SD^2: Synthetic Doppler Spectrum Denoiser using SSM

Koushik A. Manjunatha, Morris Hsu, and Rohit Kumar

Amazon Lab126, Sunnyvale, CA, USA, 94089,
{koushiam,rrohk}@amazon.com, mhsu@lab126.com

Abstract. The increasing popularity of wireless sensing applications has led to a growing demand for large datasets of realistic wireless data. However, collecting such wireless data is often time-consuming and expensive. To address this challenge, we propose a synthetic data generation pipeline using human mesh generated from videos that can generate data at scale.
The pipeline first generates a 3D mesh of the human in the video and then determines the initial synthetic Doppler of the human motion. The initial synthetic Doppler will be noisy due to the uncertainties involved during the mesh generation. To address this, we employ a trained Structured State Space for Sequence Modeling (S4) model to denoise the initial synthetic Doppler to match the Doppler signature from the radar device. We validated our pipeline on synthetic Doppler data from four hand gesture videos and found that the final synthetic Doppler closely resembles the real Doppler, outperforming existing U-Net-based models by 36% or more. Also, with smaller set of denoised synthetic Doppler, the gesture classification model performance increased from 89.5% to 92.7%.

Keywords: Video, Mesh, mmWave Radar, Wireless, Artificial Intelligence

1 Introduction

Radio frequency (RF) sensing is a promising approach that has seen a lot of development in past decades. These sensors offer signal richness comparable to that of microphones and cameras but without any privacy constraints. The mmWave radar is one of those example that has seen a lot of development over the past years. These sensors are used for presence sensing, gesture recognition, and in various smart home applications. However, the development of AI/ML models for RF sensing as compared to development of signal processing algorithms has always lagged behind due to lack of data. Comparing the datasets in computer vision or language to the available mmwave radar datasets, the radar data is very small and limited in scope. In addition, these radar datasets are typically focused on a specific problem that the authors are trying to solve, and thereby are not large enough to be used to train deep learning models.

In order to advance the development of AI/ML models for RF sensing, we need to create larger and more diverse datasets. These datasets should be representative of the real-world environments in which RF sensors will be used.

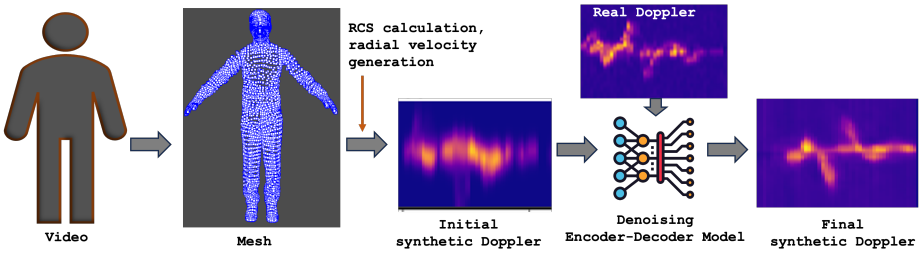


Fig. 1: End-to-end V2W pipeline to generate synthetic wireless signal from videos.

Thus, in this paper we propose an AI/ML based software pipeline that can generate wireless data (in this work we have considered FMCW radar data as an example) from vast available videos, same on the lines of [1]. The difference being the use case of gesture, evaluation of different mesh generation algorithms, and denoising of the Doppler feature using Structured State Space for Sequence Modeling (S4).

The rest of the paper is organized as follows: Section 2 outlines the synthetic wireless signal generation in detail, Section 4 presents the results and discussion, and finally, the conclusions are drawn in Section 6.

2 Synthetic Wireless Signal Generation

Although this method can be used to generate wireless signals for any technology, such as Bluetooth, WiFi, and cellular, in this work, we will consider synthetic radar Doppler signal generation at a millimeter-wave frequency of 60 GHz. To have a comparison between real wireless signal and synthetically generated wireless signal, we collect videos and radar signals simultaneously with 6 subjects. Each subject is asked to perform 4 hand gestures: i) left-to-right, ii) right-to-left, iii) up-to-down, and iv) down-to-up for 15 seconds, with the radar and video camera set at the same frame rate of 30 frames per second (FPS). The complete pipeline is shown in Fig. 1.

