** We first perform conjugate multiplication on CSI of two antennas to remove random offsets; **(2)** We perform Principal Component Analysis (PCA) algorithm on the CSI stream to extract human reflections and reduce the data dimension; **(3)** We perform short-time Fourier transform (STFT) on the processed data to extract Doppler information. The DFS output is a frequency-time matrix of dimension $(121, 1024)$.

### 3.2 Basic settings

Unless specified, the following settings are used for RF-URL.

- **Translator** has batch normalization (BN) applied to each fully-connected (FC) layer with ReLU activation function, and the output FC has no ReLU with 128-dimension (128-d). This MLP has 3 layers with hidden size 1024-d.
- **Predictor** has BN applied to each FC layer with ReLU activation function. This MLP has 2 layers with hidden size 1024-d. The cosine distance is used for the distance metric in Eqn.(1) as follows

$$\mathcal{D}(z_1, z_2) = -\frac{z_1}{\|z_1\|_2} \cdot \frac{z_2}{\|z_2\|_2}, \tag{7}$$

  where $\|\cdot\|_2$ is $\mathcal{L}_2$ normal. The predictor works only in first five epochs and is closed afterwards during pre-training stage. The $\lambda$ in Eqn. (4) is 1.0 by default.
- **Optimizer:** We use stochastic gradient descent (SGD) optimizer with a cosine decay schedule and a warm up of 10 epochs. We also find a large initial learning rate, e.g., 0.6, can work well and produce better results. This is because RF-URL is based on memory bank, which requires a large learning rate to ensure the backbone to be adapted to the stored representations. The weight decay is 0.0001 and the SGD momentum is 0.9.

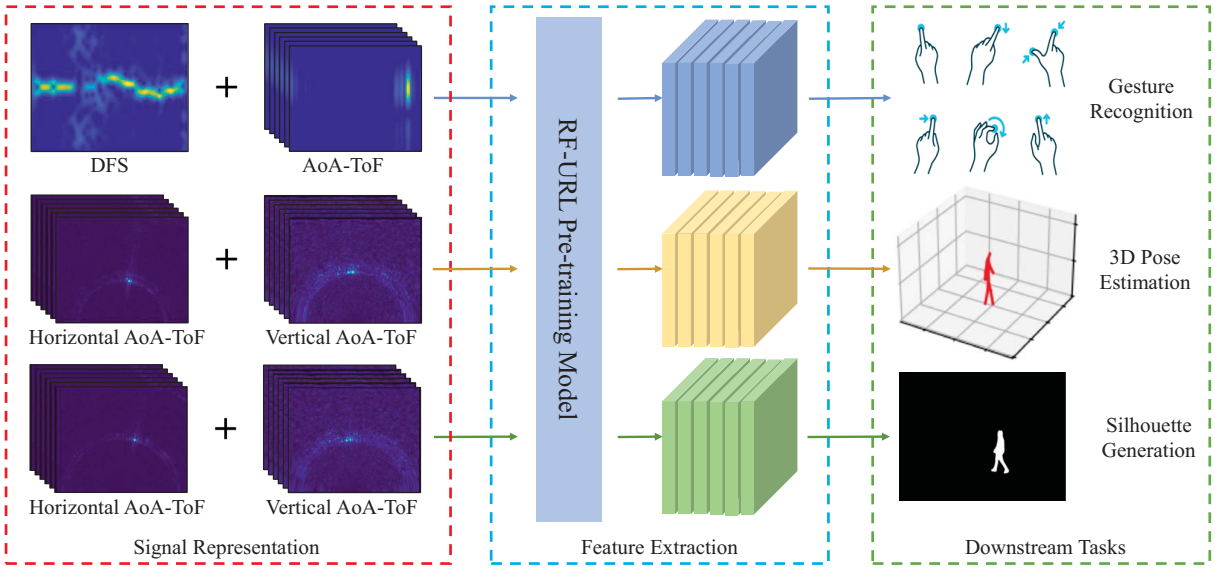


**Figure 3: An illustration of RF-URL for gesture recognition, 3D pose estimation and silhouette generation.**

- **Device:** All experiments runs on 4× NVIDIA Tesla V100 GPU(32GB) with PyTorch [34] library. All BN layer are replaced by Sync BN.
- **Hyperparameters:** The temperature parameter $t$ in Eqn. (2) is set as 0.07, and the momentum coefficient for memory bank update in Eqn. (5) is set as 0.5. The negative pairs $K$ for calculating InfoNCE loss in Eqn. 2 is 4096. The batch size of the pre-training stage is 256 by default.
- **Evaluation protocol:** We evaluate the RF-URL performance under both feature frozen setting and fine-tuning setting by freezing/ fine-tuning the backbone (initialized from RF-URL) and train a sub network from scratch. The evaluation metrics are classification accuracy, average $l_2$ distance between predicted keypoints and their ground-truth, and the average intersection-over-union (IoU) between the generating silhouette and ground-truth.

# 4 HUMAN GESTURE RECOGNITION

## 4.1 Dataset

RF-URL is designed to learn a general feature representation in an unsupervised manner using pre-training dataset without labels. Then training dataset with labels is utilized to evaluate the effectiveness of RF-URL. In practical deployment, it requires less overhead to collect unannotated data, and it is more common that the pre-training dataset is much larger than the annotated training dataset. Thus, we use a public dataset Widar3.0 [51] to evaluate the RF-URL framework for human gesture recognition, where two types of human gesture dataset are collected.

The first dataset (non number dataset) collects the widely used hand gestures for human-computer interaction, which contains 38687 samples. Due to the difficulty in constructing large-scale annotated RF dataset, it is unpractical to pre-train the model in a supervised manner. Thus, although the Widar3.0 is annotated, to simulate the real scenario, we remove the labels of first dataset to create the **pre-training dataset** for RF-URL to extract the general representation.

The second dataset (number dataset) collects some complex and semantic gestures that draw number 0-9 in the horizontal plane with a total of 5000 samples.

We perform a dynamic link selection (DLS) algorithm [51] for Widar3.0 to prune those WiFi receivers that may potentially be blocked by human torso and use the rest of the devices for human gesture recognition. Each sample in dataset 1 and dataset 2 contains 6 links, which means that we can get $6 \times 38687 = 232002$ samples for dataset 1 and $6 \times 5000 = 30000$ samples for dataset 2. After performing the DLS algorithm, we obtain 143255 samples for dataset 1 and 23574 samples for dataset 2. The dataset 2 are randomly split into **training dataset** (21335 samples) and **validation dataset** (2239 samples) with a ratio of 0.9:0.1.

## 4.2 Baseline

*4.2.1 Backbone Network.* We adopt ResNet [25] as the backbone followed with translator and predictor. Since there are two different basic block of ResNet, e.g., BasicBlock for ResNet-18/34 and Bottleneck for ResNet-50/101/152, to avoid potential inconsistency, we unify the basic block with Bottleneck and propose a new structure ResNet-17/35, i.e., ResNet-17: [1, 1, 2, 1] and ResNet-35: [2, 3, 3, 3] for Bottleneck [conv2_x, conv3_x, conv4_x, conv5_x] in [25], as an alternative version of ResNet-18/34. Given that the backbones of RF-URL for human gesture recognition are dual-branch, we split ResNet into two part by halving channels, e.g, a convolution layer with 256 $3 \times 3$ filters is split into two convolution layers with 128 $3 \times 3$ filters.

