DiffAR: Adaptive Conditional Diffusion Model for Temporal-augmented Human Activity Recognition

Shuokang Huang[1] , **Po-Yu Chen**[1,2] and **Julie McCann**[1]
[1]Imperial College London
[2]JPMorgan Chase & Co.
{s.huang21, po-yu.chen11, j.mccann}@imperial.ac.uk

Abstract

Human activity recognition (HAR) is a fundamental sensing and analysis technique that supports diverse applications, such as smart homes and healthcare. In device-free and non-intrusive HAR, WiFi channel state information (CSI) captures wireless signal variations caused by human interference without the need for video cameras or on-body sensors. However, current CSI-based HAR performance is hampered by incomplete CSI recordings due to fixed window sizes in CSI collection and human/machine errors that incur missing values in CSI. To address these issues, we propose DiffAR, a temporal-augmented HAR approach that improves HAR performance by augmenting CSI. DiffAR devises a novel Adaptive Conditional Diffusion Model (ACDM) to synthesize augmented CSI, which tackles the issue of fixed windows by forecasting and handles missing values with imputation. Compared to existing diffusion models, ACDM improves the synthesis quality by guiding progressive synthesis with step-specific conditions. DiffAR further exploits an ensemble classifier for activity recognition using both raw and augmented CSI. Extensive experiments on four public datasets show that DiffAR achieves the best synthesis quality of augmented CSI and outperforms state-of-the-art CSI-based HAR methods in terms of recognition performance. The source code of DiffAR is available at https://github.com/huangshk/DiffAR.

1 Introduction

Human activity recognition (HAR) supports a significant number of important yet differing applications in the fields of security [Lin *et al.*, 2020], smart homes [Bianchi *et al.*, 2019], healthcare [An and Ogras, 2021], *etc*. It aims to classify human actions using signals from various sources (*e.g*, cameras, wearable sensors, and radars). However, these traditional approaches have several drawbacks. People may object to being constantly videoed or wearing on-body sensors, so these devices will fail to gather signals [Yang *et al.*, 2018]. Cameras require adequate illumination and line-of-sight (LOS) conditions to capture acceptable frames to analyze [Hussain *et al.*, 2020]. Radar sensing approaches may solve these issues, replacing vision with radio frequency (RF), but rely on costly dedicated devices and require particular deployment methodologies [Nirmal *et al.*, 2021].

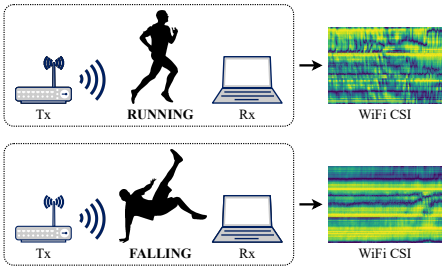

Figure 1: Different human activities interfere with wireless signals between transmitters (Tx) and receivers (Rx), manifesting distinct WiFi CSI patterns which contain implicit human features for HAR.

To overcome these drawbacks, the use of WiFi channel state information (CSI) has emerged [Yousefi *et al.*, 2017]. CSI records the state of the wireless signals that experience interference, where human movement is one such interference [Wang *et al.*, 2015]. Different human activities lead to distinct WiFi CSI patterns, as shown in Figure 1. Hence, recent studies [Tan *et al.*, 2022] have exploited CSI for non-intrusive HAR, because CSI does not require cameras or sensors, nor are they restricted by illumination or LoS constraints. More importantly, ubiquitous off-the-shelf WiFi devices can provide vast amounts of CSI data, enabling device-free HAR without the need for dedicated devices.

Since WiFi was initially designed for communication, not sensing, the implicit human features in CSI are not easy to extract, so further schemes are required to interpret CSI patterns for HAR. Much effort has been devoted to learn implicit features using deep learning (DL). Initial research applied Long Short Term Memory (LSTM) to extract temporal features from CSI [Yousefi *et al.*, 2017]. Some studies [Wang *et al.*, 2019; Moshiri *et al.*, 2021] used Convolutional Neural Networks (CNNs) to learn spatial features from CSI. Recently, significant progress has been made by attention-based models [Vaswani *et al.*, 2017], such as attention-based bi-directional LSTM (ABLSTM) [Chen *et al.*, 2018] and two-stream convolution augmented transformers (THAT) [Li *et al.*, 2021].

Regardless of this, in practice, the constraints of off-the-

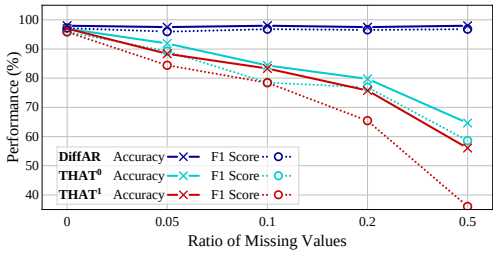

Figure 2: Comparison of **DiffAR** and **THAT** under different ratios of missing values in CSI. **THAT**^{0} is tuned by samples with missing values, while **THAT**^{1} is not tuned by samples with missing values.

shelf WiFi devices usually lead to incomplete CSI samples, limiting the maximal attainable performance of HAR. Most devices apply fixed windows to process CSI, such as the 2-second windows in [Yousefi *et al.*, 2017]. These fixed windows cannot match the durations of different activities, producing gaps where CSI is not recorded and impacting HAR performance. Similarly, device failures and/or human errors [Tashiro *et al.*, 2021] incur missing values in CSI, hindering models to extract distinctive features. For example, the performance of THAT [Li *et al.*, 2021] on the dataset Office [Yousefi *et al.*, 2017] significantly decreases with increasing numbers of missing values, as shown in Figure 2. These deep-rooted issues seriously hamper CSI-based HAR performance.

In this paper, we propose a temporal-augmented HAR approach, DiffAR, to improve the recognition performance by augmenting incomplete CSI. DiffAR devises a novel Adaptive Conditional Diffusion Model (ACDM) to synthesize augmented CSI, which tackles fixed windows by forecasting and handles missing values with imputation. Existing diffusion models [Ho *et al.*, 2020] synthesize samples through progressive steps guided by constant conditions [Tashiro *et al.*, 2021], but different steps may actually require step-specific conditions to synthesize patterns of different granularity. Intuitively, when synthesizing CSI guided by its spectrogram, low-frequency features can contribute to earlier steps to synthesize global patterns, while high-frequency features can assist in later steps to synthesize local patterns. Hence, ACDM employs an adaptive conditioner which learns step-specific conditions to guide each progressive step. Ultimately, an ensemble classifier uses both raw CSI and augmented CSI for activity recognition. Our main contributions are as follows:

- We propose a novel temporal-augmented HAR approach, DiffAR, to strengthen CSI-based HAR using diffusion models. To the best of our knowledge, this is the first attempt to augment WiFi CSI with diffusion models and to thereby improve the performance of CSI-based HAR.
- In ACDM, we present an adaptive conditioner which guides the progressive steps with step-specific conditions to synthesize patterns of different granularity. This proves the feasibility of step-specific conditions which improve the synthesis quality of diffusion models.
- Extensive experiments on four public datasets show that DiffAR realizes the best quality of augmented CSI. With augmented CSI, DiffAR also outperforms state-of-the-art CSI-based HAR methods in recognition performance.

2 Related Work

CSI-based HAR. Recent years have witnessed the increasing popularity of WiFi-based human sensing [Tan *et al.*, 2022], where WiFi CSI is the main signal source [Wang *et al.*, 2015; Ma *et al.*, 2019]. Traditional methods extracted human features from CSI using handcrafted solutions, such as short-time Fourier transform (STFT) [Yousefi *et al.*, 2017]. For example, the STFT-based random forest (ST-RF) approach was one of the best traditional models [Li *et al.*, 2021], but handcrafted solutions require expert knowledge and find it difficult to extract implicit features from complex data. With the rise of deep learning (DL), many studies have explored DL models for CSI-based HAR [Nirmal *et al.*, 2021]. Compared with ST-RF, LSTM showed better performance since it extracted implicit temporal features [Yousefi *et al.*, 2017]. Focusing on local temporal features, one-dimensional CNN (CNN-1D) [Wang *et al.*, 2019] was proposed and further improved recognition accuracy. Regarding CSI mapped as images, two-dimensional CNN (CNN-2D) [Moshiri *et al.*, 2021] was introduced to learn spatial features from CSI, resulting in further improved recognition performance. When CNN and LSTM were combined to learn both temporal and spatial features from CSI [Shalaby *et al.*, 2022], the performance was just slightly improved. Motivated by the success of attention mechanism [Vaswani *et al.*, 2017], ABLSTM [Chen *et al.*, 2018] applied an attention-based bi-directional LSTM to learn weighted temporal features and significantly increased recognition performance. Recently, THAT [Li *et al.*, 2021] has established a two-stream transformer to learn both temporal and channel features using multi-scale convolutions, achieving state-of-the-art performance in CSI-based HAR. However, the above studies relied on complete CSI and neglected the practical issues of incomplete CSI.