2.1 Mesh Fitting

The first block in this pipeline is the mesh generation of the subject in a video scene. Mesh generation is a 3D representation of the subject against a static background. In this block, the position of all vertices of the human body in the video is estimated by an AI/ML mesh model. This AI/ML model fits a mesh to

the subject by estimating the human poses and outputs a human pose mesh in each video frame.

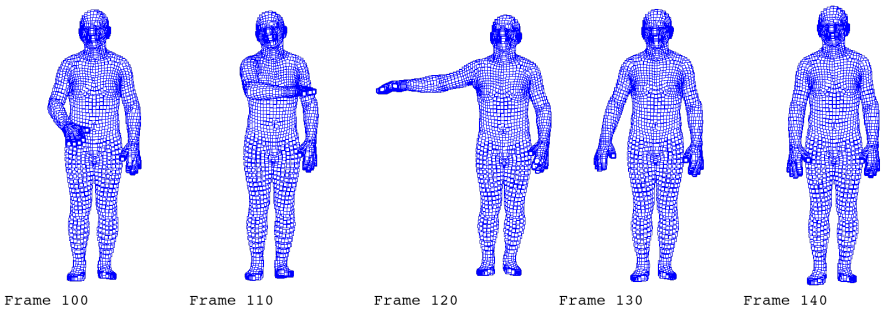


Fig. 2: Mesh generated at different video frames for a horizontal gesture. The frames shown are 100, 110, 120, 130, and 140, and the blue dots in the mesh represent vertices with (x,y,z) locations. These points will be tracked in every frame to calculate Doppler values of the gesture.

2.2 RCS and Radial Velocity Generation

To transfer the computer vision modality to the wireless domain from the meshes generated in each frame of the video, this block will generate a radar cross-section and create a radial velocity profile. To achieve that, the movement of each mesh point with respect to the virtual Doppler sensor is considered. Only the non-occluded mesh points are considered in this process. In the process, the frame rate of the video is adjusted to produce more realistic and smoother variations, which get translated to a radial velocity profile.

2.3 Initial Synthetic Doppler Generation

For each video frame, the determined velocity values of each mesh vertex are mapped to 32 Doppler bins. The counts in bins 14, 15, and 16 are nullified as they have approximately zero velocity. This step is essentially in removing any bias that will be created from the static vertices of the body mesh. Finally, the histogram is divided by the total number of vertices in the human mesh to get the probability mass function (pmf) value of each bin. These values are calculated for each frame, and the initial velocity vs. Frame heatmap is generated. The initial velocity vs. Frame heatmap is passed through a Gaussian smoothing filter

to generate the initial synthetic Doppler heatmap. The entire synthetic Doppler heatmap was divided into multiple segments of size 32×72 with an overlapping factor of 10 frames. A similar ground truth Doppler images were generated from the real FMCW radar at *30 FPS* to use during training the denoiser model.