*4.2.2 Training Details.* **(1) Pre-training:** We perform pre-training on Widar3.0 pre-training dataset with a total training number of 150 epochs. **(2) Fine-tune:** The linear classifier combined with a frozen or fine-tuned backbone is trained on training dataset with

50 epochs and evaluated on validation dataset. The batch size is 128. For frozen setting, the initial learning rate is 30. For fine-tune setting, the learning rates of backbone and classifier are 0.003 and 0.03, respectively.

## 4.3 Experimental Results

*4.3.1 Classification Accuracy.*

**Comparison study:** The performance of different models on the human gesture recognition accuracy is illustrated in Table. 1. We can see that with the RF-URL pre-training, the accuracy of all backbones are all improved, and 94.060% accuracy can be achieved for ResNet-152 with fine-tuned backbone. The highest accuracy for training from scratch is 89.326% obtained by ResNet-152, which is however even lower than the lowest accuracy of RF-URL pre-training method, i.e., 91.201% obtained by fine-tuned ResNet-17.

From Table. 1, we can also see that with the ResNet-50 backbone, a carefully designed RF-URL can improve the performance with 7.995%, achieving 97.008% accuracy, which is 4.108% higher than Widar3.0 [51]. These results show that our RF-URL pre-training model can extract general information for gesture recognition from dataset 1 and apply it to dataset 2.

**Table 1: The performance of different models on the human gesture recognition accuracy.**

| Pre-training | Method | Accuracy |
|---|---|---|
| - | EI[28] | 80.0 |
| | Widar3.0[51] | **92.9** |
| - | ResNet-17 | 86.780 |
| | ResNet-35 | 88.656 |
| | ResNet-50 | 89.013 |
| | ResNet-101 | 89.058 |
| | ResNet-152 | **89.326** |
| RF-URL (Fine-tune) | ResNet-17 | 91.201 (+4.421) |
| | ResNet-35 | 92.363 (+3.707) |
| | ResNet-50 | 92.631 (+3.618) |
| | ResNet-101 | 93.301 (+4.243) |
| | ResNet-152 | **94.060** (+4.734) |
| RF-URL (Details) | ResNet-50 (baseline) | 89.013 |
| | + RF-URL(frozen) | 92.229 (+3.216) |
| | + Predictor | 92.407 (+0.178) |
| | + Fine-tune | 92.631 (+0.224) |
| | + 3D CNN | 84.323 (-8.308) |
| | + feature in translator | 96.784 (+12.461) |
| | + Shuffle BN | **97.008** (+0.224) |

**Representation quality:** Table. 2 shows the results of frozen feature setting with backbone initialized randomly (denoted as "Random init") or from RF-URL (denoted as "RF-URL(Frozen)"). We can see that pre-training using RF-URL learns better representations, which deliver a maximum improvement of 70.334% for ResNet-101 and a minimum improvement of 44.975% for ResNet-17.

**Accuracy vs parameters:** As shown in Figure. 4, as the increase of neural network parameters, both fine-tuning and learning from scratch methods achieve better performance. However, the accuracy of learning from scratch method increases slowly and then tends to be stable. The fine-tuning method is more benefit from the larger models with a rapid and continuous increasing trend.

**Table 2: Evaluate the accuracy of RF-URL for human gesture recognition under frozen feature setting with random initialization or RF-URL.**

| Model | Parameters | Random init | RF-URL (Frozen) |
|---|---|---|---|
| ResNet-17 | 11.18M | **46.539** | 91.514 |
| ResNet-35 | 21.85M | 34.971 | 91.603 |
| ResNet-50 | 25.55M | 28.093 | **92.407** |
| ResNet-101 | 44.54M | 21.617 | 91.961 |
| ResNet-152 | 60.19M | 22.778 | 92.095 |

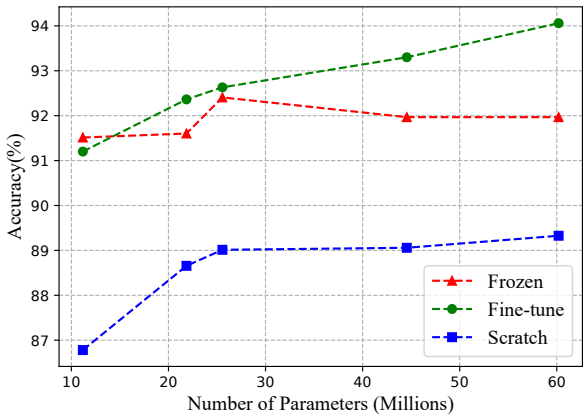


**Figure 4: Evaluation of RF-URL for human gesture recognition under different parameters and settings.**

*4.3.2 Different Annotated Samples.* Table. 3 shows the performance of ResNet-17/50/152 with different annotated samples. The training dataset of Widar3.0 are randomly sampled *n*% of samples, e.g., 100%labels (21335 samples), 50% (10667samples) and 10% (2133 samples). All results are evaluated on Widar3.0 validation dataset.

**Table 3: Evaluate the accuracy of RF-URL with different annotated samples, Frozen and Fine-tune are backbone initialization from RF-URL.**

| Model | Pre-training | 100%labels | 50%labels | 10%labels | 0%labels |
|---|---|---|---|---|---|
| ResNet-17 | - | 86.780 | 82.269 | 65.699 | 10.540 |
| | Frozen | **91.514** | **89.549** | **82.314** | 10.808 |
| | Fine-tune | 91.201 | 84.591 | 63.510 | - |
| ResNet-50 | - | 89.013 | 84.815 | 64.448 | 11.121 |
| | Frozen | 92.407 | **90.621** | **83.519** | 10.630 |
| | Fine-tune | **92.631** | 90.174 | 71.103 | - |
| ResNet-152 | - | 89.326 | 84.323 | 61.411 | 10.585 |
| | Frozen | 92.095 | 90.889 | **84.323** | 9.558 |
| | Fine-tune | **94.060** | **91.157** | 72.086 | - |

Overall, as the annotated samples increase from 0% to 100%, the performance improves but the improvement decreases. Compared

with training from scratch, RF-URL can achieve significant performance improvement with fewer annotated samples. For example, the performance of RF-URL pre-training models trained on 50% of training dataset is higher than the model trained form scratch using 100% of training dataset. Even though only using 10% of training dataset, RF-URL pre-training model also achieves a remarkable performance of 82.314%, 83.519% and 84.323%, which demonstrates the effectiveness of RF-URL. We should notice that RF-URL pre-training model only shows the effectiveness when combined with a fine-tune process. Without the training dataset, i.e., 0% training dataset, RF-URL only obtains about 10% accuracy, which behaves like a random classifier. This is because the RF-URL only extracts the general feature representations in an unsupervised learning manner, which does not include the training process of classifier. The classifier works well only when it is fine-tuned using annotated dataset. Thus, we use 10% labels as the smallest annotated samples in the following human 3D pose estimation and silhouette generation task.

**Pre-training vs Scratch:** From Table. 3, we can observe that as the number of annotated training data decreases, the accuracy of all methods decreases, e.g. removing 90% of training dataset, ResNet-50 drops 24.565% (Scratch), 8.888% (Frozen) and 21.528% (Fine-tune). Compared with learning from scratch, the RF-URL pre-training models (both Frozen and Fine-tune) maintain relatively high accuracy, which demonstrates that RF sensing task could be benefit from the RF-URL pre-training models, even in the condition of limited annotated samples.