Generative Time-series Models. In real-world applications, time-series data are omnipresent and generative time-series models have attracted much attention from researchers [Wen *et al.*, 2021]. For time-series synthesis, generative adversarial networks (GANs) [Goodfellow *et al.*, 2020] have been widely used [Mogren, 2016; Esteban *et al.*, 2017]. For example, TimeGAN [Yoon *et al.*, 2019] regulated GANs with autoregressive models to obtain satisfactory synthesis quality. For better synthesis quality, recent studies have exploited diffusion models [Ho *et al.*, 2020; Yang *et al.*, 2022], which have achieved state-of-the-art performance in image generation [Rombach *et al.*, 2022], waveform synthesis [Kong *et al.*, 2021], *etc*. For time-series forecasting, TimeGrad [Rasul *et al.*, 2021] combined diffusion models with an RNN, whose hidden states were used as conditions to guide the synthesis in diffusion models. For time-series imputation, CSDI [Tashiro *et al.*, 2021] integrated diffusion models with a Transformer encoder [Vaswani *et al.*, 2017] to impute missing values in time series, showing competitive imputation quality. DiffWave [Kong *et al.*, 2021] developed a non-autoregressive diffusion model to synthesize waveforms conditioned on mel-spectrogram, achieving the best synthesis quality. Though none of these studies have investigated CSI augmentation, they proved the potential of diffusion models in coping with incomplete CSI by forecasting and imputation.

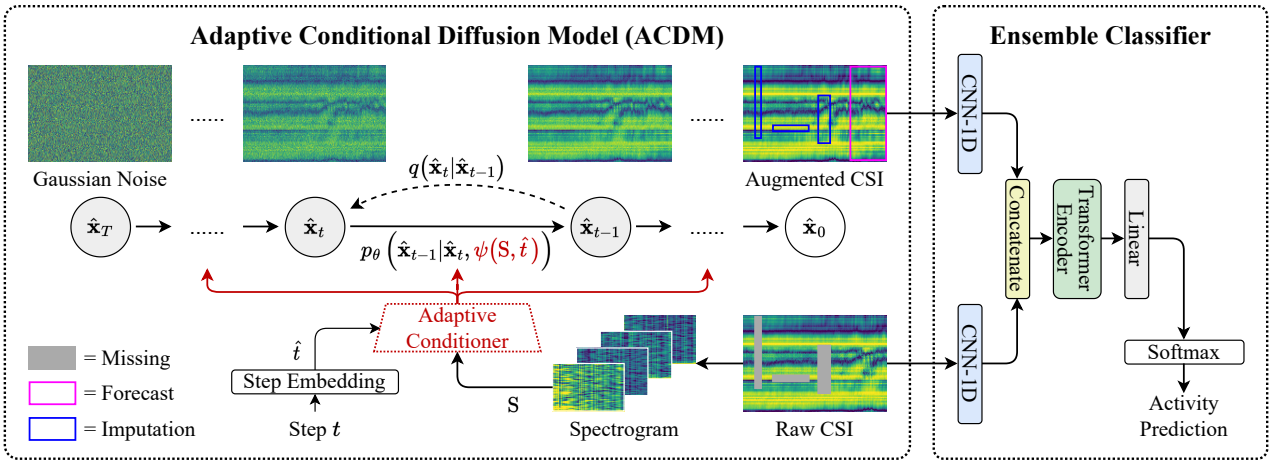

Figure 3: Overview of the proposed DiffAR.

3 DiffAR

We outline the overview of DiffAR in Figure 3, consisting of an ACDM and an ensemble classifier. ACDM synthesizes augmented CSI in line with typical diffusion models [Ho *et al.*, 2020] which generate high-quality samples from Gaussian noise by progressive steps. In contrast to typical diffusion models which guide progressive steps with constant conditions [Ho and Salimans, 2021; Rombach *et al.*, 2022], ACDM applies step-specific conditions to guide different steps. Specifically, ACDM exploits the spectrogram of CSI as input conditions, from which an adaptive conditioner distinguishes step-specific conditions that are critical to different steps. It enables ACDM to synthesize conditional patterns of different granularity in different steps. After augmentation, DiffAR feeds both raw CSI and augmented CSI to an ensemble classifier to recognize human activities.

3.1 Preliminaries

Problem Definition

Given a raw CSI sample $\mathbf{x} \in \mathbb{R}^{C \times N}$ with C channels, N denotes the time steps of its fixed window size, while λ_{miss} denotes the ratio of missing values in it. Temporal-augmented HAR includes two objectives: (1) to augment CSI samples by forecasting and imputation; (2) to recognize human activities with augmented CSI samples.

Towards the first objective, a forecasting model $g_{\text{fc}}(\cdot)$ forecasts a future sequence $\mathbf{x}_{\text{fc}} \in \mathbb{R}^{C \times N_{\text{fc}}}$ with $\mathbf{x}_{\text{fc}} = g_{\text{fc}}(\mathbf{x})$, where $N_{\text{fc}} = \lambda_{\text{fc}} N$ is the future steps to forecast, and λ_{fc} represents the forecasting ratio. Subsequently, an imputation model $g_{\text{im}}(\cdot)$ imputes the missing values in $\mathbf{x}$ by $\mathbf{x}_{\text{im}} = g_{\text{im}}(\mathbf{x})$ to obtain $\mathbf{x}_{\text{im}} \in \mathbb{R}^{C \times N}$ under the imputation ratio $\lambda_{\text{im}} = \lambda_{\text{miss}}$. After forecasting and imputation, the augmented CSI is $\hat{\mathbf{x}} = \mathbf{x}_{\text{fc}} + \mathbf{x}_{\text{im}}$, where $\hat{\mathbf{x}} \in \mathbb{R}^{C \times (1+\lambda_{\text{fc}})N}$. We formulate this self-supervised augmentation as $\hat{\mathbf{x}} = g(\mathbf{x})$.

Towards the second objective, an ensemble classifier $f(\cdot)$ uses both raw CSI and augmented CSI to predict activity label $\hat{\mathbf{y}} = f(\mathbf{x}, \hat{\mathbf{x}}) = f(\mathbf{x}, g(\mathbf{x}))$. $f(\cdot)$ aims to maximize the accuracy of $\hat{\mathbf{y}}$ with respect to the ground-true activity label $\mathbf{y}$.

Background: Diffusion Models

We apply diffusion models to augment CSI samples by forecasting and imputation. Diffusion models [Ho *et al.*, 2020] aim to learn a model distribution $p_\theta(\hat{\mathbf{x}}_0)$ to approximate a data distribution $q(\hat{\mathbf{x}}_0)$ using two mutually inverse processes: the *forward* process and the *reverse* process. The *forward* process converts $q(\hat{\mathbf{x}}_0)$ to a Gaussian distribution $q(\hat{\mathbf{x}}_T)$ with a fixed T-step Markov chain, while the *reverse* process converts a Gaussian distribution $p(\hat{\mathbf{x}}_T) = \mathcal{N}(\hat{\mathbf{x}}_T; \mathbf{0}, \mathbf{I})$ to $p_\theta(\hat{\mathbf{x}}_0)$ with a learnable T-step Markov chain. The *forward* process is formulated as $q(\hat{\mathbf{x}}_{1:T}|\hat{\mathbf{x}}_0)$ with fixed Gaussian transitions $q(\hat{\mathbf{x}}_t|\hat{\mathbf{x}}_{t-1})$ for $t = [1, ..., T]$. Conversely, the *reverse* process is formulated as $p_\theta(\hat{\mathbf{x}}_{0:T})$ with learnable Gaussian transitions $p_\theta(\hat{\mathbf{x}}_{t-1}|\hat{\mathbf{x}}_t)$ for $t = [T, ..., 1]$. Applying these formulations in practice, diffusion models optimize a denoising function $\boldsymbol{\epsilon}_\theta(\cdot)$ to synthesize $\hat{\mathbf{x}}_0$ by iterating $t = [T, ..., 1]$. We attach the detailed formulations and corresponding objective function of diffusion models in Appendix A.

3.2 Adaptive Conditional Diffusion Model

We propose ACDM in line with the formulations of diffusion models. In particular, ACDM synthesizes augmented CSI $\hat{\mathbf{x}} = \hat{\mathbf{x}}_0$ from Gaussian noise $\hat{\mathbf{x}}_T \in \mathbb{R}^{C \times (1+\lambda_{\text{fc}})N}$ by T-step progressive synthesis conditioned on CSI spectrogram $\mathbf{S}$. To the best of our knowledge, this work is the first to adopt diffusion models for CSI augmentation.