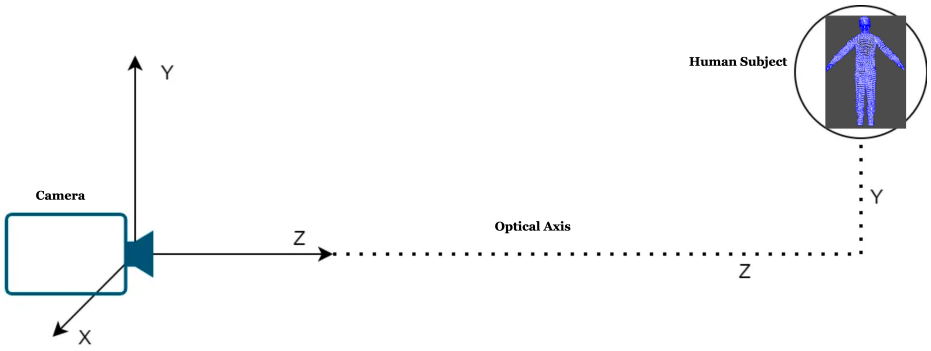


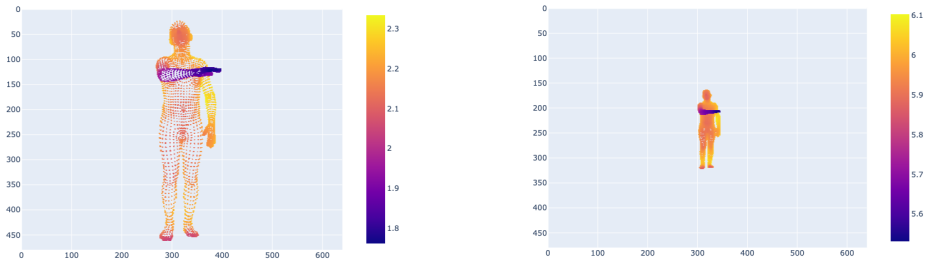
Fig. 3: Camera axis description.

2.4 Doppler Data Augmentation with Range Synthesis

The Z-axis is the optical axis of the camera (see Fig 3) which is converted to world coordinates in the VIBE model through weak perspective camera model. The euclidean distance between an assumed camera position (in world coordinates) and each mesh points (in world coordinates) provide the range for each mesh points. The demonstration of range estimation for each mesh point shown using pyvista toolbox. The rendered depth plot of same image at distances of $2.2m$ and $6m$ from camera position are shown in Fig. 4a and 4b, respectively. The intensity values shown in color bar (on the right hand side) indicate the distance from camera position to the respective vertex. The estimated range values from different camera position is used to adjust the intensity of the Doppler signal and added to the synthetic dataset for training.

2.5 Denoising Encoder-Decoder

The initial synthetic Doppler data is at a very coarse level and has noise because of non-smooth mesh point movements from frame to frame. Considering higher radar configurations such as the number of chirps and frame rate, the dimension of the initial synthetic Doppler can be large. Also, synthetic Doppler has a time-dependency between historical frames and are very long and implicitly



(a) Range at $2m$ (b) Range at $6m$

Fig. 4: Range synthesis of generated mesh from various virtual sensor locations.

continuous. Input data with long sequences and historical dependency is more expensive for transformer-based models and achieve sub-optimal performances. This made us look beyond attention-based approaches and thus consider a state-space-based model.

State Space Models A state space model contains a minimum number of variables that fully describe a system state and make predictions of what their next could be, depending on some input. The S4 [4] is built based on the linear time-invariant system. The principle is that simple continuous state space Models (SSMs) are defined by two equations—one capturing the change over time to a hidden state $x(t)$ and another capturing the relationship between the hidden state, an input $u(t)$ and an output $y(t)$.

$$\begin{aligned} x'(t) &= \boldsymbol{A}x(t) + \boldsymbol{B}u(t) \\ y(t) &= \boldsymbol{C}x(t) + \boldsymbol{D}u(t) \end{aligned} \tag{1}$$

Here, A is called the state matrix, and B, C, and D are system parameters. For the long sequence models, the idea is to consider an SSM as a function-to-function map parameterized by parameters A,B,C,D which are simply learned by gradient descent.

Finding the hidden state representation, $x(t)$ of a continuous signal is challenging, and in the digital domain, the signals are discrete (like an image, textual sequences). To accommodate discrete-time sequences sampled with a step size Δ, a learnable parameter (based on *Zero-order hold technique*), the equation (1) is modified as

$$\begin{aligned} x'_k &= \boldsymbol{A}x_k + \boldsymbol{B}u_k \\ y_k &= \boldsymbol{C}x_k + \boldsymbol{D}u_k \end{aligned} \tag{2}$$

The SSMs can be computed with the recurrence as

$$\begin{aligned} x_k &= \bar{\boldsymbol{A}} x_{k-1} + \bar{\boldsymbol{B}} x_k \\ y_k &= \bar{\boldsymbol{C}} x_k + \bar{\boldsymbol{D}} x_k \end{aligned} \tag{3}$$

Here $\bar{A}, \bar{B}, \bar{C}, \bar{D}$ are the parameters of the recurrent model which have simple closed formulas in terms of the base parameters A,B,C,D.