**Frozen vs Fine-tune:** We also observe that RF-URL with fine-tuned backbone achieves the best performance located at the lower left corner of the Table. 3. Thus, RF-URL with fine-tuned backbone is benefit from a larger model associated with large-scale annotated dataset. Nevertheless, frozen RF-URL is more stable w.r.t the change of models and annotated samples, which indicates that frozen RF-URL can be a general scheme with some performance loss, e.g., drops 1.965% for ResNet152 combined with 100% labels.

*4.3.3 Different Size of Pre-training Dataset.* Table. 4 reports the performance of ResNet-50 with different size of pre-training dataset. Overall, a smaller pre-training dataset has worse performance, which may be due to the over-fitting effect on the smaller pre-training dataset.

**Table 4: Evaluate the accuracy of ResNet-50 under different size of pre-training dataset.**

| Size | 100% | 80% | 60% | 40% | 20% | 0% |
|---|---|---|---|---|---|---|
| Frozen | 92.407 | 89.192 | 82.448 | 76.061 | 65.386 | 28.093 |
| Fine-tune | 92.631 | 92.586 | 89.951 | 84.949 | 84.055 | 84.011 |

*4.3.4 Ablation.* In this subsection, we conduct ablation studies to evaluate some important components of our RF-URL framework for gesture recognition.

**Predictor:** We ablate the predictor of RF-URL by using ResNet-17 with a total pre-training number of 100 epochs and a bigger batchsize 512. We adopt frozen RF-URL method to train the classifier. The results are shown in Table. 5, where epochs $n$ indicates that the predictor participates in training during epoch 0 to $n$ and is discarded afterwards. We get the highest accuracy 88.566% when predictor is only trained 5 epochs, which improves the accuracy of 1.027% compared to that without predictor. However, a longer trained predictor reduces the performance about 0.402% (epoch 50) compared to that without predictor. The results indicate that RF-URL could be benefit from a short trained predictor.

**Table 5: Predictor with different training epochs.**

| Epochs | 0 (w/o pred.) | 5 | 10 | 25 | 50 | 100 |
|---|---|---|---|---|---|---|
| Acc. | 87.539 | **88.566** | 87.673 | 87.271 | 87.137 | 88.164 |

**Backbone and representation extracted layer:** We ablate the dual branch of RF-URL with symmetrical backbone where both DFS and AoA-ToF use ResNet-50, and asymmetrical backbone where DFS and AoA-ToF use ResNet-50 and 3D-ResNet-50, respectively. Note that AoA-ToF is a 3D tensor (AoA-ToF-time) rather than a 2D tensor. Thus, in the asymmetrical backbone, ResNet-50 is replaced by 3D-ResNet-50 [18]. For the symmetrical backbone, a 3D tensor can be treated as multiple 2D tensors stacked over the channel dimension. Hence, we adopt the same ResNet-50 with different input channels numbers for 2D and 3D tensor. Since the translator plays the role of unifying the representation of ResNet-50 and 3D-ResNet-50 by transforming the feature of 2D tensor and 3D tensor into a vector, we also take the representation extracted layer in translator into consideration.

**Table 6: Evaluate the accuracy of RF-URL with different backbone and the representation extracted layer.**

| Models | Rep. in | Frozen | Fine-tune |
|---|---|---|---|
| ResNet-50 + ResNet-50 | layer-0 | 92.407 | 92.631 |
| | layer-1 | 90.531 | 92.720 |
| | layer-2 | 90.621 | 95.489 |
| ResNet-50 + 3D-ResNet-50 | layer-0 | 84.903 | 84.323 |
| | layer-1 | 87.628 | 85.753 |
| | layer-2 | **94.239** | **96.784** |

As shown in Table. 6, the asymmetrical backbone (ResNet-50 + 3D-ResNet-50) with the representation in layer-2 obtains the highest accuracy in both frozen (94.239%) and fine-tuning (96.784%) strategies. This result shows that RF-URL is benefit from a customized backbone for input RF signals.

Table.6 also reports that the asymmetrical backbone significantly decreases the accuracy in layer-0, compared with symmetrical backbone, which decreases 7.504% and 8.308% for frozen and fine-tune strategy respectively. This is because it is difficult for the classifier to handle the simply stacked 2D spatial and 3D spatio-temporal features.

By comparing layer-2 with layer-0, there is a great gain +9.336% (frozen) and +12.461% (fine-tune) in ResNet-50 + 3D-ResNet-50, which supports our hypothesis that translator plays the role of transforming information. It is also worth noting that when using a symmetrical backbone, representation in a deeper extracted layer (frozen with layer-1 or layer-2) might has a negative impact on performance. Therefore, a symmetrical backbone should use

layer-0 as representation layer for frozen strategy and layer-2 as representation layer for fine-tuning strategy.

**Shuffle BN vs Sync BN:** As similarly reported in [23], our empirical studies show that using Sync BN has a negative impact on performance. This is possibly because the intra-batch communication among samples leaks information. We solve this problem by shuffling BN [23] that trains with multiple GPUs and performs BN on the samples independently for each GPU. The results are shown in Table. 1, from which we can see that a shuffle BN mitigates the leaking information of BN, and improves the performance by 0.224% for a fine-tuned backbone.

# 5 3D HUMAN POSE ESTIMATION

## 5.1 Dataset

We collect a multi-modal dataset, RFP3D (RFPose3D), to evaluate the RF-URL for 3D human pose estimation. As shown in Figure. 5, RFP3D synchronously captures the millimeter wave radar signals by dual perpendicular TI MMWCAS-RF-EVM FMCW radars with an antenna array of 12 transmitters and 16 receivers, and the optical images by a 13-view camera system. The sweep ranges of dual FMCW radar are 77-78.23 GHz and 79-80.23 GHz respectively. The multi-camera system senses the target from different views to generate 3D keypoints using AlphaPose [17] associated with a triangulation process. The hardware system and data processing methods are similar to literature [54].

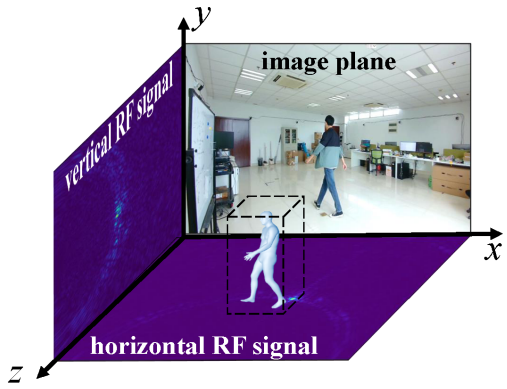


**Figure 5: RF signals and RGB image synchronous record.**

We collect data under 11 different conditions, including random walk without occlusion, random walk under occlusion (styrofoam, carton, yoga mat and dark) and random action (stand, walk, squat and sit). The RFP3D dataset contains three parts: pre-training dataset, training dataset and validation dataset. Pre-training dataset includes 149506 samples. Both training dataset and validation dataset include 25842 annnotated samples. We feed 10 frames of RF signal into neural network to generate the 3D keypoint of last frame.

## 5.2 Baseline

*5.2.1 Network Structure.* Our pose estimation network follows the design of RF-Extractor in RFGAN [47], which utilizes two RF encoding networks to extract human pose information from vertical and horizontal RF signal with a fusion module to combine the extracted information.

