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

Shuokang Huang¹, Po-Yu Chen¹,² and Julie McCann¹
¹Imperial College London
²JPMorgan 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].

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-
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.
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 \( x \in \mathbb{R}^{C \times N} \) with \( C \) channels, \( N \) denotes the time steps of its fixed window size, while \( \lambda_{\text{miss}} \) 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 \( \hat{x}_{\text{fc}} \in \mathbb{R}^{C \times N_{\text{fc}}} \) with \( x_{\text{fc}} = g_{\text{fc}}(x) \), where \( N_{\text{fc}} = \lambda_{\text{fc}}N \) is the future steps to forecast, and \( \lambda_{\text{fc}} \) represents the forecasting ratio. Subsequently, an imputation model \( g_{\text{im}}(\cdot) \) imputes the missing values in \( x \) by \( \hat{x}_{\text{im}} = g_{\text{im}}(x) \) to obtain \( \hat{x}_{\text{im}} \in \mathbb{R}^{C \times N} \) under the imputation ratio \( \lambda_{\text{im}} = \lambda_{\text{miss}} \). After forecasting and imputation, the augmented CSI is \( \hat{x} = x_{\text{fc}} + \hat{x}_{\text{im}} \), where \( \hat{x} \in \mathbb{R}^{C \times (1+\lambda_{\text{fc}})N} \). We formulate this self-supervised augmentation as \( \hat{x} = g(x) \).

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

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 \( p_\theta(x_0) \) to approximate a data distribution \( q(x_0) \) using two mutually inverse processes: the forward process and the reverse process. The forward process converts \( q(x_0) \) to a Gaussian distribution \( q(x_T) \) with a fixed \( T \)-step Markov chain, while the reverse process converts a Gaussian distribution \( p(\hat{x}_T) = N(\hat{x}_T; 0, 1) \) to \( p_\theta(x_0) \) with a learnable \( T \)-step Markov chain. The forward process is formulated as \( q(\hat{x}_1:T|x_0) \) with fixed Gaussian transitions \( q(x_t|x_{t-1}) \) for \( t = 1, \ldots, T \). Conversely, the reverse process is formulated as \( p_\theta(x_{0:T}) \) with learnable Gaussian transitions \( p_\theta(\hat{x}_{t-1}|\hat{x}_t) \) for \( t = T, \ldots, 1 \). Applying these formulations in practice, diffusion models optimize a denoising function \( \epsilon_\theta(\cdot) \) to synthesize \( x_0 \) by iterating \( t = [T, \ldots, 1] \). 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 \( \hat{x} = x_0 \) from Gaussian noise \( \hat{x}_T \in \mathbb{R}^{C \times (1+\lambda_{\text{fc}})N} \) by \( T \)-step progressive synthesis conditioned on CSI spectrogram \( S \). 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 \( \epsilon_\theta(\cdot) \), ACDM takes \( x_t \) as inputs and uses a \( 5 \times 5 \) convolution followed by an ReLU activation to extract both temporal and channel-wise features. To incorporate the step information into \( \epsilon_\theta(\cdot) \), ACDM performs step encoding and linear projections on each step \( t \) to obtain the step embedding \( \hat{t} \). 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 \( S \), 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 as

![Figure 3: Overview of the proposed DiffAR.](image-url)
Linear $\tanh$ Concatenate

Original diffusion models are unconditional [Ho et al., 2020] and cannot be directly leveraged for CSI augmentation. 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:

$$c_t = \psi(S, t) = \nu(S) \odot (\omega t + b),$$

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

Intuitively, for different steps, $\varphi(\omega t + b)$ acts as a step-specific filter to extract critical information from input condition features $\nu(S)$. Hence, $c_t$ represents the critical condition information for different steps. ACDM feeds $c_t$ to every residual block, so the adaptive conditioner can be jointly optimized with $e_\theta(\cdot)$. 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 $\mathbf{x}_t$, after which the linear projection of $\mathbf{t}$ is added as a bias term. To guide the progressive steps in ACDM, the projection of step-specific conditions $c_t$ 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 ($\varphi$) to learn the nonlinear features.

**Step Embedding**

To synthesize augmented CSI by progressive steps, it is necessary to take steps as inputs to estimate $e_\theta(\cdot)$. 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 $\mathbf{t}$. 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 $t^c \in \mathbb{R}^M$:

$$t^c = \left[\sin\left(10\pi^{b_1}t^*\right), \ldots, \cos\left(10\pi^{b_1}t^*\right), \ldots, \right],$$

for $m \in [0, \ldots, (M/2 - 1)]$. We further adopt two linear projection layers to compute $\hat{t} = (\omega^0(\omega^1t^c + b^0) + b^1$ as the step embedding, where $\omega^0$ and $\omega^1$ are the weights of two layers, and $b^0$ and $b^1$ are the biases of two layers. We formulate this step embedding as $\hat{t} = \text{embed}(t)$, which is further fed to the adaptive conditioner and every residual block.
After training ACDM, we can exploit it to synthesize augmented CSI samples and calculate missing values under CSI, so taking raw CSI as inputs can ensure no information is lost. 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.