$$\begin{aligned} \bar{\boldsymbol{A}} &= (\boldsymbol{I} - \frac{\Delta}{2} \cdot \boldsymbol{A})^{-1} (\boldsymbol{I} + \frac{\Delta}{2} \cdot \boldsymbol{A}) \\ \bar{\boldsymbol{B}} &= (\boldsymbol{I} - \frac{\Delta}{2} \cdot \boldsymbol{A})^{-1} \Delta \boldsymbol{B} \end{aligned} \tag{4}$$

The linear recurrences can be explicitly computed in parallel as a convolution with an SSM kernel $\bar{\boldsymbol{K}}$ to reveal a closed formula for the output y in terms of the input u.

$$\bar{\boldsymbol{K}} = (\bar{C}\bar{A}^i\bar{B})_{i\in[L]} = (\bar{C}\bar{B}, \bar{C}\bar{A}\bar{B}, ..., \bar{C}\bar{A}^{L-1}\bar{B}), \quad y = \bar{\boldsymbol{K}} * u \tag{5}$$

This means that the entire output of the SSM is simply the (non-circular) convolution of the input u with the convolution filter $\bar{\boldsymbol{K}}$.

S4 S4 is a particular instantiation of SSM that parameterizes $\bar{A}$ as a diagonal plus low-rank (DPLR) matrix, $\bar{A} = \Lambda + pq^*$. This parameterization has two key properties. First, this is a structured representation that allows faster computation—S4 uses a special algorithm to compute the convolution kernel K (5) very quickly. Second, this parameterization includes special matrices called HiPPO matrices [3], which theoretically and empirically allow the SSM to overcome vanishing/exploding gradients problems and allow the state x to memorize the history of the input u. In particular, HiPPO specifies a special equation $x'(t) = \boldsymbol{A}x(t) + \boldsymbol{B}u(t)$ with closed formulas for $\boldsymbol{A}$ and $\boldsymbol{B}$. The HiPPo matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ is given by

$$(\textbf{HiPPO Matrix}) \quad \boldsymbol{A}_{nk} = \begin{cases} (2n+1)^{1/2}(2k+1)^{1/2}, & n > k \\ n+1, & n = k \\ 0, & n < k \end{cases} \tag{6}$$

The input to the S4 model is the synthetic Doppler signal generated from a video, and the output is the real Doppler signal from the radar during training. This forces the S4 to learn to remove noise from the synthetic Doppler signal. The structure of the S4 encoder-decoder-based denoiser model is shown in Fig 5.

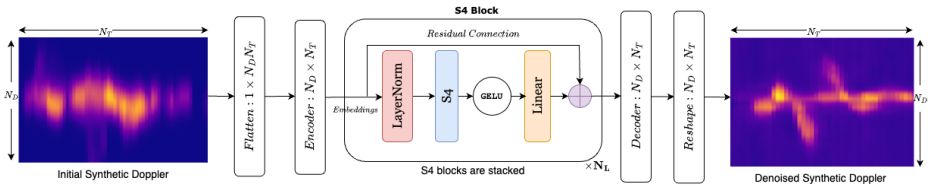


Fig. 5: S4 encoder-decoder-based synthetic Doppler denoiser model.

3 Doppler Estimation from FMCW Radar

Frequency Modulated Continuous Wave (FMCW) [5] radar is a type of radar system that utilizes a continuous wave signal with a linearly varying frequency to measure the range and velocity of targets .

3.1 Frequency Modulation

A key feature of FMCW radar is the frequency modulation applied to the transmitted signal. In this approach, the radar transmits a continuous wave signal with a frequency that increases linearly over time. This frequency modulation is achieved by applying a saw-tooth or triangular waveform to the frequency of the transmitted signal.

The rate of change of the frequency, known as the frequency sweep rate, is determined by the frequency bandwidth B and the sweep period T. The transmitted signal in FMCW radar can be expressed mathematically as:

$$f(t) = f_0 + \frac{B}{T}t \tag{7}$$

where $f(t)$ is the instantaneous frequency of the transmitted signal, and f_0 is the starting frequency. The frequency sweep bandwidth B and the sweep period T are key design parameters that influence the performance characteristics of the FMCW radar system.