**Backbone network:** The RF encoding network utilizes 6 layers of $5 \times 5$ convolutions with strides 2 and padding 2. The channels of 6 convolution layers are $[10\alpha, 5\alpha, 16\alpha, 32\alpha, 128\alpha, 128\beta]$, where $\beta = 4$, $\alpha = 0.5, 1, 2$, denoted as RFPose-Tiny (RFP-T), RFPose-Base (RFP-B) and RFPose-Large (RFP-L), respectively. Each convolution layer is followed by a BN layer and ReLU activation function.

**Cross spatial attention (CSA) module** is an information aggregation module that makes vertical and horizontal RF signals interact with each other. The CSA is calculated as

$$\begin{aligned} CSA(i,j) &= \left(V(i) \cdot H^T(j)\right)/\sqrt{D}, \quad i,j \in [0, w \cdot h], \\ Z_{[1,w\cdot h, w\cdot h]} &= \textbf{Conv}\left(CSA_{[\beta, w\cdot h, w\cdot h]}\right), \end{aligned} \tag{8}$$

where $V, H$ (with a reshape operation $[c, w, h] \rightarrow [\beta, c/\beta, w \cdot h]$) are feature maps of vertical and horizontal RF signals that are extracted by backbone, $c, h, w, D$ are channel, height, width of feature maps and scale factor, and **Conv** is a $5 \times 5$ convolution layer with strides 2 and padding 2 to process CSA information.

**Pose estimation network (PEN):** The PEN network receives the representations from CSA to estimate 3D keypoints, which is composed of 2 FC layers with hidden size 256. Each FC layer is followed by a BN layer and ReLU activation function. The output layer size is $14 \times 3$ without activation function followed. The loss function is

$$L = \frac{1}{N}\sum_{i=1}^{N} \|x_i - y_i\|_2 + \left\| \frac{1}{N}\sum_{i=1}^{N} (x_i - y_i) \right\|_2, \tag{9}$$

where $N$ indicates the number of human keypoints, $x_i$ and $y_i$ are prediction and ground-truth of 3D coordinates for $i$-th keypoint. It is noted that the proposed PEN model works only for single-user case since only single-user dataset has been accessible. A multi-user case could be supported by adding some additional modules like region proposal network (RPN) and ROI Pooling as [53, 54].

*5.2.2 Training Details.* **(1) Pre-training:** We perform pre-training on the pre-training dataset with a total training number of 50 epochs. **(2) Fine-tune:** The PEN module combined with a frozen or fine-tuned backbone is trained on the training dataset with 50 epochs and evaluated on the validation dataset. The batch size is 128. The learning rate is 0.03 for PEN and 0.003 for fine-tuning backbone.

## 5.3 Experimental Results

*5.3.1 3D Pose Estimation Performance.*

**Comparison study:** The performance of 3D pose estimation with different methods are shown in Table. 7 and Figure. 6. We can see that the RF-URL pre-training method achieves centimeter accuracy for RFP-T (63*mm*), RFP-B (64*mm*) and RFP-L (64*mm*), while the trained model from scratch achieves decimeter accuracy for RFP-T (102*mm*), RFP-B (114*mm*) and RFP-L (262*mm*). These results demonstrate that 3D pose estimation task is benefit from the RF-URL pre-training model. From Table. 7, we can also see that with RFP-T, an elaborated designed RF-URL can improve the performance by 39*mm*. The result shows that our RF-URL pre-training model can generalize well to 3D pose estimation task.

The performance of RFP-T trained from scratch (102*mm*) is higher than that of RF-Pose3D (112.7*mm*) although the network architecture of RFP-T is simple. This is due to the fact that the

training dataset of RFP3D is relatively small which may not be adequate to tune the parameters in RF-Pose3D. A similar phenomenon occurs for RFP-B (114*mm*) and RFP-L (262*mm*), which have more parameters but achieve poorer performance.

**Table 7: Evaluation of different models on 3D pose estimation.**

| Pre-training | Method | Pose Err.(mm) |
|---|---|---|
| - | RF-Pose3D[54] | 112.7 |
| - | RFP-T | **102** |
| | RFP-B | 114 |
| | RFP-L | 262 |
| RF-URL (Fine-tune) | RFP-T | **63** (-39) |
| | RFP-B | 64 (-50) |
| | RFP-L | 64 (-198) |
| RF-URL (Details) | Baseline: RFP-T(w/o IAM) | 97 |
| | + RF-URL(frozen) | 79 (-18) |
| | + CSA | 70 (-9) |
| | + Predictor | 68 (-2) |
| | + Fine-tune | 63 (-5) |
| | + Shuffle BN | 62 (-1) |

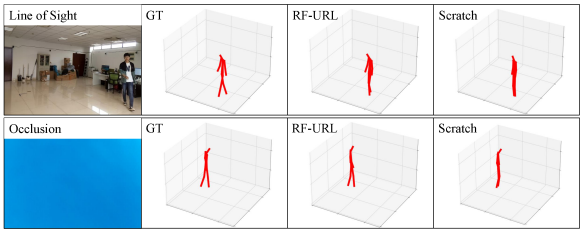


**Figure 6: Pose estimation under different environments, where RF-URL adopts fine-tuning strategy and GT stands for ground-truth.**

**Representation quality:** Table. 8 reveals that pre-training using RF-URL learns high-quality representations, which delivers a maximum improvement of 135*mm* for RFP-B and a minimum improvement of 123*mm* for RFP-L compared with random initialization backbone.

**Table 8: Evaluate the performance (Pose Err.(mm)) of RF-URL for 3D pose estimation by RFP under frozen feature setting with different backbone initialization strategy.**

| Model | Parameters | Random init | RF-URL (Frozen) |
|---|---|---|---|
| RFP-T | 2.66M | **198** | **68** |
| RFP-B | 3.87M | 206 | 71 |
| RFP-L | 7.09M | 200 | 77 |

**Does RFP-B/L exist over-fitting?** Table. 7 shows that a larger model gets a worse accuracy, which might exist over-fitting due to the limited dataset. Therefore, an early terminated training experiment for RFP-B is conducted to verify our conjecture with RF-URL (Frozen), and the results are shown in Table. 9. When only training 30 epochs, RFP-B obtains the highest accuracy 69*mm* that is almost the same accuracy as RFP-T. This result suggests that pre-training dataset should be as rich as possible and an early terminated training can relieve the over-fitting problem.

**Table 9: Early terminated training of RFP-B**

| Epoch | 10 | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|
| Pose Err.(mm) | 89 | 73 | **69** | 71 | 71 |

*5.3.2 Different Annotated Samples.* Table. 10 shows the performance of RFP-T/B/L with different annotated samples. The training dataset are randomly sampled *n*% of samples, e.g., 100%labels (25842 samples), 50% (12921samples) and 10% (2584 samples). All results are evaluated on RFP3D validation dataset. From Table. 10, we can see that all RF-URL pre-training methods outperform the corresponding scratch methods. Even only with 10% of training data, the RF-URL pre-training methods still maintain relatively high accuracy for RFP-T (103*mm*), RFP-B (109*mm*) and RFP-L (119*mm*). Fine-tune RF-URL method is higher than frozen method about 4.1 mm. These results show that the RF-URL pre-training models are scalable to the limited data condition.

