IEEE Open Journal of Vehicular Technology

Received 9 July 2024; accepted 26 July 2024. Date of publication 2 August 2024;
date of current version 26 August 2024. The review of this article was coordinated by Editor Chong Han.

Digital Object Identifier 10.1109/OJVT.2024.3436857

Measurement-Based Prediction of mmWave Channel Parameters Using Deep Learning and Point Cloud

HANG MI [1,2], BO AI [1] (Fellow, IEEE), RUISI HE [1] (Senior Member, IEEE), ANURAAG BODI [3], RAIED CAROMI [4], JIAN WANG [4] (Member, IEEE), JELENA SENIC [4], CAMILLO GENTILE [4] (Member, IEEE), AND YANG MIAO [2] (Senior Member, IEEE)

[1]State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China
[2]EEMCS faculty, University of Twente, 7522 NB Enschede, The Netherlands
[3]Associate, National Institute of Standards and Technology, Prometheus Computing LLC, Gaithersburg, MD 20899 USA
[4]Wireless Networks Division, National Institute of Standards and Technology, Gaithersburg, MD 20899 USA

CORRESPONDING AUTHOR: YANG MIAO (e-mail: y.miao@utwente.nl).

ABSTRACT Millimeter-wave (MmWave) channel characteristics are quite different from sub-6 GHz frequency bands. The major differences include higher path loss and sparser multipath components (MPCs), resulting in more significant time-varying characteristics in mmWave channels. It is difficult to characterize the time-varying characteristics of mmWave channels through statistical models, e.g. slope-intercept models for path loss and lognormal models for delay spread and angular spread. Therefore, highly accurate channel modeling and prediction are necessary for deployment of mmWave communication systems. In this paper, a mmWave channel parameter prediction method using deep learning and environment point cloud is proposed. The parameters predicted include path loss, root-mean-square (RMS) delay spread, angular spread and Rician K factor. First, we introduce a novel measurement campaign for indoor mmWave channel at 60 GHz, where a light detection and ranging (LiDAR) sensor and panoramic camera were co-located with a channel sounder and then time-synchronized point clouds and images were captured to describe environmental information. Furthermore, a fusion method between the point clouds and images is proposed based on geometric relationship between the LiDAR and camera, to compress the size of the data collected. Second, based on a classic point cloud classification model (PointNet), we propose a novel regression PointNet model applied to channel parameter prediction. To overcome generalization problem of model under limited measurements, an area-by-area training and testing method is proposed. Third, we evaluate the proposed prediction model and training method, by comparing prediction results with measured ground truth. To provide insights on what training inputs are best, we demonstrate the impacts of different combinations of input information on prediction accuracy. Last, the deployment and implementation method of the proposed model is recommended to the readers.

INDEX TERMS Channel measurement, channel prediction, deep learning, mmWave channel, point cloud.

I. INTRODUCTION

Millimeter-wave (mmWave) communication technology is one of the key enablers of fifth-generation mobile networks (5G) to enable high-rate and low-latency communications [1]. All the while, mmWave is expected to continue to play a critical role in beyond 5G (B5G) and 6G communication systems. Namely, B5G or 6G technologies such as ultra massive multiple-input-multiple-output (MIMO) beamforming, ultra-dense networks and joint communications and sensing (JCAS) have shown greater potential in the mmWave frequency band [2], [3], [4]. In order to better enable these future technologies, accurate mmWave channel modeling and prediction in desired scenarios will be more essential. However, the special characteristics of mmWave channels bring great

challenges to the design and deployment of mmWave systems [5]. For example, free space, penetration and diffraction losses increase significantly at mmWave frequencies. This enhances the sparsity of multipath components (MPCs) in some scenarios [6], [7], [8]. Although beamforming with ultra massive MIMO can offer high-gain beams to compensate for the attenuation of mmWave, these techniques necessitate beam acquisition and tracking [9]. This further increases the requirement for exploiting the information of mmWave channels, especially for the description of time-varying characteristics. Furthermore, in the emerging JCAS technology, the MPCs information in the sensing channel needs to be extracted and modeled to capture the changes in the surrounding environment [10]. These put forward more stringent demands on the accuracy of channel prediction.