3.2 Range and Doppler

The fundamental principle behind FMCW radar is the measurement of the frequency difference between the transmitted and received signals. When the transmitted signal reaches a target and is reflected back, the returned signal experiences a time delay that is directly related to the distance separating the radar and the target. This time delay ultimately gives rise to a frequency difference

between the transmitted and received signals, a difference referred to as the beat frequency.

FMCW radar systems possess the ability to simultaneously assess the range and velocity of a target by analyzing the beat frequency. The range is calculated as a function of the time delay τ between the transmitted and received signals, and the target's radial velocity v_r is inferred from the Doppler shift f_d observed in the received signal. The range R to a target is determined using the following equation:

$$R = \frac{c\tau}{2} \tag{8}$$

where c is the speed of light. The Doppler shift f_d experienced by the reflected signal due to the relative motion of the target is given by:

$$f_d = \frac{2v_r}{\lambda} \tag{9}$$

where λ is the wavelength of the transmitted signal. The radial velocity v_r can then be calculated from the Doppler shift f_d using the following equation:

$$v_r = \frac{f_d\lambda}{2} \tag{10}$$

By combining the range and Doppler information, FMCW radar systems can provide a comprehensive understanding of the target's position and motion, enabling a wide range of applications in areas such as transportation, security, and environmental monitoring.

4 Results and Discussion

In this section, we will first evaluate the performance of the denoising S4 model with real radar Doppler signal.

4.1 Comparison of Synthetic Doppler with Real Doppler

The initial synthetic Doppler generated by the pipeline is very noisy due to errors in mesh vertex estimation and changes in the mesh vertices over time that may not follow the real pattern. These errors can induce artificial Doppler artifacts in the output. For training, we will use 1.5 hours of video data containing gestures to generate the synthetic Doppler and use the real Doppler signal from the radar as an output. The real Doppler is extracted from the FMCW radar operating at 60GHz, with 64 chirps per frame and 32 samples per chirp. The video and the radar data were captured synchronously in a set up.

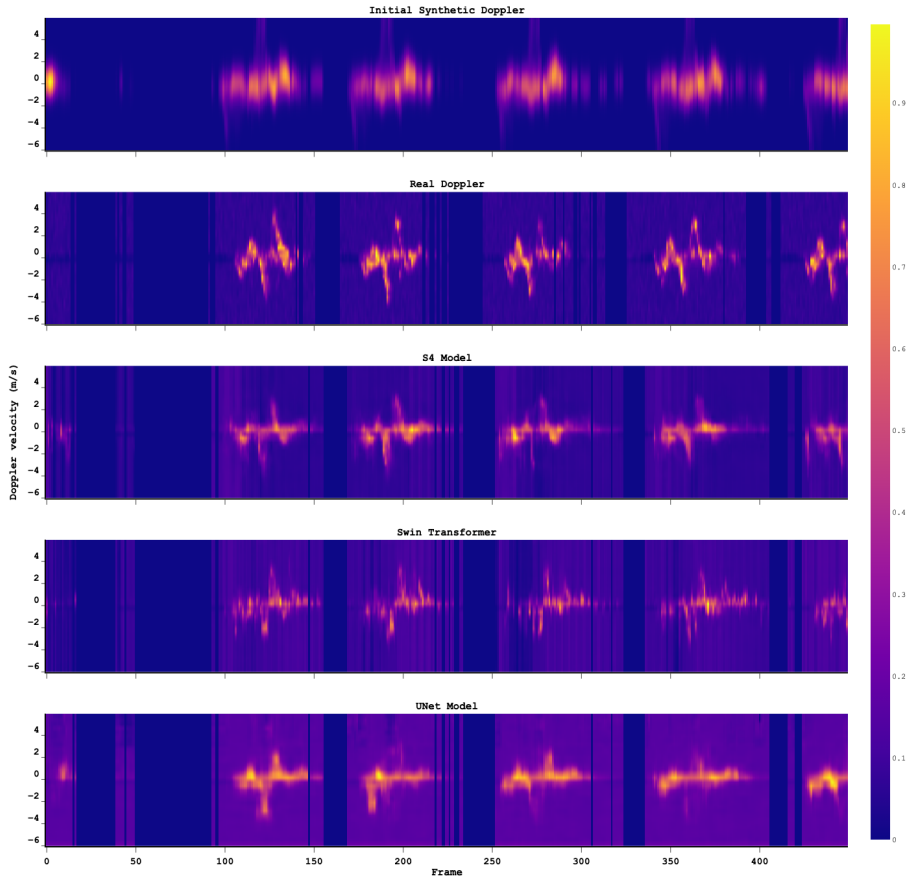


Fig. 6: Comparison of denoised synthetic Doppler generated from different models and the real Doppler from the FMCW radar.