**Table 10: Evaluate the performance (Pose Err.(mm)) of RFP with different annotated samples, Frozen and Fine-tune are backbone initialized from RF-URL.**

| Model | Pre-training | 100% labels | 50% labels | 10% labels |
|---|---|---|---|---|
| RFP-T | - | 102 | 304 | 305 |
| | Frozen | 68 | 72 | 104 |
| | Fine-tune | **63** | **68** | **103** |
| RFP-B | - | 114 | 288 | 305 |
| | Frozen | 71 | 76 | 111 |
| | Fine-tune | **64** | **70** | **109** |
| RFP-L | - | 262 | 303 | 305 |
| | Frozen | 77 | 82 | **119** |
| | Fine-tune | **64** | **71** | 122 |

*5.3.3 Different Size of Pre-training Dataset.* We evaluate the performance of RFP-T under different size of pre-training dataset. Table. 11 shows that as the size of pre-training dataset reduces from 100% to 0%, the performance decreases from 63*mm* to 86*mm*. The decreased performance is expected since the pre-training model gradually suffers from the over-fitting problem with the size reduction of pre-training dataset.

**Table 11: Evaluate the performance (Pose Err.(mm)) of RFP-T under different size of pre-training dataset.**

| Size | 100% | 80% | 60% | 40% | 20% | 0% |
|---|---|---|---|---|---|---|
| Frozen | 68 | 79 | 88 | 101 | 109 | 198 |
| Fine-tune | 63 | 67 | 72 | 78 | 83 | 86 |

*5.3.4 Ablation.* In this subsection, we conduct ablation studies to evaluate some important components of our RF-URL framework for 3D pose estimation.

**Predictor:** We ablate the predictor of RF-URL by using RFP-T (Frozen) with different training duration. The results are shown in Table. 12, from which we can see that the highest accuracy 68*mm* is obtained when predictor is only trained 5 epochs. However, a longer trained predictor (epoch 50) reduces the performance by 4*mm* compared that without predictor. The results indicate that RF-URL could be benefit from a short trained predictor.

**Table 12: Predictor with different training epochs.**

| Epochs | 0 (w/o predictor) | 5 | 15 | 30 | 50 |
|---|---|---|---|---|---|
| Pose Err.(mm) | 70 | **68** | 69 | 69 | 74 |

**Information aggregation module (IAM):** We ablate IAM for RF-URL 3D pose estimation using RFP-T network. The candidate modules include CSA, channel shuffle (CS) [33], layer-3 (feature extracted from layer 3 of translator) and without IAM (just stack the extracted feature maps). The results are illustrated in Table. 13. CSA obtains the highest accuracy of 68*mm*. Compared without IAM, CSA, CS and layer-3 improves the accuracy by 11*mm*, 4*mm* and 8*mm*. Note that CSA reduces the error by 5*mm* when training from scratch, according to Table. 7 where RFP-T with CSA gets 102*mm* accuracy and RFP-T without IAM gets 97*mm* accuracy. These results demonstrate that CSA is more suitable for RF-URL pre-training model, but has a negative impact for training from scratch.

**Table 13: Different information aggregation module.**

| IAM | CSA | CS | w/o IAM | layer-3 |
|---|---|---|---|---|
| Pose Err.(mm) | **68** | 75 | 79 | 71 |

**Shuffle BN vs Sync BN:** As reported in Section. 4.3.4, BN might leak information that prevents RF-URL from learning good representations. Thus, we ablate Shuffle BN and Sync BN for RF-URL in RFP-T network. The results are illustrated in Table. 7. We can see that a shuffle BN improves 1*mm* for fine-tuned backbone. Although only a little performance improvement, Sync BN also has a positive impact on RF-URL for 3D pose estimation.

# 6 HUMAN SILHOUETTE GENERATION

## 6.1 Dataset

Using the same hardware system as in Section 5, we collect a multi-modal dataset to evaluate the RF-URL framework for human silhouette generation task. The multi-camera system captures images to generate human silhouette ground-truth using Mask R-CNN [24]. We collect data in four different environments under 11 different conditions, including random walk with no occlusion, random walk under occlusion (styrofoam, carton, yoga mat and dark) and random action (stand, walk, squat and sit) for both single-person and multi-person scenarios. We use three environments data as pre-training dataset (119280 samples), and one environments data excluding occlusion parts as training dataset (16272 samples) and validate dataset (3312 samples). We feed 12 frames of RF signals to generate 6 frames of human silhouette segmentation.

## 6.2 Baseline

*6.2.1 Network Structure.* For fair comparison with the existing methods, we do not adopt the same backbone in Section 5. Instead, our human silhouette generation network, named as RFSG, follows the design of RF-Pose [52], which uses two RF encoding networks to extract features from vertical and horizontal RF signals. Then, the outputs of encoding networks are concatenated and fed into generation network to generate human silhouette segmentation.

**Backbone network:** The RF encoding network uses 10 $9\times5\times5$ 3D convolutions layers with $1\times2\times2$ strides and $16\alpha$ channels followed by a BN layer and a ReLU activation function, where $\alpha = 1$ for the last layer. The channels of backbone network $\alpha = 0.5, 1, 2$ are named as RFSG-T, RFSG-B and RFSG-L.

**Silhouette generation network (SGN):** The generation network is composed of 4 deconvolution layers, where the first three layers are equipped with the kernel of size $3\times6\times6$ and stride $1\times2\times2$, while the last one has the kernel of size $3\times6\times6$ and stride $1\times4\times4$. The number of channels at different layers are [64, 32, 16, 1], respectively. RF-Pose [52] uses Parametric ReLU (PReLU) activation function without BN layer. However, RFSG uses Leaky ReLU with slope of 0.02 and BN layer.

*6.2.2 Training Details.* **(1) Pre-training:** We perform pre-training on pre-training dataset with a total training number of 50 epochs. **(2)Fine-tune:** The SGN module combined with a frozen or fine-tuned backbone is trained on training dataset with 50 epochs and evaluated on validation dataset. The batch size is 64. The learning rate of SGN and backbone is 1.0.

## 6.3 Experimental Results

*6.3.1 Human Silhouette Generation Performance.*

**Comparison study:** The performance of human silhouette generation with different methods are shown in Table. 14 and Figure. 7. The RF-URL pre-training method achieves IoU of 0.610, 0.619, 0.613 for RFSG-T, RFSG-B and RFSG-L, while the trained model from scratch only achieves IoU of 0.539, 0.556, 0.571 for RFSG-T, RFSG-B and RFSG-L, respectively. Compared to RF-Pose [52], with a similar number of parameters, RFSG-B improves the performance of IoU about 0.036. The results show that our RF-URL pre-training models can generalize well to human silhouette generation.

**Representation quality:** Table. 15 reveals that pre-training using RF-URL learns high quality representations, which delivers an IoU improvement of 0.327, 0.317, 0.288 for RFSG-T, RFSG-B and RFSG-L, respectively, compared with random initialization backbone.