We present the network architecture of ACDM in Figure 4. To estimate the conditional denoising function $\boldsymbol{\epsilon}_\theta(\cdot)$, ACDM takes $\hat{\mathbf{x}}_t$ as inputs and uses a 5×5 convolution followed by an ReLU activation to extract both temporal and channel-wise features. To incorporate the step information into $\boldsymbol{\epsilon}_\theta(\cdot)$, ACDM performs step encoding and linear projections on each step t to obtain the step embedding $\hat{\boldsymbol{t}}$. The primary novelty of ACDM lies in two core components: the adaptive conditioner and the residual blocks. The adaptive conditioner extracts step-specific conditions from the spectrogram $\mathbf{S}$, so that ACDM can synthesize patterns of different granularity in different steps. The residual blocks apply multi-scale dilated convolutions to learn both local and global features for comprehensive synthesis. The output of each residual block acts

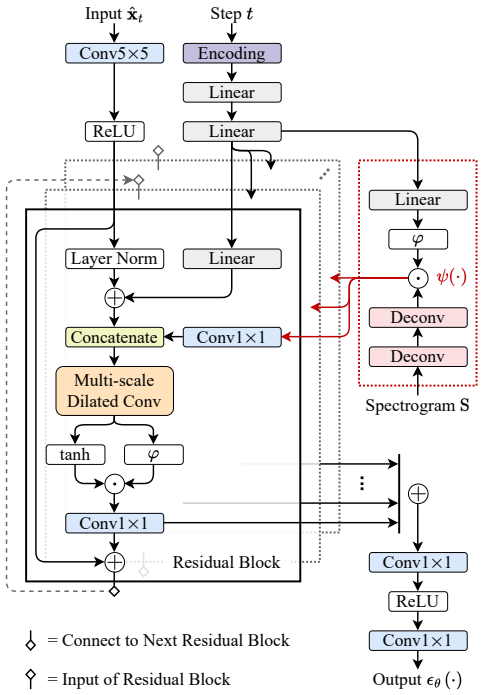

Figure 4: The network architecture of ACDM to estimate $\boldsymbol{\epsilon}_\theta(\cdot)$. $\psi(\cdot)$ is the adaptive conditioner to extract step-specific conditions.

as the input of next residual block, while the outputs of all residual blocks are summarized to estimate $\boldsymbol{\epsilon}_\theta(\cdot)$.

Adaptive Conditioner

Original diffusion models are unconditional [Ho *et al.*, 2020] and cannot be directly leveraged for CSI augmentation. To implement conditional diffusion models, a common practice [Ho and Salimans, 2021; Rombach *et al.*, 2022] is to add conditions $\mathbf{c}$ to the *reverse* process as:

$$p_\theta(\hat{\mathbf{x}}_{0:T}|\mathbf{c}) := p(\hat{\mathbf{x}}_T)\prod_{t=1}^{T} p_\theta(\hat{\mathbf{x}}_{t-1}|\hat{\mathbf{x}}_t, \mathbf{c}), \tag{1}$$

where $p_\theta(\hat{\mathbf{x}}_{t-1}|\hat{\mathbf{x}}_t, \mathbf{c}) = \mathcal{N}(\hat{\mathbf{x}}_{t-1}; \boldsymbol{\mu}_\theta(\hat{\mathbf{x}}_t, t, \mathbf{c}), \boldsymbol{\Sigma}_\theta(\hat{\mathbf{x}}_t, t, \mathbf{c}))$ is a conditional Gaussian transition. The objective function of conditional diffusion models is formulated as:

$$\mathcal{L}^{\mathbf{c}}(\theta) := \mathbb{E}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}\hat{\mathbf{x}}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t, \mathbf{c}\right)\right\|^2\right]. \tag{2}$$

Such conditional diffusion models guide progressive steps with constant conditions, but different steps may actually require step-specific information to synthesize conditional patterns of different granularity. For example, when synthesizing CSI conditioned on the spectrogram, the earlier steps (*i.e.*, the smaller t) may require the low-frequency features of spectrogram to synthesize global patterns, while the later steps (*i.e.*, the larger t) may require the high-frequency features of spectrogram to synthesize local patterns. Revisiting the preliminary of diffusion models, samples are gradually synthesized from Gaussian noise, so the variance β_t is varying along progressive steps. The variance schedule contributes to synthesize the details of samples to different extents in different steps. Unlike the variance, existing diffusion models apply constant conditions along progressive steps, where models may fail to distinguish critical information for different steps and result in limited synthesis quality.

To address this issue, we introduce a novel adaptive conditioner $\psi(\cdot)$ in ACDM to learn step-specific conditions from input conditions for different steps:

$$\mathbf{c}_t = \psi(\mathbf{S}, \hat{\boldsymbol{t}}) = \upsilon(\mathbf{S}) \odot \varphi\left(\boldsymbol{\omega}\hat{\boldsymbol{t}} + \boldsymbol{b}\right), \tag{3}$$

where $\odot$ is the element-wise multiplication, $\hat{\boldsymbol{t}}$ is the step embedding of t, and φ is a sigmoid function. $\boldsymbol{\omega}$ and $\boldsymbol{b}$ are weights and biases to compute the linear projection of $\hat{\boldsymbol{t}}$. υ is a resample function composed of deconvolutional layers (Deconv) [Zeiler *et al.*, 2010] to project conditions to the latent space.

Intuitively, for different steps, $\varphi\left(\boldsymbol{\omega}\hat{\boldsymbol{t}} + \boldsymbol{b}\right)$ acts as a step-specific filter to extract critical information from input condition features $\upsilon(\mathbf{S})$. Hence, $\mathbf{c}_t$ represents the critical conditional information for different steps. ACDM feeds $\mathbf{c}_t$ to every residual block, so the adaptive conditioner can be jointly optimized with $\boldsymbol{\epsilon}_\theta(\cdot)$. This adaptive conditioner can also expand to other conditional diffusion models to improve their synthesis quality.

Residual Blocks

The stack of residual blocks is the core component of ACDM to synthesize augmented CSI. In each residual block, we apply layer normalization [Vaswani *et al.*, 2017] on the feature maps of $\hat{\mathbf{x}}_t$, after which the linear projection of $\hat{\boldsymbol{t}}$ is added as a bias term. To guide the progressive steps in ACDM, the projection of step-specific conditions $\boldsymbol{c}_t$ is concatenated in each residual block. Further, we use a multi-scale dilated convolution layer to learn both local and global features for comprehensive synthesis. Multi-scale convolution is able to learn local features in a range-based fashion [Li *et al.*, 2021], so we utilize it in each residual block. Dilated convolution can extract global features by skipping values at certain intervals [Kong *et al.*, 2021], so we employ it over the stack of residual blocks, where the interval in each residual block follows a dilation cycle (*e.g.*, [1, 2, 4, 8]). Finally, we adopt a gated activation unit [Oord *et al.*, 2016] based on a tanh function and a sigmoid function (φ) to learn the nonlinear features.

Step Embedding

To synthesize augmented CSI by progressive steps, it is necessary to take steps as inputs to estimate $\boldsymbol{\epsilon}_\theta(\cdot)$. The adaptive conditioner also requires step information to adapt input conditions to step-specific conditions. Herein, we convert each step t into a learnable step embedding $\hat{\boldsymbol{t}}$. Step embedding involves step encoding and linear projections. We apply sine and cosine functions [Vaswani *et al.*, 2017; Kong *et al.*, 2021] to compute the step encoding $\boldsymbol{t}^e \in \mathbb{R}^M$:

$$\boldsymbol{t}^e = \left[\sin\left(10^{\frac{4m}{M/2-1}}t\right), \ldots, \cos\left(10^{\frac{4m}{M/2-1}}t\right), \ldots\right], \tag{4}$$

for $m \in [0, \ldots, (M/2-1)]$. We further adopt two linear projection layers to compute $\hat{\boldsymbol{t}} = (\boldsymbol{\omega}^1(\boldsymbol{\omega}^0\boldsymbol{t}^e + \boldsymbol{b}^0) + \boldsymbol{b}^1)$ as the step embedding, where $\boldsymbol{\omega}^0$ and $\boldsymbol{\omega}^1$ are the weights of two layers, and $\boldsymbol{b}^0$ and $\boldsymbol{b}^1$ are the biases of two layers. We formulate this step embedding as $\hat{\boldsymbol{t}} = \mathrm{embed}(t)$, which is further fed to the adaptive conditioner and every residual block.