Each subject was asked to perform all four gestures in random order, with radar data being collected in parallel with the video. The coarse synthetic Doppler were serialized and fed into a linear block to generate an embedding of 128, followed by 4 layers of S4 filter, followed by a decoding layer to generate the denoised synthetic Doppler. The model was trained at a batch size of 64 for 500 epochs using root mean squared error as the error metric. The generated denoised outputs were also compared with other models, including U-Net, the training method mentioned in [1], and using the Swin transformer [2]. Comparing the outputs of each model, the S4 model shows the better denoising performance, and

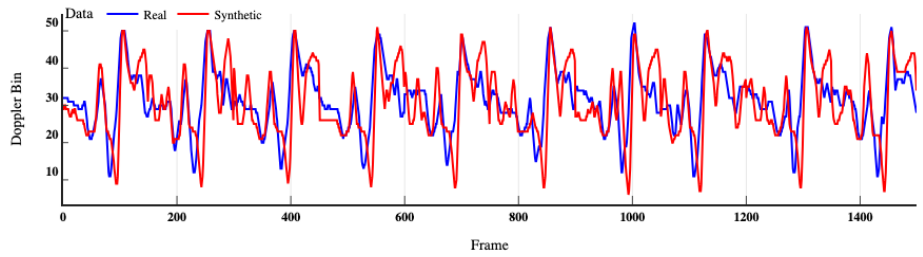
they are compared using the metrics mean absolute error (MAE) and standard deviation (σ). The model performances are shown in 1.

Model	MAE (m/s)	σ (m/s)
S4 [4]	1.0126	1.6329
Swin Transformer [2]	1.1640	1.8734
U-Net [1]	1.5835	2.6549

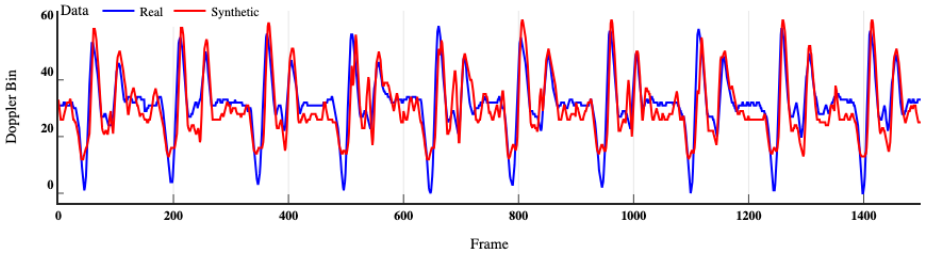
Table 1: Comparison of denoiser models' performances in terms of MAE and σ.

5 Activity Classification

We developed an AI model to classify between *horizontal-swipe* and *vertical-swipe* using Doppler signature. We develop a classifier model with 3 classes, *horizontal-swipe*, *vertical-swipe*, and *background*. The Doppler bin with a maximum intensity in a frame is extracted as an input feature.



(a) Horizontal Swipe



(b) Vertical Swipe

Fig. 7: Doppler spectrum plots for horizontal and vertical swipe gestures.

5.1 Model Architecture

The model consists of a sequential architecture with two LSTM (Long Short-Term Memory) layers, followed by layer normalization, a dense layer, dropout regularization, and a final dense layer with a softmax activation function for multi-class classification. The LSTM layers are designed to be stateful with 16 states, allowing the model to maintain its internal state between batches, which can help it capture temporal dependencies in the input data.

5.2 Model Performance

The model was initially trained using real-radar samples. Each class has $\approx 30K$ samples from real Doppler while each class has $\approx 12K$ samples from synthetic Doppler.