*6.3.2 Different Annotated Samples.* Table. 16 shows the performance of RFSG-T/B/L with different annotated samples. The training dataset are randomly sampled *n*% of samples, e.g., 100%labels (16272 samples), 50% (8136 samples) and 10% (1627 samples). All results are evaluated on the validation dataset. From Table. 16, we can see that all RF-URL pre-training methods outperform the corresponding scratch methods. Even only with 10% of training data, the RF-URL pre-training methods still maintain relatively high IoU for RFSG-T (0.581), RFSG-B (0.586) and RFSG-L (0.565). However, the performance without pre-training decreases rapidly, i.e., decrease

**Table 14: The performance of different models on the human silhouette generation.**

| Pre-training | Method | IoU |
|---|---|---|
| - | RF-Pose[52] | **0.583** |
| - | RFSG-T | 0.539 |
| | RFSG-B | 0.556 |
| | RFSG-L | **0.571** |
| RF-URL (Fine-tune) | RFSG-T | 0.610 (+0.071) |
| | RFSG-B | **0.619** (+0.063) |
| | RFSG-L | 0.613 (+0.042) |
| RF-URL (Details) | RFSG-B (baseline) | 0.556 |
| | + RF-URL(frozen) | 0.557 (+0.001) |
| | + Fine-tune | 0.611 (+0.054) |
| | + Predictor | 0.619 (+0.008) |
| | + Shuffle BN | 0.614 (-0.005) |

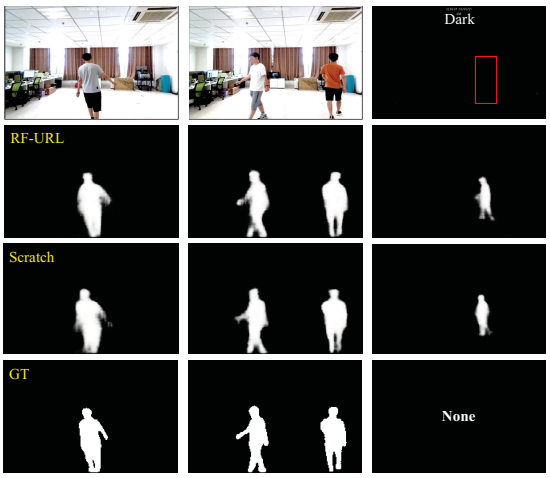


**Figure 7: Human silhouette generation under different environments, where RF-URL adopts fine-tuning strategy and GT stands for ground-truth.**

**Table 15: Evaluate the performance (IoU) of RF-URL for human silhouette generation under frozen feature setting with different backbone initialization strategy.**

| Model | Parameters | Random init | RF-URL (Frozen) |
|---|---|---|---|
| RFSG-T | 0.39M | 0.225 | 0.552 |
| RFSG-B | 0.76M | 0.239 | 0.556 |
| RFSG-L | 2.09M | 0.248 | 0.536 |

IoU of 0.082, 0.075 and 0.069. These results show that a small annotated dataset could be benefit from RF-URL pre-training method.

*6.3.3 Different Size of Pre-training Dataset.* Table. 17 shows the performance of RFSG-B with different size of pre-training dataset. It illustrates that RFSG-B works well in a larger pre-training dataset for both forzen and fine-tune strategy.

**Table 16: Evaluate the performance (IoU) of RFSG with different annotated samples, Frozen and Fine-tune are backbone initialization from RF-URL.**

| Model | Pre-training | 100% labels | 50% labels | 10% labels |
|---|---|---|---|---|
| RFSG-T | - | 0.539 | 0.539 | 0.457 |
| | Frozen | 0.552 | 0.553 | 0.532 |
| | Fine-tune | **0.610** | **0.611** | **0.581** |
| RFSG-B | - | 0.556 | 0.550 | 0.481 |
| | Frozen | 0.557 | 0.552 | 0.537 |
| | Fine-tune | **0.619** | **0.614** | **0.586** |
| RFSG-L | - | 0.571 | 0.591 | 0.502 |
| | Frozen | 0.536 | 0.529 | 0.506 |
| | Fine-tune | **0.613** | **0.612** | **0.565** |

**Table 17: Evaluate the performance (IoU) of RFSG-B under different size of pre-training dataset.**

| Size | 100% | 80% | 60% | 40% | 20% | 0% |
|---|---|---|---|---|---|---|
| Frozen | 0.557 | 0.531 | 0.529 | 0.489 | 0.426 | 0.239 |
| Fine-tune | 0.619 | 0.602 | 0.585 | 0.573 | 0.562 | 0.556 |

*6.3.4 Ablation.* In this subsection, we conduct ablation studies to evaluate some important components of our RF-URL framework for human silhouette generation.

**Predictor:** We ablate the predictor of RF-URL by using RFSG-B (Fine-tune) with different training duration. The results are shown in Table.18, from which we can see that the highest IoU 0.619 is obtained when predictor is only trained 5 epochs. The results indicate that RF-URL could be benefit from a short trained predictor.

**Table 18: Predictor with different training epochs.**

| Epochs | 0 (w/o predictor) | 5 | 15 | 30 | 50 |
|---|---|---|---|---|---|
| IoU | 0.611 | **0.619** | 0.615 | 0.617 | 0.615 |

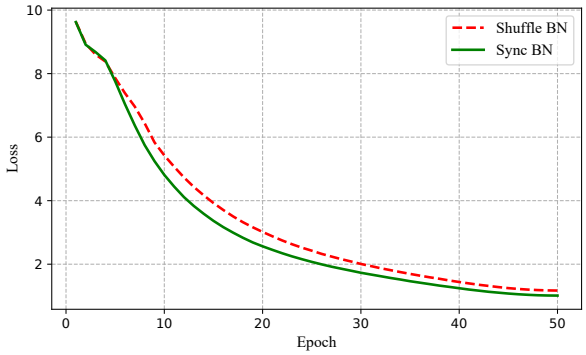


**Figure 8: Training RF-URL with Shuffle BN or Sync BN.**

**Shuffle BN vs Sync BN:** We ablate Shuffle BN and Sync BN for RF-URL based on RFSG-B. As shown in Table. 14, shuffle BN

decreases the performance of RF-URL pre-training model, which is different from those in Section 4.3.4 and 5.3.4. This may be reason that 3D full convolution network RFSG is hard to train. As shown in Figure. 8, compared with the negative impact of leaking information, the poor convergence brought by Shuffle BN is more harmful.

## 7 RELATED WORK

**RF sensing:** Learning-based RF sensing has recently gained attentions in health care and smart homes, including human gesture recognition [26, 31, 41, 42, 51], activity recognition [11, 15, 29, 43, 45, 46], human pose estimation [30, 52–54], person re-identification [16, 32], fall detection [39], vital sign monitoring [12, 13, 48–50, 55], and so on. These existing works mainly rely on supervised learning which requires large-scale annotated RF datasets, while the proposed framework exploits unannotated data for model training.

**Masking and predicting model:** The pre-training method has achieved unprecedented success in NLP community, e.g. GPT [4, 35, 36] and BERT [14]. These methods mask a portion of the input sequence and try to predict the missing content, which have been shown with excellent scalability and generalization. Although language and RF signals have similar sequential structure, relevant information in radio signals are typically very sparse while language signals are highly information-dense, causing a big gap between RF sensing and NLP community.

**Contrastive learning:** Recently, contrastive learning has became popular for learning effective representations. The learned representations make downstream tasks solved easier, and the performance even surpasses the supervised methods [8, 23]. The core idea of contrastive learning is to attract the positive sample pairs and repulse the negative sample pairs. The commonly used contrastive learning frameworks include memory bank method (e.g., InsDis [44], MoCo [8, 23] and contrastive multiview coding (CMC) [38]), big batchsize (e.g., SimCLR [7]), clustering (e.g., SwAV [5]), transformer (e.g., MoCov3 [10] and DINO [6]) and negative-pairs-free methods (e.g., BYOL [20] and SimSiam [9]). However, these methods strongly depend on data augmentation [7, 9, 20], which always tends to learn shortcut rather than meaningful information for RF signals [32].