Algorithm 1 Training

repeat

$\hat{\mathbf{x}}_0 \sim q(\mathbf{x})$ # regard raw CSI as augmented CSI

$\mathbf{S}' = \mathrm{stft}(\mathbf{x}')$ where $\mathbf{x}' = \mathrm{mask}(\hat{\mathbf{x}}_0)$

$\hat{\boldsymbol{t}} = \mathrm{embed}(t)$ where $t \sim \mathrm{Uniform}(\{1, ..., T\})$

$\mathbf{c}'_t = \psi(\mathbf{S}', \hat{\boldsymbol{t}})$ # apply the adaptive conditioner

$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

Take gradient step on

$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta \left(\sqrt{\bar{\alpha}_t}\hat{\mathbf{x}}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t, \mathbf{c}'_t \right) \right\|^2$$

until converged

Adaptive Conditional Training and Synthesis

Combining the above components, we can add the step-specific conditions to the *reverse* process in ACDM as:

$$\begin{aligned} p_\theta(\hat{\mathbf{x}}_{0:T}|\mathbf{S}) &:= p(\hat{\mathbf{x}}_T) \prod_{t=1}^{T} p_\theta(\hat{\mathbf{x}}_{t-1}|\hat{\mathbf{x}}_t, \mathbf{c}_t), \\ p_\theta(\hat{\mathbf{x}}_{t-1}|\hat{\mathbf{x}}_t, \mathbf{c}_t) &= p_\theta\left(\hat{\mathbf{x}}_{t-1}|\hat{\mathbf{x}}_t, \psi(\mathbf{S}, \hat{\boldsymbol{t}})\right). \end{aligned} \tag{5}$$

The objective function based on step-specific conditions can be formulated as:

$$\mathcal{L}^{\mathbf{a}}(\theta) := \mathbb{E}\left[\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta \left(\sqrt{\bar{\alpha}_t}\hat{\mathbf{x}}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t, \mathbf{c}_t \right) \right\|^2 \right]. \tag{6}$$

Training. We train ACDM in a self-supervised manner, where we mask certain values of raw CSI to simulate incomplete CSI $\mathbf{x}'$ and regard raw CSI as the augmented CSI $\hat{\mathbf{x}}_0$. With $\mathbf{x}' = \mathrm{mask}(\hat{\mathbf{x}}_0)$, we use random masks to simulate missing values under λ_{im} and mask the rear part of CSI to simulate the forecasting targets under λ_{fc}. We perform short-time Fourier transform (STFT) on $\mathbf{x}'$ to calculate its spectrogram $\mathbf{S}' = \mathrm{stft}(\mathbf{x}')$. Since $\hat{\mathbf{x}}_0$ acts as ground-true targets, we can train ACDM conditioned on $\mathbf{c}'_t = \psi(\mathbf{S}', \hat{\boldsymbol{t}})$ by $\min_\theta \mathcal{L}^{\mathbf{a}}(\theta)$, as illustrated in Algorithm 1.

Synthesis. After training ACDM, we can exploit it to synthesize augmented CSI samples $\hat{\mathbf{x}}_0$ based on incomplete CSI $\mathbf{x} \in \mathbb{R}^{C \times N}$. We again perform STFT to obtain the spectrogram $\mathbf{S} = \mathrm{stft}(\mathbf{x})$ and sample $\hat{\mathbf{x}}_T \in \mathbb{R}^{C \times (1+\lambda_{\mathrm{fc}})N}$ from Gaussian noise for synthesis. For each step t in $[T, .., 1]$, ACDM computes its step-specific condition $\mathbf{c}_t = \psi(\mathbf{S}, \hat{\boldsymbol{t}})$ to guide the synthesis $p_\theta(\hat{\mathbf{x}}_{t-1}|\hat{\mathbf{x}}_t, \mathbf{c}_t)$, as illustrated in Algorithm 2.

3.3 Ensemble Classifier

After augmentation, an ensemble classifier in DiffAR empolys both raw CSI and augmented CSI to recognize activities. Though ACDM has imputed the missing values in raw CSI, the positions of missing values may have certain patterns that are useful for recognition. Besides, ACDM synthesizes the augmented CSI as a whole instead of patching up raw CSI, so taking raw CSI as inputs can ensure no information loss and improve model robustness towards incomplete CSI.

In the ensemble classifier, two CNN-1D networks extract the local temporal features from inputs, after which their feature maps are concatenated for subsequent learning. A Transformer encoder [Vaswani *et al.*, 2017] further learns implicit features using the self-attention mechanism. Finally, a linear layer followed by a softmax function predicts the probability of each activity.

Algorithm 2 Synthesis

Input: incomplete CSI $\mathbf{x} \in \mathbb{R}^{C \times N}$

$\mathbf{S} = \mathrm{stft}(\mathbf{x})$

$\hat{\mathbf{x}}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ where $\hat{\mathbf{x}}_T \in \mathbb{R}^{C \times (1+\lambda_{\mathrm{fc}})N}$

for $t = T, ..., 1$ **do**

$\hat{\boldsymbol{t}} = \mathrm{embed}(t)$

$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$ else $\mathbf{z} = 0$

$\mathbf{c}_t = \psi(\mathbf{S}, \hat{\boldsymbol{t}})$ # apply the adaptive conditioner

$\hat{\mathbf{x}}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\hat{\mathbf{x}}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\hat{\mathbf{x}}_t, t, \mathbf{c}_t) \right) + \sigma_t \mathbf{z}$

end for

return $\hat{\mathbf{x}}_0$

4 Experiments

4.1 Datasets

We evaluate DiffAR on four public datasets, which differ in the number of samples, the number of activities, sample rate and window sizes. The variety of datasets enables a comprehensive evaluation. Table 1 describes the statistics of datasets. **Office [Yousefi *et al.*, 2017]** contains 557 CSI recordings of 6 individuals in an office area. As suggested by the authors, we segment these CSI recordings into 2-second windows and obtain 1984 samples, each of which owns 90 channels. **SignFi [Ma *et al.*, 2018]** involves 276 activities (sign language gestures) captured by WiFi CSI with 90 channels. Each activity comprises 30 samples for recognition. **Interactions [Alazrai *et al.*, 2020]** consists of CSI samples with 180 channels monitoring 12 human-to-human interactions between 40 pairs of individuals. **Widar 3.0 [Zhang *et al.*, 2021]** includes CSI samples with 90 channels collected in 15 days. We use the samples of 6 activities from 4 individuals for evaluation.

4.2 Baselines

We compare DiffAR with 11 baselines to demonstrate its effectiveness. To examine the quality of augmented CSI, we compare DiffAR with the following state-of-the-art generative time-series models. (1) **TimeGrad** [Rasul *et al.*, 2021] combined diffusion models with RNNs for time-series forecasting. (2) **CSDI** [Tashiro *et al.*, 2021] applied diffusion models based on Transformer encoders for time-series imputation. (3) **WaveGrad** [Chen *et al.*, 2020] utilized diffusion models with a gradient-based sampler for waveform synthesis. (4) **DiffWave** [Kong *et al.*, 2021] synthesized waveform using diffusion models based on dilated convolutions.

To evaluate the recognition performance, we compare DiffAR with the following CSI-based HAR methods. (1) **ST-RF** [Yousefi *et al.*, 2017] employed STFT to extract handcrafted features for HAR. (2) **LSTM** [Yousefi *et al.*, 2017] learned

Datasets	Samples	Activities	Rate (Hz)	Window (s)
Office	1984	7	1000	2.00
SignFi	8280	276	12.5∼200	1.00∼16.0
Interactions	4800	12	320	3.25∼7.03
Widar 3.0	17986	6	1000	0.26∼3.90

Table 1: Statistics of four public CSI-based HAR datasets.

Models	Ratio λ_{fc}	Ratio λ_{im}	Office MAE	Office MSE	Office CRPS	SignFi MAE	SignFi MSE	SignFi CRPS	Interactions MAE	Interactions MSE	Interactions CRPS	Widar 3.0 MAE	Widar 3.0 MSE	Widar 3.0 CRPS
Forecast														
TimeGrad	0.2	0.0	1.074	1.886	1.321	0.938	1.596	1.351	1.247	2.626	1.406	1.028	1.696	1.282
CSDI	0.2	0.0	1.084	1.764	1.363	0.856	1.246	1.199	0.864	1.204	1.100	1.038	1.661	1.285
WaveGrad	0.2	0.0	0.862	1.181	1.084	0.730	1.031	1.020	0.860	1.212	1.094	0.878	1.210	1.087
DiffWave	0.2	0.0	0.848	1.141	1.066	0.779	1.154	1.089	0.835	1.126	1.062	0.858	1.140	1.061
DiffAR (Ours)	0.2	0.0	**0.819**	**1.071**	**1.029**	**0.721**	**1.010**	**1.007**	**0.811**	**1.058**	**1.032**	**0.816**	**1.019**	**1.009**
Imputation														
CSDI	0.0	0.2	1.064	1.708	1.338	0.856	1.246	1.200	0.860	1.187	1.095	1.006	1.599	1.246
WaveGrad	0.0	0.2	0.868	1.215	1.091	0.730	1.030	1.019	0.864	1.247	1.099	0.881	1.237	1.090
DiffWave	0.0	0.2	0.855	1.166	1.074	0.784	1.164	1.095	0.840	1.153	1.069	0.864	1.167	1.069
DiffAR (Ours)	0.0	0.2	**0.811**	**1.117**	**1.019**	**0.718**	**1.003**	**1.002**	**0.808**	**1.066**	**1.028**	**0.827**	**1.051**	**1.023**
Forecast + Imputation														
CSDI	0.2	0.2	1.046	1.667	1.315	0.856	1.245	1.199	0.889	1.265	1.132	0.986	1.537	1.221
WaveGrad	0.2	0.2	0.862	1.195	1.083	0.729	1.029	1.018	0.856	1.216	1.089	0.874	1.211	1.082
DiffWave	0.2	0.2	0.850	1.150	1.068	0.754	1.096	1.054	0.834	1.132	1.060	0.858	1.147	1.062
DiffAR (Ours)	0.2	0.2	**0.822**	**1.134**	**1.033**	**0.717**	**1.003**	**1.001**	**0.809**	**1.068**	**1.028**	**0.817**	**1.023**	**1.011**