Many channel modeling and prediction methods have been developed in the literature, which can be classified into four main categories: statistical modeling, deterministic modeling, geometry-based stochastic modeling and machine learning-based modeling.

i) *Statistical modeling* is usually based on channel measurements and is fitted to the measured results using specific equations or models. For example, [11] shows the alpha-beta-gamma (ABG) model and close-in (CI) model, which are widely used in the modeling and prediction of path loss. These models are based on the received power and distance between transceivers, and fit the parameters in the model and provide efficient predictions. However, the slope-intercept models such as ABG and CI are limited to linear progression of path loss with the change of distance, while in highly non-stationary scenarios, the large scale parameters might change abruptly along the distance. For other large-scale parameters (LSPs), such as root-mean-square (RMS) delay spread, angular spread and Rician K factor, the lognormal distribution is usually used [12], [13]. Among them, [13] uses two sets of lognormal distribution parameters to fit line-of-sight (LOS) and non-line-of-sight (NLOS) scenarios, respectively. However, the fitting parameters of the model are usually suitable for channel parameter prediction in some specific scenarios. In the face of diverse application scenarios for B5G and 6G communication, the limitations of statistical channel models will be more significant.

ii) *Deterministic modeling* is a physical simulation model based on the principle of radio wave propagation, which is mainly represented by ray tracing (RT). The RT is based on geometric optics and is able to compute the direction, reflections, scattering and diffraction of MPCs [14], [15]. RT could also be combined with empirical models to compute diffuse scattering components [15], [16]. Channel parameters can be obtained from simulated MPCs. Meanwhile, RT can support the simulation of dynamic channels to explore time-varying characteristics [17]. However, the accuracy of RT simulation is too dependent on the geometric and material information of the simulated environment. Although some research has focused on the calibration of RT [18], it is still difficult to generalize the calibrated RT to another environment. In addition, RT has high computational complexity, especially in electrically large-scale scenarios. Although some acceleration methods have been applied in RT [19], [20], it is still difficult to achieve real-time channel prediction.

iii) The *geometry-based stochastic models* assume that the scatterers in the channel are randomly distributed on the surface of regular geometry [21], [22], [23]. The MPCs parameters and channel impulse response can be derived according to the geometrical relationship. Stochastic models are not based on measurements and lack description of the real environment (described by regular geometry). Therefore, it only provides a theoretical channel in some scenarios and cannot be used for channel prediction.

iv) Since *machine learning* has good performance in solving nonlinear or unstructured problems, it is especially suitable for predicting stochastic and time-varying channel characteristics. In recent years, a lot of research with focus on machine learning-based channel prediction and classification has emerged. On the one hand, some research utilizes channel data to predict channel or environment characteristics based on machine learning, which is called "channel-based feature identification" [24]. For example, [25] identifies LOS/NLOS conditions based on channel parameters including RMS delay spread, Rician K factor and angular characteristics. Similarly, these channel parameters can also be used to classify and identify scenarios such as urban, tunnels and highways [26].

On the other hand, environmental information is exploited and extracted using machine learning-based methods, and the extracted environmental features are used to predict channel parameters. This work is called "channel parameters prediction" [27]. Among them, path loss prediction based on images or map is represented [28], [29], [30].

In this paper, we focus on channel parameter prediction based on deep learning and environment data described by point clouds. In the process, the prediction and evaluation of multi-channel parameters including path loss, RMS delay spread, angular spread and Rician K factor are presented.

A. RELATED WORK

1) LEARNING-BASED CHANNEL PARAMETERS PREDICTION

Recent existing works on learning-based channel prediction can be divided into two categories. *The first category* is to predict channel parameters using image or other features. For example, [28], [31] uses aerial images to predict path loss. The prediction model evolved from the classic image network ResNet-50 [32]. However, because they are high-resolution images, they incur large computational costs. Thus, it is also

feasible to use compressed environmental features to predict path loss. Among them, [33] uses a convolutional neural network (CNN)-based autoencoder to encode 3D buildings information, and combines features such as distance, clutter, street width and Rx height to jointly predict path loss using different learning methods. In addition to outdoor scenarios, some learning-based models are also presented in indoor scenarios [34]. The main function is the trained CNN-based propagation model. The geometry of the indoor environment, the transmitting antenna location and the receiving points are inputted to the model as features, and received signal strength is predicted. However, the channel parameters predicted by this category of model usually only focus on path loss or other single parameters. There is a lack of studies on learning-based multi-channel parameter prediction models.

The second category is to calibrate the parameters of the channel model using learning-based methods. The calibrated channel model is then used to predict the channels [35], [36], [37], [38]. Generally, the calibrated channel model can predict more channel parameters than just path loss, such as delay spread, angular spread and MPCs clusters. However, limited by the scale of channel measurements, overfitting may exist in learning-based calibrations. The model needs to be calibrated again when applied to other scenarios. This makes it difficult for the model to be widely deployed.