**Contrastive multiview coding (CMC) [38]:** Our RF-URL is a form of CMC, but different from the classical CMC in following ways. Firstly, existing investigations have demonstrated that traditional data augmentation methods are inefficient for RF data. To resolve this problem, we have noted that different signal processing methods could naturally generate different representations of the same signal. Inspired by this phenomenon, RF-URL utilizes RF signal processing methods to construct positive and negative pairs rather than data augmentation. Secondly, the inputs of CMC are symmetric but RF-URL adopts an asymmetric signal representations as input(e.g., DFS and AoA-ToF). The asymmetric signal representations make the CMC-based method not converge, while RF-URL adopts a translator to solve this problem. In addition, RF-URL adopts a predictor with stop-grad operation to improve the quality of learned representations through a short-term training.

**Synthetic data:** Synthetic dataset can be generated by a RF ray-tracing simulator to solve the data-hungry problem [21, 37]. Although synthetic RF signals share some similar properties with real RF data, the simulators may not capture all physical RF phenomena such as multi-path, reflections, diffraction and polarization effects causing a gap between synthetic and real RF signals. Nevertheless, these techniques can also be integrated with the proposed framework to generate more data for general semantic feature extraction.

## 8 DISCUSSIONS

**Why contrastive learning is a reasonable choice?** Different from visual images or natural languages, annotating RF signals is much more costly since they are non-intuitive and non-interpretable. Contrastive learning is an unsupervised learning method that could learn a general semantic representation from unannotated dataset. The main challenge of contrastive learning lies in the design of the principle to construct positive and negative pairs. To this end, we have noted that different signal processing methods could naturally generate different representations of the same signal, which could be utilized to construct positive/negative pairs. Since various signal processing methods have been developed for RF sensing, we could utilize contrastive learning to seamlessly integrate these well-developed signal processing techniques. In this sense, contrastive learning is a good choice.

**Signal representation:** In this paper, the backbone of RF-URL is based on convolutional neural networks (CNNs), which could extract features of the signals based on their spatial distributions. Although linear transformations are utilized to generate different signal representations which wouldn't change the information of signal itself, they could rearrange the spatial characteristics of the input signal to yield different spatial information. Thus, different linear transformations of the same RF signal could expose different spatial features to CNNs, which is helpful for training.

**The role of predictor and InfoNCE:** The functionality of InfoNCE is to attract positive pairs and repulse negative pairs, where both the positive and negative pairs are composed by the features from memory bank and the output of neural network. However, since the memory bank is randomly initialized, at the early stages of the training processes, random noise may be sampled to form positive pairs with the network output, leading to fluctuations of training. On the other hand, predictor directly shortens the distance between the output of multi-branch neural network, which can weaken the negative impact of noise in memory bank and smooth the training processes. Thus, in this paper, we enable the predictor in the first 5 epochs of the training processes, which lead to slightly performance gain. Note that the further training of predictor tends to prevent the backbone from aligning the features of the memory bank, which may lead to performance degradation.

## 9 CONCLUSION

This paper presented a novel URL framework, RF-URL, for RF-based sensing through contrastive learning. The positive and negative pairs were constructed with different signal processing technologies, which achieved effective contrastive learning in an unsupervised manner by minding the gap between signal processing and neural network. All experimental results strongly demonstrated that RF-URL took an important step towards deploying learning-based solutions for large-scale RF-based sensing applications.

# REFERENCES


[1] Fadel Adib, Zachary Kabelac, Dina Katabi, and Robert C. Miller. 2014. 3D Tracking via Body Radio Reflections. In *Proc. of the 11th USENIX (NSDI'14)*. 317–329.

[2] Fadel Adib, Hongzi Mao, Zachary Kabelac, Dina Katabi, and Robert C. Miller. 2015. Smart Homes That Monitor Breathing and Heart Rate. In *Proc. of the 33rd ACM CHI (CHI '15)*. 837–846.

[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 8 (2013), 1798–1828.

[4] Tom Brown, Benjamin Mann, Nick Ryder, and et. al. 2020. Language Models are Few-Shot Learners. In *Proc. of the 34th NeurIPS (NIPS'20, Vol. 33)*. 1877–1901.

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *Proc. of the 34th NeurIPS (NIPS'20)*. Article 831, 13 pages.

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging Properties in Self-Supervised Vision Transformers. In *Proc. of the IEEE/CVF ICCV*. 9630–9640.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proc. of the 37th ICML (ICML'20)*. Article 149, 11 pages.

[8] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. 2020. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297* (2020).

[9] Xinlei Chen and Kaiming He. 2021. Exploring Simple Siamese Representation Learning. In *Proc. of the IEEE/CVF CVPR*. 15745–15753.

[10] Xinlei Chen, Saining Xie, and Kaiming He. 2021. An Empirical Study of Training Self-Supervised Vision Transformers. In *Proc. of the IEEE/CVF ICCV*. 9620–9629.

[11] Yan Chen, Hongyu Deng, Dongheng Zhang, and Yang Hu. 2021. SpeedNet: Indoor Speed Estimation With Radio Signals. *IEEE Internet of Things Journal* 8, 4 (2021), 2762–2774.

[12] Yan Chen, Xiang Su, Yang Hu, and Bing Zeng. 2020. Residual Carrier Frequency Offset Estimation and Compensation for Commodity WiFi. *IEEE Transactions on Mobile Computing* 19, 12 (2020), 2891–2902.

[13] Zhe Chen, Tianyue Zheng, Chao Cai, and Jun Luo. 2021. MoVi-Fi: Motion-Robust Vital Signs Waveform Recovery via Deep Interpreted RF Sensing. In *Proc. of the 27th ACM MobiCom (MobiCom '21)*. 392–405.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of the ACL*. 4171–4186.

[15] Shuya Ding, Zhe Chen, Tianyue Zheng, and Jun Luo. 2020. RF-Net: A Unified Meta-Learning Framework for RF-Enabled One-Shot Human Activity Recognition. In *Proc. of the 18th ACM SenSys*. 517–530.

[16] Lijie Fan, Tianhong Li, Rongyao Fang, Rumen Hristov, Yuan Yuan, and Dina Katabi. 2020. Learning Longterm Representations for Person Re-Identification Using Radio Signals. In *Proc. of the IEEE/CVF CVPR*. 10696–10706.

[17] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. 2017. RMPE: Regional Multi-person Pose Estimation. In *Proc. of the IEEE/CVF ICCV*. 2353–2362.

[18] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. SlowFast Networks for Video Recognition. In *Proc. of the IEEE/CVF ICCV*. 6201–6210.

[19] Anirudh Goyal and Yoshua Bengio. 2020. Inductive biases for deep learning of higher-level cognition. *arXiv preprint arXiv:2011.15091* (2020).

[20] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. 2020. Bootstrap Your Own Latent a New Approach to Self-Supervised Learning. In *Proc. of the 34th NeurIPS (NIPS'20)*. Article 1786, 14 pages.

[21] Junfeng Guan, Sohrab Madani, Suraj Jog, Saurabh Gupta, and Haitham Hassanieh. 2020. Through Fog High-Resolution Imaging Using Millimeter Wave Radar. In *Proceedings of the IEEE/CVF CVPR*. 11461–11470.