3.3 Ensemble Classifier

After augmentation, an ensemble classifier in DiffAR employs 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 1 Training

repeat
1: $\hat{x}_0 \sim q(x)$ # regard raw CSI as augmented CSI
2: $S' = \text{stft}(x')$ where $x' = \text{mask}(x_0)$
3: $t = \text{embed}(t)$ where $t \sim \text{Uniform}\{1, \ldots, T\}$
4: $c_t' = \psi(S', t)$ # apply the adaptive conditioner
5: $\epsilon \sim \mathcal{N}(0, I)$
6: Take gradient step on
7: $\nabla_\theta \|\epsilon - \epsilon_0 (\sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}\epsilon, t, c_t')\|^2$
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:

$$
p_0(x_{0:T}|S) := p(\hat{x}_T) \prod_{t=1}^T p_0(\hat{x}_{t-1}|\hat{x}_t, c_t),$$

(5)

$$
p_0(\hat{x}_{t-1}|\hat{x}_t, c_t) = p_0(\hat{x}_{t-1}|\hat{x}_t, \psi(S, t)).$$

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

$$
L^a(\theta) := E \left[ \|\epsilon - \epsilon_0 (\sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}\epsilon, t, c_t')\|^2 \right].
$$

(6)

Training. We train ACDM in a self-supervised manner, where we mask certain values of raw CSI to simulate incomplete CSI $x'$ and regard raw CSI as the augmented CSI $\hat{x}_0$. With $x' = \text{mask}(x_0)$, we use random masks to simulate missing values under $\lambda_{\text{im}}$ and mask the rear part of CSI to simulate the forecasting targets under $\lambda_\epsilon$. We perform short-time Fourier transform (STFT) on $x'$ to calculate its spectrogram $S' = \text{stft}(x')$. Since $x_0$ acts as ground-truth targets, we can train ACDM conditioned on $c_t' = \psi(S', t)$ by $\min_\theta L^a(\theta)$, as illustrated in Algorithm 1.

Synthesis. After training ACDM, we can exploit it to synthesize augmented CSI samples $\hat{x}_0$ based on incomplete CSI $x \in \mathbb{R}^{C \times N}$. We again perform STFT to obtain the spectrogram $S = \text{stft}(x)$ and sample $x_T \in \mathbb{R}^{C \times (1+\lambda_\epsilon)N}$ from Gaussian noise for synthesis. For each step $t$ in $[T, \ldots, 1]$, ACDM computes its step-specific condition $c_t = \psi(S, t)$ to guide the synthesis $p_0(\hat{x}_{t-1}|\hat{x}_t, c_t)$, as illustrated in Algorithm 2.

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...
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). Lower results indicate better quality. **Bold** highlights the best results.

<table>
<thead>
<tr>
<th>Models</th>
<th>Ratio</th>
<th>( \lambda_\text{fc} )</th>
<th>( \lambda_\text{im} )</th>
<th>Office</th>
<th>SignFi</th>
<th>Interactions</th>
<th>Widar 3.0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MAE</td>
<td>MSE</td>
<td>CRPS</td>
<td>MAE</td>
</tr>
<tr>
<td><strong>Forecast</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TimeGrad</td>
<td>0.2</td>
<td>0.0</td>
<td></td>
<td>1.074</td>
<td>1.886</td>
<td>1.321</td>
<td>0.938</td>
</tr>
<tr>
<td>CSDI</td>
<td>0.2</td>
<td>0.0</td>
<td></td>
<td>1.084</td>
<td>1.764</td>
<td>1.363</td>
<td>0.856</td>
</tr>
<tr>
<td>WaveGrad</td>
<td>0.2</td>
<td>0.0</td>
<td></td>
<td>0.862</td>
<td>1.181</td>
<td>1.084</td>
<td>0.730</td>
</tr>
<tr>
<td>DiffWave</td>
<td>0.2</td>
<td>0.0</td>
<td></td>
<td>0.848</td>
<td>1.141</td>
<td>1.066</td>
<td>0.779</td>
</tr>
<tr>
<td>DiffAR (Ours)</td>
<td>0.2</td>
<td>0.0</td>
<td></td>
<td>0.819</td>
<td>1.071</td>
<td>1.029</td>
<td>0.721</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>**0.811</td>
<td>1.058</td>
<td><strong>1.032</strong></td>
<td><strong>0.811</strong></td>
</tr>
<tr>
<td><strong>Imputation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSDI</td>
<td>0.0</td>
<td>0.2</td>
<td></td>
<td>1.064</td>
<td>1.708</td>
<td>1.338</td>
<td>0.856</td>
</tr>
<tr>
<td>WaveGrad</td>
<td>0.0</td>
<td>0.2</td>
<td></td>
<td>0.868</td>
<td>1.215</td>
<td>1.091</td>
<td>0.730</td>
</tr>
<tr>
<td>DiffWave</td>
<td>0.0</td>
<td>0.2</td>
<td></td>
<td>0.855</td>
<td>1.166</td>
<td>1.074</td>
<td>0.784</td>
</tr>
<tr>
<td>DiffAR (Ours)</td>
<td>0.0</td>
<td>0.2</td>
<td></td>
<td><strong>0.811</strong></td>
<td>1.117</td>
<td><strong>1.019</strong></td>
<td><strong>0.718</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>0.808</strong></td>
<td>1.066</td>
<td><strong>1.028</strong></td>
<td><strong>0.827</strong></td>
</tr>
</tbody>
</table>

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.

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 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 split into a training set (80%), a validation set (10%), and a test set (10%). We leverage training sets to optimize ACDM for 100 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.
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 score. 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 exceeds 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.
References