Table 2: The quality of augmented CSI using different generative time-series models in terms of Mean Absolute Error (MAE), Mean Squared Error (MSE) and Continuous Ranked Probability Score (CRPS). Lower results indicate better quality. **Bold** highlights the best results.

temporal features for HAR. (3) **CNN-1D** [Wang *et al.*, 2019] applied convolutions to learn local spatial features. (4) **CNN-2D** [Moshiri *et al.*, 2021] regarded CSI as images to learn local channel-wise features. (5) **CNN-LSTM** [Shalaby *et al.*, 2022] combined CNN with LSTM to learn both temporal and spatial features. (6) **ABLSTM** [Chen *et al.*, 2018] equipped bi-directional LSTM with attention to learn feature dependencies. (7) **THAT** [Li *et al.*, 2021] exploited both attention and convolutions to outperform other CSI-based HAR methods.

4.3 Evaluation Metrics

To measure the quality of augmented CSI, we adopt three common metrics for time-series models, including Mean Absolute Error (MAE), Mean Squared Error (MSE) and Continuous Ranked Probability Score (CRPS) [Tashiro *et al.*, 2021]. MAE calculates the absolute differences between synthesized samples and ground-true samples, while MSE calculates their squared differences. CRPS [Matheson and Winkler, 1976] evaluates the compatibility of generative distributions with ground-true observations [Rasul *et al.*, 2021].

To measure the recognition performance, we employ Accuracy (Acc.), Weighted Precision (WP) and F1 score as metrics. Accuracy indicates the performance of classifying all activities, while WP summarizes the recognition precision of each activity as a weighted average. F1 score is the harmonic mean of Precision and Recall for comprehensive evaluation.

4.4 Implementation Details

In ACDM, we establish 10 residual blocks whose dimension for skip connections is 32. Each residual block applies multi-scale dilated convolutions whose kernel sizes are $\{1, 3, 5\}$, and the dilation cycle across these blocks is $[1, 2, 4, 8, 16]$. The dimension of step embedding is set to $M = 128$. To use CSI spectrogram as conditions, we set the size of STFT to 256, and the hop length to 64. We adopt a linear spaced noise schedule where $\beta_t \in \left[10^{-5}, 10^{-2}\right]$ with diffusion steps $T = 100$. In the ensemble classifier, each CNN-1D network contains 3 convolutional layers whose numbers of filters are $\{32, 64, 128\}$ and kernel sizes are $\{7, 5, 3\}$ with strides $\{3, 2, 1\}$. Each convolutional layer is followed by an ReLU activation with a dropout rate of 0.1. After concatenation, the feature dimension becomes 256, which is the input dimension of Transformer encoder. The Transformer encoder contains 2 encoder layers, where the number of heads is 8.

We train ACDM in a self-supervised manner, as mentioned in Section 3.2. (1) To evaluate the quality of augmented CSI, we apply masks to raw CSI and augment the masked CSI using DiffAR or other generative time-series models. We assess the quality by measuring the similarity between augmented CSI and raw CSI, where we set $\lambda_{\text{im}} = \lambda_{\text{fc}} = 0.2$. (2) To evaluate the recognition performance, we further augment raw CSI using DiffAR or other generative time-series models. Specifically, we simulate the missing values with random masks and lengthen the raw CSI by forecasting. With the further augmented CSI, we compare the performance of CSI-based HAR methods, where we set $\lambda_{\text{im}} = 0.5$ and $\lambda_{\text{fc}} = 0.2$. (3) We further conduct a hyper-parameter sensitivity study of DiffAR, where we set $\lambda_{\text{fc}} = \lambda_{\text{im}} = \{0.2, 0.4, 0.6, 0.8\}$, as attached in Appendix B.

We implement DiffAR using Pytorch 1.13 with Python 3.9 and train it on a single Nvidia RTX A5000 GPU. The model is optimized by Adam [Kingma and Ba, 2014] with a fixed learning rate 10^{-4} and the batch size of 16. Each dataset is splited into a training set (80%), a validation set (10%), and a test set (10%). We leverage training sets to optimize ACDM for 10^5 epochs and exploit the trained ACDM to augment all three sets. The augmented training sets are used to optimize the ensemble classifier for 200 epochs. We apply validation sets to select the best models for evaluation on test sets.

Methods	Ratio λ_{fc}	Ratio λ_{im}	Office Acc.	Office WP	Office F1	SignFi Acc.	SignFi WP	SignFi F1	Interactions Acc.	Interactions WP	Interactions F1	Widar 3.0 Acc.	Widar 3.0 WP	Widar 3.0 F1
Baselines														
ST-RF	0.0	0.0	89.95	90.58	85.61	84.06	88.98	82.37	75.42	75.23	74.58	55.42	56.06	55.39
LSTM	0.0	0.0	94.44	94.53	91.50	88.40	90.99	86.15	82.71	82.67	81.89	67.13	67.07	66.73
CNN-1D	0.0	0.0	95.45	95.86	93.47	**97.34**	**98.04**	**97.36**	82.92	83.05	82.45	77.98	78.69	78.19
CNN-2D	0.0	0.0	96.46	96.67	94.85	97.34	97.77	96.19	90.21	90.90	89.86	87.99	88.02	87.95
CNN-LSTM	0.0	0.0	91.92	92.32	88.58	88.04	91.04	87.37	76.25	77.51	75.64	70.52	70.91	70.63
ABLSTM	0.0	0.0	95.96	96.13	94.92	96.38	97.04	95.60	86.46	86.88	85.92	73.47	73.50	73.36
THAT	0.0	0.0	**96.97**	**97.02**	**95.85**	96.74	97.42	96.29	**90.63**	**91.19**	**90.30**	**90.04**	**90.06**	**90.01**
Forecast														
THAT + TimeGrad	0.2	0.0	97.49	97.51	96.25	95.65	96.76	95.58	90.83	91.36	90.68	91.22	91.32	91.21
THAT + DiffWave	0.2	0.0	96.98	97.04	95.99	97.10	97.92	96.28	90.83	91.13	90.75	91.39	91.53	91.40
DiffAR (Ours)	0.2	0.0	**97.99**	**98.10**	**97.16**	**98.07**	**98.78**	**97.72**	**94.17**	**94.37**	**94.02**	**91.78**	**91.76**	**91.71**
Imputation														
CNN-1D + DiffWave	0.0	0.5	95.48	95.46	93.76	97.58	98.01	96.92	85.42	85.62	85.07	81.39	81.65	81.14
THAT + DiffWave	0.0	0.5	96.48	96.40	94.46	96.98	97.75	95.47	90.63	90.78	90.44	91.11	91.09	91.07
DiffAR (Ours)	0.0	0.5	**97.99**	**97.95**	**96.82**	**97.95**	**98.48**	**97.39**	**93.75**	**93.96**	**93.40**	**91.67**	**91.67**	**91.61**
Forecast + Imputation														
CNN-1D + DiffWave	0.2	0.5	96.98	97.01	95.54	97.71	98.03	96.80	85.83	86.18	85.46	82.17	82.41	82.03
THAT + DiffWave	0.2	0.5	96.48	96.76	95.03	97.22	97.48	95.60	90.63	90.97	90.22	90.61	90.62	90.58
DiffAR (Ours)	0.2	0.5	**98.49**	**98.54**	**98.22**	**98.19**	**98.59**	**98.22**	**94.58**	**94.67**	**94.50**	**92.06**	**92.19**	**92.04**

Table 3: The recognition performance (unit: %) of CSI-based HAR methods in terms of Accuracy (Acc.), Weighted Precision (WP) and F1 score. Higher results indicate better performance. **Bold** highlights the best results.

4.5 Results and Discussions

Table 2 compares the quality of augmented CSI with different generative time-series models, and Table 3 presents the recognition performance of CSI-based HAR methods.