[22] R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *Proc. of the IEEE/CVF CVPR )*, Vol. 2. 1735–1742.

[23] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proc. of the IEEE/CVF CVPR*. 9726–9735.

[24] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask R-CNN. In *Proc. of the IEEE/CVF ICCV*. 2980–2988.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proc. of the IEEE/CVF CVPR*. 770–778.

[26] Ying He, Yan Chen, Yang Hu, and Bing Zeng. 2020. WiFi Vision: Sensing, Recognition, and Detection With Commodity MIMO-OFDM WiFi. *IEEE Internet of Things Journal* 7, 9 (2020), 8296–8317.

[27] Chen-Yu Hsu, Rumen Hristov, Guang-He Lee, Mingmin Zhao, and Dina Katabi. 2019. Enabling Identification and Behavioral Sensing in Homes Using Radio Reflections. In *Proc. of the 2019 ACM CHI (CHI '19)*. 1–13.

[28] Wenjun Jiang, Chenglin Miao, Fenglong Ma, Shuochao Yao, Yaqing Wang, Ye Yuan, Hongfei Xue, Chen Song, Xin Ma, Dimitrios Koutsonikolas, Wenyao Xu, and Lu Su. 2018. Towards Environment Independent Device Free Human Activity Recognition. In *Proc. of the 24th ACM MobiCom (MobiCom '18)*. 289–304.

[29] Wenjun Jiang, Chenglin Miao, Fenglong Ma, Shuochao Yao, Yaqing Wang, Ye Yuan, Hongfei Xue, Chen Song, Xin Ma, Dimitrios Koutsonikolas, Wenyao Xu, and Lu Su. 2018. Towards Environment Independent Device Free Human Activity Recognition. In *Proc. of the 24th ACM MobiCom (MobiCom '18)*. 289–304.

[30] Wenjun Jiang, Hongfei Xue, Chenglin Miao, Shiyang Wang, Sen Lin, Chong Tian, Srinivasan Murali, Haochen Hu, Zhi Sun, and Lu Su. 2020. Towards 3D Human Pose Construction Using Wifi. In *Proc. of the 26th ACM MobiCom*. Article 23, 14 pages.

[31] Hong Li, Wei Yang, Jianxin Wang, Yang Xu, and Liusheng Huang. 2016. WiFinger: Talk to Your Smart Devices with Finger-Grained Gesture. In *Proc. of the ACM UbiComp (UbiComp '16)*. 250–261.

[32] Tianhong Li, Lijie Fan, Yuan Yuan, and Dina Katabi. 2022. Unsupervised Learning for Human Sensing Using Radio Signals. In *Proc. of the IEEE/CVF WACV*. 1091–1100.

[33] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. 2018. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In *Proc. of the ECCV*. Cham, 122–138.

[34] Adam Paszke, Sam Gross, Francisco Massa, and et. al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Proc. of the 33rd NeurIPS*. Article 721, 12 pages.

[35] Alec Radford and Karthik Narasimhan. 2018. Improving Language Understanding by Generative Pre-Training.

[36] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* (2019).

[37] Hem Regmi, Moh Sabbir Saadat, Sanjib Sur, and Srihari Nelakuditi. 2021. SquiggleMilli: Approximating SAR Imaging on Mobile Millimeter-Wave Devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 3, Article 125 (sep 2021), 26 pages.

[38] Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2020. Contrastive Multiview Coding. In *Proc. of the 16th ECCV*. 776–794.

[39] Yonglong Tian, Guang-He Lee, Hao He, Chen-Yu Hsu, and Dina Katabi. 2018. RF-Based Fall Monitoring Using Convolutional Neural Networks. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 3, Article 137 (sep 2018), 24 pages.

[40] Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[41] Raghav H. Venkatnarayan, Shakir Mahmood, and Muhammad Shahzad. 2021. WiFi based Multi-User Gesture Recognition. *IEEE Transactions on Mobile Computing* 20, 3 (2021), 1242–1256.

[42] Aditya Virmani and Muhammad Shahzad. 2017. Position and Orientation Agnostic Gesture Recognition Using WiFi. In *Proc. of the 15th ACM MobiSys (MobiSys '17)*. 252–264.

[43] Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu. 2014. E-Eyes: Device-Free Location-Oriented Activity Identification Using Fine-Grained WiFi Signatures. In *Proc. of the 20th ACM MobiCom (MobiCom '14)*. 617–628.

[44] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. 2018. Unsupervised Feature Learning via Non-parametric Instance Discrimination. In *Proc. of the IEEE/CVF CVPR*. 3733–3742.

[45] Qinyi Xu, Yan Chen, BeiBei Wang, and K. J. Ray Liu. 2017. Radio Biometrics: Human Recognition Through a Wall. *IEEE Transactions on Information Forensics and Security* 12, 5 (2017), 1141–1155.

[46] Qinyi Xu, Yan Chen, Beibei Wang, and K. J. Ray Liu. 2017. TRIEDS: Wireless Events Detection Through the Wall. *IEEE Internet of Things Journal* 4, 3 (2017), 723–735.

[47] Cong Yu, Zhi Wu, Dongheng Zhang, Zhi Lu, Yang Hu, and Yan Chen. 2022. RFGAN: RF-Based Human Synthesis. *IEEE Transactions on Multimedia* (2022), 1–1.

[48] Dongheng Zhang, Yang Hu, and Yan Chen. 2021. MTrack: Tracking Multiperson Moving Trajectories and Vital Signs With Radio Signals. *IEEE Internet of Things Journal* 8, 5 (2021), 3904–3914.

[49] Dongheng Zhang, Yang Hu, Yan Chen, and Bing Zeng. 2019. BreathTrack: Tracking Indoor Human Breath Status via Commodity WiFi. *IEEE Internet of Things Journal* 6, 2 (2019), 3899–3911.

[50] Dongheng Zhang, Yang Hu, Yan Chen, and Bing Zeng. 2020. Calibrating Phase Offsets for Commodity WiFi. *IEEE Systems Journal* 14, 1 (2020), 661–664.

[51] Yi Zhang, Yue Zheng, Kun Qian, Guidong Zhang, Yunhao Liu, Chenshu Wu, and Zheng Yang. 2021. Widar3.0: Zero-Effort Cross-Domain Gesture Recognition with Wi-Fi. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021), 1–1.

[52] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. 2018. Through-Wall Human Pose Estimation Using Radio Signals. In *Proc. of the IEEE/CVF CVPR*. 7356–7365.

[53] Mingmin Zhao, Yingcheng Liu, Aniruddh Raghu, Hang Zhao, Tianhong Li, Antonio Torralba, and Dina Katabi. 2019. Through-Wall Human Mesh Recovery Using Radio Signals. In *Proc. of the IEEE/CVF ICCV*. 10112–10121.

[54] Mingmin Zhao, Yonglong Tian, Hang Zhao, Mohammad Abu Alsheikh, Tianhong Li, Rumen Hristov, Zachary Kabelac, Dina Katabi, and Antonio Torralba. 2018. RF-Based 3D Skeletons. In *Proc. of the ACM SIGCOMM (SIGCOMM '18)*. 267–281.

[55] Tianyue Zheng, Zhe Chen, Chao Cai, Jun Luo, and Xu Zhang. 2020. V2iFi: In-Vehicle Vital Sign Monitoring via Compact RF Sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 2, Article 70 (jun 2020), 27 pages.