$\frac{Predicted}{True}$	Horizontal Swipe	Vertical Swipe	Background
Horizontal Swipe	87.3%	5.3%	7.4%
Vertical Swipe	6.8%	89.1%	4.1%
Background	3.8%	4.1%	92.1%

Table 2: Confusion matrix for the gesture recognition model (data trained with real Doppler).

$\frac{Predicted}{True}$	Horizontal Swipe	Vertical Swipe	Background
Horizontal Swipe	91.6%	4.0%	4.5%
Vertical Swipe	4.1%	92.2%	3.7%
Background	2.5%	3.2%	94.3%

Table 3: Confusion matrix for the gesture recognition model after training with synthetic Doppler plus real Doppler.

The gesture recognition model was evaluated using two different training approaches. The first confusion matrix, shown in Table 2, presents the model's performance when trained solely on real Doppler data. The diagonal entries in the confusion matrix indicate the classification accuracy for each gesture class, with the horizontal swipe achieving 87.3%, the vertical swipe 89.1%, and the

background class 92.1%. The off-diagonal entries represent the misclassification rates between the different classes.

To improve the model's performance, a second approach was explored, which incorporated both real and synthetic Doppler data during training. The confusion matrix in Table 3 shows the results of this approach. The classification accuracy for the horizontal swipe, vertical swipe, and background classes increased to 91.6%, 92.2%, and 94.3%, respectively, indicating that the addition of synthetic Doppler data helped the model generalize better and achieve higher overall performance.

These confusion matrices provide a comprehensive evaluation of the gesture recognition model's classification capabilities, highlighting the improvement in the model's performance with the synthetic data.

6 Conclusion

In this work, we have presented a novel pipeline for generating synthetic wireless signal data, specifically focusing on the generation of synthetic Doppler signatures from videos. The motivation behind this approach is the growing demand for large and diverse datasets in the field of wireless sensing and AI/ML applications, which are often hindered by the challenges of collecting real-world data.

The proposed pipeline consists of several key components. First, it generates a 3D mesh representation of the human subject in the input video, tracking the movement of the mesh vertices over time. This mesh information is then used to calculate the initial synthetic Doppler signatures, which, as expected, are quite noisy due to the uncertainties in the mesh generation process.

To address this issue, we employ a state-of-the-art Structured State Space for Sequence Modeling (S4) model to denoise the initial synthetic Doppler data and make it closely resemble the real Doppler signatures obtained from a 60GHz FMCW radar. Our experiments show that the S4-based denoiser model outperforms other approaches, such as U-Net and Swin Transformer, by a significant margin, achieving up to 36% lower mean absolute error and standard deviation.

Furthermore, we demonstrate the utility of the denoised synthetic Doppler data by incorporating it into a gesture recognition model. The results show that the inclusion of synthetic data, in addition to the real Doppler samples, can improve the classification performance from 89.5% to 92.7%, highlighting the value of the synthetic data in enhancing the generalization capabilities of the model.

Going forward, we plan to expand the pipeline to generate synthetic data for other wireless modalities, such as IMU and WiFi, and explore the integration

of this approach with various wireless sensing applications, including presence detection, activity recognition, and localization. By continuously advancing the capabilities of synthetic data generation, we aim to empower the research community to develop more robust and versatile AI/ML models for wireless sensing, ultimately driving the progress of this important field.

References

1. Karan Ahuja, Yue Jiang, Mayank Goel, and Chris Harrison. Vid2Doppler: Synthesizing Doppler Radar Data from Videos for Training Privacy-Preserving Activity Recognition. CHI '21, New York, NY, USA, 2021. Association for Computing Machinery.
2. Marcos V Conde, Ui-Jin Choi, Maxime Burchi, and Radu Timofte. Swin2SR: Swinv2 Transformer for Compressed Image Super-Resolution and Restoration. In *European Conference on Computer Vision*, pages 669–687. Springer, 2022.
3. Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33:1474–1487, 2020.
4. Albert Gu, Karan Goel, and Christopher Ré. Efficiently Modeling Long Sequences with Structured State Spaces. *arXiv preprint arXiv:2111.00396*, 2021.
5. Fan Liu and Christos Masouros. A tutorial on joint radar and communication transmission for vehicular networks—part ii: State of the art and challenges ahead. *IEEE Communications Letters*, 25(2):327–331, 2020.