DiffAR achieves the best quality of augmented CSI. We compare DiffAR with four generative time-series models regarding the quality of forecast, imputation and forecast + imputation. DiffAR obtains better quality than other models in all these situations, as shown in Table 2. Compared with other models forecasting CSI, DiffAR reduces MAE, MSE and CRPS by 1.2∼35.0%, 2.0∼59.7% and 1.3∼26.6%, respectively. For CSI imputation, DiffAR realizes 1.7∼23.8% lower MAE, 2.7∼34.6% lower MSE, and 1.7∼23.9% lower CRPS than other models. If we preform both forecasting and imputation, DiffAR outperforms other models by 1.6∼21.4% on MAE, 1.4∼33.4% on MSE, and 1.7∼21.5% on CRPS. DiffAR outperforms other models since it adopts multi-scale dilated convolutions to learn both local and global features, while other models either failed to extract long-range feature dependencies (TimeGrad), or did not consider channel-wise features (WaveGrad and DiffWave). More critically, DiffAR can learn step-specific conditions for progressive steps to synthesize high-quality samples under different granularity.

DiffAR outperforms state-of-the-art CSI-based HAR methods. Compared with existing CSI-based HAR baselines without augmentation, DiffAR attains better performance with augmented CSI, as shown in Table 3. In contrast to the best baselines without forecasts, DiffAR increases the accuracy by 0.75∼3.9%. For imputation-augmented HAR, the accuracy of DiffAR is 0.62∼3.44% higher than that of the baselines. If we augment CSI by forecasting and imputation, DiffAR outperforms the baselines by 0.87∼4.35% on accuracy. Similar results can be observed in terms of WP and F1. We also discuss the impact of missing values in Appendix C and conduct an ablation study in Appendix D.

To further illustrate the effectiveness of DiffAR, we equip CSI-based HAR baselines with generative models for comparison. For forecast-augmented HAR, DiffAR obviously excels THAT assisted by TimeGrad or DiffWave, though they have already achieved better performance than THAT without forecasts. Compared with imputation-augmented HAR baselines, DiffAR also attains the highest accuracy, WP, and F1. With both forecasting and imputation, the accuracy of DiffAR outperforms the second best results by 0.49∼4.35%.

In summary, using generative time-series models to temporally augment CSI can enhance the performance of CSI-based HAR. DiffAR achieves the best quality of augmented CSI and thus outperforms state-of-the-art CSI-based HAR methods.

5 Conclusion

We propose DiffAR as a pioneering work in WiFi sensing to augment incomplete CSI with diffusion models and improve CSI-based HAR. In DiffAR, we devise ACDM to forecast CSI from fixed windows and to impute missing values in CSI. ACDM adopts a novel adaptive conditioner which learns step-specific conditions for progressive steps to synthesize conditional patterns of different granularity. It proves the feasibility of using step-specific conditions to improve synthesis quality and can expand to other conditional diffusion models. Extensive experiments illustrate that DiffAR achieves the best quality of augmented CSI and outperforms state-of-the-art CSI-based HAR methods in recognition performance.

A Diffusion Models

Diffusion models [Ho *et al.*, 2020] aim to learn a model distribution $p_\theta(\hat{\mathbf{x}}_0)$ to approximate a data distribution $q(\hat{\mathbf{x}}_0)$ using two mutually inverse processes: the *forward* process and the *reverse* process. The *forward* process converts $q(\hat{\mathbf{x}}_0)$ to a Gaussian distribution $q(\hat{\mathbf{x}}_T)$ with a fixed T-step Markov chain, while the *reverse* process converts a Gaussian distribution $p(\hat{\mathbf{x}}_T) = \mathcal{N}(\hat{\mathbf{x}}_T; \mathbf{0}, \mathbf{I})$ to $p_\theta(\hat{\mathbf{x}}_0)$ with a learnable T-step Markov chain. The *forward* process is formulated as:

$$q(\hat{\mathbf{x}}_{1:T}|\hat{\mathbf{x}}_0) := \prod_{t=1}^{T} q(\hat{\mathbf{x}}_t|\hat{\mathbf{x}}_{t-1}), \tag{7}$$

where $q(\hat{\mathbf{x}}_t|\hat{\mathbf{x}}_{t-1}) = \mathcal{N}\left(\hat{\mathbf{x}}_{t-1}; \sqrt{1-\beta_t}\hat{\mathbf{x}}_{t-1}, \beta_t\mathbf{I}\right)$ is a fixed Gaussian transition with variance $\beta_t \in (0,1)$. $q(\hat{\mathbf{x}}_{1:T}|\hat{\mathbf{x}}_0)$ converts $q(\hat{\mathbf{x}}_0)$ to $q(\hat{\mathbf{x}}_T)$ by gradually adding noise to data with an increasing variance schedule $[\beta_1, ..., \beta_T]$.

Conversely, the *reverse* process is formulated as:

$$p_\theta(\hat{\mathbf{x}}_{0:T}) := p(\hat{\mathbf{x}}_T) \prod_{t=1}^{T} p_\theta(\hat{\mathbf{x}}_{t-1}|\hat{\mathbf{x}}_t), \tag{8}$$

where $p_\theta(\hat{\mathbf{x}}_{t-1}|\hat{\mathbf{x}}_t) = \mathcal{N}(\hat{\mathbf{x}}_{t-1}; \boldsymbol{\mu}_\theta(\hat{\mathbf{x}}_t, t), \boldsymbol{\Sigma}_\theta(\hat{\mathbf{x}}_t, t))$ is a learnable Gaussian transition with parameters θ. The joint distribution $p_\theta(\hat{\mathbf{x}}_{0:T})$ converts $p(\hat{\mathbf{x}}_T)$ to $p_\theta(\hat{\mathbf{x}}_0)$. The objective of diffusion models is to minimize the negative log likelihood $\mathbb{E}[-\log p_\theta(\hat{\mathbf{x}}_0)]$, which is equivalent to minimize the Kullback-Leibler (KL) divergence between $p_\theta(\hat{\mathbf{x}}_{t-1}|\hat{\mathbf{x}}_t)$ and $q(\hat{\mathbf{x}}_{t-1}|\hat{\mathbf{x}}_t, \hat{\mathbf{x}}_0)$. Since they are both Gaussian distributions, the mean and variance of $p_\theta(\hat{\mathbf{x}}_{t-1}|\hat{\mathbf{x}}_t)$ should fit the mean and variance of $q(\hat{\mathbf{x}}_{t-1}|\hat{\mathbf{x}}_t, \hat{\mathbf{x}}_0)$, respectively. The mean and variance of $q(\hat{\mathbf{x}}_{t-1}|\hat{\mathbf{x}}_t, \hat{\mathbf{x}}_0)$ can be formulated as:

$$\tilde{\boldsymbol{\mu}}_t = \frac{1}{\sqrt{\alpha_t}}\left(\hat{\mathbf{x}}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\right) \text{ and } \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t, \tag{9}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$. Thus, the mean and variance of $p_\theta(\hat{\mathbf{x}}_{t-1}|\hat{\mathbf{x}}_t)$ are parameterized as:

$$\begin{aligned} \boldsymbol{\mu}_\theta(\hat{\mathbf{x}}_t, t) &= \frac{1}{\sqrt{\alpha_t}}\left(\hat{\mathbf{x}}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\hat{\mathbf{x}}_t, t)\right), \\ \boldsymbol{\Sigma}_\theta(\hat{\mathbf{x}}_t, t) &= \sigma_t^2\mathbf{I}, \text{ where } \sigma_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \end{aligned} \tag{10}$$

$\boldsymbol{\epsilon}_\theta(\cdot)$ is a denoising function to estimate noise $\boldsymbol{\epsilon}$ from $\hat{\mathbf{x}}_t$. Since both $p_\theta(\hat{\mathbf{x}}_{t-1}|\hat{\mathbf{x}}_t)$ and $q(\hat{\mathbf{x}}_{t-1}|\hat{\mathbf{x}}_t, \hat{\mathbf{x}}_0)$ have constant variance, the KL divergence between them can be simplified to the distance between $\tilde{\boldsymbol{\mu}}_t$ and $\boldsymbol{\mu}_\theta(\hat{\mathbf{x}}_t, t)$:

$$\begin{aligned} \mathcal{L}(\theta) :=& \mathbb{E}_{\hat{\mathbf{x}}_0 \sim q(\hat{\mathbf{x}}_0)}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\hat{\mathbf{x}}_t, t)\right\|^2\right], \\ =& \mathbb{E}_{\hat{\mathbf{x}}_0 \sim q(\hat{\mathbf{x}}_0)}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}\hat{\mathbf{x}}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t\right)\right\|^2\right], \end{aligned} \tag{11}$$

which is the objective function to optimize parameters θ. After training the denoising function $\boldsymbol{\epsilon}_\theta(\cdot)$ by $\min_\theta \mathcal{L}(\theta)$, we can use $p_\theta(\hat{\mathbf{x}}_{t-1}|\hat{\mathbf{x}}_t)$ to synthesize $\hat{\mathbf{x}}_0$ by iterating $t = [T, ..., 1]$:

$$\hat{\mathbf{x}}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\hat{\mathbf{x}}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\hat{\mathbf{x}}_t, t)\right) + \sigma_t\mathbf{z}, \tag{12}$$

where $\hat{\mathbf{x}}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ with $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ when $t > 1$ and $\mathbf{z} = 0$ when $t = 1$.

B Hyper-parameter Sensitivity Study

Herein, we study the impact of hyper-parameters λ_{fc} and λ_{im} on the quality of augmented CSI using DiffAR.

B.1 Dataset: Office

Table 4 presents the results of hyper-parameter sensitivity study of DiffAR on dataset Office [Yousefi *et al.*, 2017].

λ_{fc}	λ_{im}	MAE (Stdev)	MSE (Stdev)	CRPS (Stdev)
0.2	0.0	0.819 (0.017)	1.071 (0.009)	1.029 (0.006)
0.4	0.0	0.836 (0.022)	1.109 (0.040)	1.051 (0.021)
0.6	0.0	0.884 (0.036)	1.248 (0.093)	1.111 (0.042)
0.8	0.0	0.893 (0.040)	1.285 (0.113)	1.122 (0.047)
0.0	0.2	0.811 (0.066)	1.117 (0.134)	1.019 (0.083)
0.0	0.4	0.854 (0.023)	1.177 (0.033)	1.074 (0.022)
0.0	0.6	0.900 (0.028)	1.328 (0.051)	1.132 (0.033)
0.0	0.8	0.924 (0.022)	1.418 (0.024)	1.161 (0.023)
0.2	0.2	0.822 (0.058)	1.134 (0.121)	1.033 (0.074)
0.4	0.4	0.841 (0.020)	1.133 (0.025)	1.057 (0.017)
0.6	0.6	0.870 (0.023)	1.231 (0.031)	1.093 (0.021)
0.8	0.8	0.892 (0.022)	1.322 (0.033)	1.121 (0.019)
0.2	0.8	0.891 (0.021)	1.302 (0.025)	1.119 (0.021)
0.4	0.6	0.908 (0.024)	1.352 (0.036)	1.142 (0.025)
0.6	0.4	0.906 (0.028)	1.339 (0.050)	1.139 (0.030)
0.8	0.2	0.892 (0.031)	1.286 (0.069)	1.121 (0.034)

Table 4: The quality of augmented CSI on dataset Office using DiffAR with different λ_{fc} and λ_{im} in terms of Mean Absolute Error (MAE), Mean Squared Error (MSE) and Continuous Ranked Probability Score (CRPS). ("Stdev": Standard Deviation)

B.2 Dataset: SignFi

Table 5 presents the results of hyper-parameter sensitivity study of DiffAR on dataset SignFi [Ma *et al.*, 2018].

λ_{fc}	λ_{im}	MAE (Stdev)	MSE (Stdev)	CRPS (Stdev)
0.2	0.0	0.721 (0.061)	1.010 (0.008)	1.007 (0.005)
0.4	0.0	0.824 (0.051)	1.262 (0.037)	1.153 (0.037)
0.6	0.0	0.921 (0.046)	1.565 (0.036)	1.290 (0.055)
0.8	0.0	0.952 (0.047)	1.806 (0.033)	1.333 (0.053)
0.0	0.2	0.718 (0.061)	1.003 (0.004)	1.002 (0.003)
0.0	0.4	0.808 (0.052)	1.219 (0.028)	1.130 (0.030)
0.0	0.6	0.889 (0.047)	1.470 (0.023)	1.244 (0.046)
0.0	0.8	0.918 (0.048)	1.720 (0.022)	1.285 (0.045)
0.2	0.2	0.717 (0.061)	1.003 (0.004)	1.001 (0.003)
0.4	0.4	0.799 (0.053)	1.197 (0.032)	1.117 (0.030)
0.6	0.6	0.880 (0.048)	1.446 (0.025)	1.231 (0.045)
0.8	0.8	0.911 (0.049)	1.709 (0.023)	1.274 (0.043)
0.2	0.8	0.920 (0.049)	1.731 (0.022)	1.288 (0.045)
0.4	0.6	0.931 (0.048)	1.760 (0.024)	1.304 (0.048)
0.6	0.4	0.939 (0.048)	1.776 (0.028)	1.314 (0.050)
0.8	0.2	0.943 (0.048)	1.786 (0.031)	1.320 (0.051)

Table 5: The quality of augmented CSI on dataset SignFi using DiffAR with different λ_{fc} and λ_{im} in terms of Mean Absolute Error (MAE), Mean Squared Error (MSE) and Continuous Ranked Probability Score (CRPS). ("Stdev": Standard Deviation)

B.3 Dataset: Interactions

Table 6 presents the results of hyper-parameter sensitivity study of DiffAR on dataset Interactions [Alazrai *et al.*, 2020].

λ_{fc}	λ_{im}	MAE (Stdev)	MSE (Stdev)	CRPS (Stdev)
0.2	0.0	0.811 (0.028)	1.058 (0.033)	1.032 (0.017)
0.4	0.0	0.863 (0.035)	1.186 (0.074)	1.097 (0.035)
0.6	0.0	0.940 (0.040)	1.408 (0.097)	1.196 (0.043)
0.8	0.0	0.945 (0.037)	1.425 (0.089)	1.203 (0.040)
0.0	0.2	0.808 (0.028)	1.066 (0.025)	1.028 (0.013)
0.0	0.4	0.880 (0.032)	1.281 (0.042)	1.120 (0.024)
0.0	0.6	0.962 (0.029)	1.554 (0.033)	1.224 (0.024)
0.0	0.8	0.951 (0.024)	1.525 (0.021)	1.210 (0.020)
0.2	0.2	0.809 (0.033)	1.068 (0.040)	1.028 (0.022)
0.4	0.4	0.866 (0.031)	1.230 (0.040)	1.101 (0.021)
0.6	0.6	0.927 (0.032)	1.420 (0.051)	1.178 (0.024)
0.8	0.8	0.905 (0.031)	1.351 (0.047)	1.151 (0.021)
0.2	0.8	0.905 (0.029)	1.378 (0.026)	1.151 (0.018)
0.4	0.6	0.931 (0.033)	1.445 (0.047)	1.184 (0.025)
0.6	0.4	0.940 (0.034)	1.449 (0.058)	1.195 (0.027)
0.8	0.2	0.935 (0.033)	1.411 (0.065)	1.190 (0.029)

Table 6: The quality of augmented CSI on dataset Interactions using DiffAR with different λ_{fc} and λ_{im} in terms of Mean Absolute Error (MAE), Mean Squared Error (MSE) and Continuous Ranked Probability Score (CRPS). ("Stdev": Standard Deviation)

B.4 Dataset: Widar 3.0

Table 7 presents the results of hyper-parameter sensitivity study of DiffAR on dataset Widar 3.0 [Zhang *et al.*, 2021].

λ_{fc}	λ_{im}	MAE (Stdev)	MSE (Stdev)	CRPS (Stdev)
0.2	0.0	0.816 (0.026)	1.019 (0.016)	1.009 (0.009)
0.4	0.0	0.879 (0.024)	1.189 (0.043)	1.087 (0.023)
0.6	0.0	0.937 (0.024)	1.363 (0.059)	1.159 (0.031)
0.8	0.0	0.936 (0.024)	1.365 (0.058)	1.158 (0.030)
0.0	0.2	0.827 (0.026)	1.051 (0.019)	1.023 (0.009)
0.0	0.4	0.904 (0.022)	1.288 (0.023)	1.118 (0.021)
0.0	0.6	0.954 (0.019)	1.461 (0.014)	1.181 (0.025)
0.0	0.8	0.969 (0.020)	1.506 (0.021)	1.199 (0.027)
0.2	0.2	0.817 (0.027)	1.023 (0.013)	1.011 (0.008)
0.4	0.4	0.887 (0.024)	1.230 (0.027)	1.097 (0.019)
0.6	0.6	0.923 (0.021)	1.350 (0.022)	1.142 (0.020)
0.8	0.8	0.941 (0.022)	1.404 (0.028)	1.164 (0.023)
0.2	0.8	0.938 (0.020)	1.403 (0.018)	1.161 (0.023)
0.4	0.6	0.950 (0.021)	1.437 (0.027)	1.176 (0.025)
0.6	0.4	0.947 (0.022)	1.419 (0.034)	1.171 (0.026)
0.8	0.2	0.936 (0.023)	1.375 (0.042)	1.158 (0.027)

Table 7: The quality of augmented CSI on dataset Widar 3.0 using DiffAR with different λ_{fc} and λ_{im} in terms of Mean Absolute Error (MAE), Mean Squared Error (MSE) and Continuous Ranked Probability Score (CRPS). ("Stdev": Standard Deviation)

C The Impact of Missing Values in CSI on CSI-based HAR Methods

Table 8 presents the recognition performance of THAT [Li *et al.*, 2021] on four datasets under different ratios of missing values in CSI. The performance of THAT significantly decreases along with the increasing ratios of missing values.

	λ_{miss}	THAT^0 Acc.	WP	F1	THAT^1 Acc.	WP	F1
Office	0.00	96.97	97.02	95.85	96.97	97.02	95.85
	0.05	91.92	92.51	89.25	88.38	88.88	84.39
	0.10	84.34	84.66	78.46	83.33	85.46	78.45
	0.20	79.80	81.20	76.92	75.76	80.32	65.44
	0.50	64.65	68.19	58.60	56.06	57.83	36.02
SignFi	0.00	96.74	97.42	96.29	96.74	97.42	96.29
	0.05	92.03	92.66	89.72	93.24	94.67	91.44
	0.10	84.78	88.53	80.47	86.96	90.10	85.20
	0.20	63.53	71.31	59.53	66.30	74.91	63.59
	0.50	23.67	28.51	20.70	24.88	32.91	22.19
Interactions	0.00	90.63	91.19	90.30	90.63	91.19	90.30
	0.05	84.38	85.15	83.90	89.38	90.22	88.88
	0.10	80.83	81.23	80.13	88.33	88.99	87.89
	0.20	83.75	83.90	83.01	86.67	87.51	86.29
	0.50	74.38	74.99	73.72	71.88	75.74	71.75
Widar 3.0	0.00	90.04	90.06	90.01	90.04	90.06	90.01
	0.05	70.75	70.86	70.71	55.23	72.42	48.90
	0.10	67.63	67.99	67.55	49.05	66.53	39.15
	0.20	61.79	62.59	61.49	45.83	61.85	33.95
	0.50	52.56	51.05	50.65	41.94	24.41	30.02

Table 8: Performance (unit: %) of THAT under different ratios of missing values in CSI in terms of Accuracy (Acc.), Weighted Precision (WP) and F1 score. THAT^0 is tuned by samples with missing values, while THAT^1 is not tuned by samples with missing values.

D Ablation Study

Table 9 presents the results of ablation study involving DiffAR without (w/o) Adaptive Conditioner or ACDM, where we set $\lambda_{\text{fc}} = 0.2$ and $\lambda_{\text{im}} = 0.5$.

Datasets	Models	Acc.	WP	F1
Office	**DiffAR (Ours)**	**98.49**	**98.54**	**98.22**
	w/o Adaptive Conditioner	95.48	95.42	93.70
	w/o ACDM	71.85	71.99	66.10
SignFi	**DiffAR (Ours)**	**98.19**	**98.59**	**98.22**
	w/o Adaptive Conditioner	97.22	97.94	95.93
	w/o ACDM	55.80	63.04	50.04
Interactions	**DiffAR (Ours)**	**94.58**	**94.67**	**94.50**
	w/o Adaptive Conditioner	91.67	92.51	91.34
	w/o ACDM	77.29	78.66	76.33
Widar 3.0	**DiffAR (Ours)**	**92.06**	**92.19**	**92.04**
	w/o Adaptive Conditioner	88.50	88.44	88.36
	w/o ACDM	58.44	59.02	57.59

Table 9: Ablation results (unit: %) of DiffAR in terms of Accuracy (Acc.), Weighted Precision (WP) and F1 score.

References

[Alazrai *et al.*, 2020] Rami Alazrai, Ali Awad, Alsaify Baha'A, Mohammad Hababeh, and Mohammad I Daoud. A dataset for wi-fi-based human-to-human interaction recognition. *Data in brief*, 31:105668, 2020.

[An and Ogras, 2021] Sizhe An and Umit Y Ogras. Mars: mmwave-based assistive rehabilitation system for smart healthcare. *ACM Transactions on Embedded Computing Systems (TECS)*, 20(5s):1–22, 2021.

[Bianchi *et al.*, 2019] Valentina Bianchi, Marco Bassoli, Gianfranco Lombardo, Paolo Fornacciari, Monica Mordonini, and Ilaria De Munari. Iot wearable sensor and deep learning: An integrated approach for personalized human activity recognition in a smart home environment. *IEEE Internet of Things Journal*, 6(5):8553–8562, 2019.

[Chen *et al.*, 2018] Zhenghua Chen, Le Zhang, Chaoyang Jiang, Zhiguang Cao, and Wei Cui. Wifi csi based passive human activity recognition using attention based blstm. *IEEE Transactions on Mobile Computing*, 18(11):2714–2724, 2018.

[Chen *et al.*, 2020] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713*, 2020.

[Esteban *et al.*, 2017] Cristóbal Esteban, Stephanie L Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633*, 2017.

[Goodfellow *et al.*, 2020] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[Ho and Salimans, 2021] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[Ho *et al.*, 2020] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[Hussain *et al.*, 2020] Zawar Hussain, Quan Z Sheng, and Wei Emma Zhang. A review and categorization of techniques on device-free human activity recognition. *Journal of Network and Computer Applications*, 167:102738, 2020.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Kong *et al.*, 2021] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021.

[Li *et al.*, 2021] Bing Li, Wei Cui, Wei Wang, Le Zhang, Zhenghua Chen, and Min Wu. Two-stream convolution augmented transformer for human activity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 286–293, 2021.

[Lin *et al.*, 2020] Yuxiang Lin, Yi Gao, Bingji Li, and Wei Dong. Revisiting indoor intrusion detection with wifi signals: do not panic over a pet! *IEEE Internet of Things Journal*, 7(10):10437–10449, 2020.

[Ma *et al.*, 2018] Yongsen Ma, Gang Zhou, Shuangquan Wang, Hongyang Zhao, and Woosub Jung. Signfi: Sign language recognition using wifi. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(1):1–21, 2018.

[Ma *et al.*, 2019] Yongsen Ma, Gang Zhou, and Shuangquan Wang. Wifi sensing with channel state information: A survey. *ACM Computing Surveys (CSUR)*, 52(3):1–36, 2019.

[Matheson and Winkler, 1976] James E Matheson and Robert L Winkler. Scoring rules for continuous probability distributions. *Management science*, 22(10):1087–1096, 1976.

[Mogren, 2016] Olof Mogren. C-rnn-gan: Continuous recurrent neural networks with adversarial training. *arXiv preprint arXiv:1611.09904*, 2016.

[Moshiri *et al.*, 2021] Parisa Fard Moshiri, Mohammad Nabati, Reza Shahbazian, and Seyed Ali Ghorashi. Csi-based human activity recognition using convolutional neural networks. In *2021 11th International Conference on Computer Engineering and Knowledge (ICCKE)*, pages 7–12. IEEE, 2021.

[Nirmal *et al.*, 2021] Isura Nirmal, Abdelwahed Khamis, Mahbub Hassan, Wen Hu, and Xiaoqing Zhu. Deep learning for radio-based human sensing: Recent advances and future directions. *IEEE Communications Surveys & Tutorials*, 23(2):995–1019, 2021.

[Oord *et al.*, 2016] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[Rasul *et al.*, 2021] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, pages 8857–8868. PMLR, 2021.

[Rombach *et al.*, 2022] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[Shalaby *et al.*, 2022] Eman Shalaby, Nada ElShennawy, and Amany Sarhan. Utilizing deep learning models in csi-based human activity recognition. *Neural Computing and Applications*, 34(8):5993–6010, 2022.

[Tan *et al.*, 2022] Sheng Tan, Yili Ren, Jie Yang, and Yingying Chen. Commodity wifi sensing in ten years: Status, challenges, and opportunities. *IEEE Internet of Things Journal*, 9(18):17832–17843, 2022.

[Tashiro *et al.*, 2021] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34:24804–24816, 2021.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

[Wang *et al.*, 2019] Fei Wang, Jianwei Feng, Yinliang Zhao, Xiaobin Zhang, Shiyuan Zhang, and Jinsong Han. Joint activity recognition and indoor localization with wifi fingerprints. *IEEE Access*, 7:80058–80068, 2019.

[Wen *et al.*, 2021] Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4653–4660. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Survey Track.

[Yang *et al.*, 2018] Jianfei Yang, Han Zou, Hao Jiang, and Lihua Xie. Device-free occupant activity sensing using wifi-enabled iot devices for smart homes. *IEEE Internet of Things Journal*, 5(5):3991–4002, 2018.

[Yang *et al.*, 2022] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Yingxia Shao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *arXiv preprint arXiv:2209.00796*, 2022.

[Yoon *et al.*, 2019] Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32, 2019.

[Yousefi *et al.*, 2017] Siamak Yousefi, Hirokazu Narui, Sankalp Dayal, Stefano Ermon, and Shahrokh Valaee. A survey on behavior recognition using wifi channel state information. *IEEE Communications Magazine*, 55(10):98–104, 2017.

[Zeiler *et al.*, 2010] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. Deconvolutional networks. In *2010 IEEE Computer Society Conference on computer vision and pattern recognition*, pages 2528–2535. IEEE, 2010.

[Zhang *et al.*, 2021] Yi Zhang, Yue Zheng, Kun Qian, Guidong Zhang, Yunhao Liu, Chenshu Wu, and Zheng Yang. Widar3. 0: Zero-effort cross-domain gesture recognition with wi-fi. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.