2) POINT CLOUD-BASED CHANNEL PREDICTION

Point clouds are three-dimensional data representations consisting of numerous individual points in space. They are commonly obtained through techniques like light detection and ranging (LiDAR) scanning or photogrammetry, providing a detailed and precise depiction of the physical world. With the rise of applications such as autonomous driving and JCAS, LiDAR has gradually become commonplace in vehicles, mobile terminals, etc. The acquisition of point cloud data will be more and more available in these commonly used devices. The MPCs parameters of channels are affected by the environment; there is a mappable relationship between the channel parameters and the environment described by the point cloud. Sensing and prediction of channels based on point clouds will be realized and is promising.

Because of the precise description of the environment by point clouds, it is common in most research to reconstruct the simulation environment using point clouds. The channel is then simulated using RT or other deterministic modeling methods based on the reconstructed environment [39], [40], [41]. Meanwhile, the point cloud-based ray tracing simulators have been proposed in [42], [43]. The point cloud is used for directly calculating the propagation mechanisms such as reflection and transmission by using, e.g., physical optics, instead of reconstructing the environment. Furthermore, prediction of channel characteristics directly based on point clouds has been proposed. For example, a prediction method for the angular characteristics from a point cloud has been proposed by the aperture field integration method in [44]. In addition, the prediction of mmWave received signal strength indicator (RSSI) using a point cloud and supervised learning has been presented in [45]. Specifically, the point cloud is converted into voxels by dividing the 3D space into uniform cubical regions and setting the value of each voxel to be the number of points in that voxel. Then the voxelized data is the input to a 3D CNN and a gradient boosting regression tree. However, the conversion from a point cloud to voxels will increase the computational cost, and the processing of voxels has higher hardware requirements on the performance of deployed devices. Therefore, efficiently utilizing point clouds and other environmental information for channel parameter prediction needs to be further explored.

B. OUR CONTRIBUTIONS

Going beyond the literature, our motivation is to exploit the accessibility of the low-cost point cloud and image data for efficient data-driven channel prediction of both large and small scales parameters, whereas model-driven approach requires more accurate point cloud data.

In this paper, a channel parameter prediction method based on deep learning and point clouds is proposed for LiDAR-assisted mmWave communication scenarios. We carry out LiDAR and panoramic camera-assisted mmWave channel measurement at 60 GHz frequency band. The proposed prediction method, a regression network model, is trained and tested by employing the channel measurement data. The trained network model can accurately predict path loss, RMS delay spread, angular spread and Rician K factor. The main contributions and novelties of this paper are summarized as follows:

- The LiDAR and camera-assisted channel measurement campaign has been carried out in a typical lecture-room environment. Channel measurements are conducted at 60 GHz frequency band, and the channel sounder, LiDAR and panoramic camera are well synchronized. This means that each channel snapshot is matched to point clouds and images generated by LiDAR and panoramic camera, respectively. In addition, based on the geometric relationship between LiDAR and camera, a point cloud and image fusion method is proposed to compress the size of the raw data.
- The channel parameter prediction model based on deep learning is presented. Based on the classic point cloud classification model PointNet, a regression PointNet for channel parameters prediction is proposed. Among them, the point cloud is efficiently utilized as it is directly used as the input to the proposed model, instead of reconstructing the environment or converting it to voxels in literature. In this paper, path loss, RMS delay spread, angular spread and Rician K factor are predicted by the regression PointNet model.
- An area-by-area training and testing method is proposed based on the measurement data. The proposed area-by-area training and testing method avoids overfitting and

improves the generalization performance by using different combinations of areas for training and testing.
- The prediction results of the proposed model are compared with the measurements in terms of the predicted channel parameters. The impact of using different combinations of input information (along with the point cloud) on the prediction performance is analyzed. Finally, the utilization and deployment method of the proposed model is recommended.

The rest of the paper is organized as follows. Section II elaborates the measurement configuration and campaign, as well as the proposed data fusion. Section III presents the classic PointNet model, and the proposed regression PointNet for channel parameters prediction. Based on the proposed model and method in Section III, the prediction results of channel parameters are presented and evaluated in Section IV. Meanwhile, the deployment method of the proposed model is also presented in Section IV. Finally, conclusive remarks are included in Section V.

II. CHANNEL MEASUREMENTS AND DATA SYNCHRONIZATION

In this section, we present the LiDAR-assisted channel measurement campaign conducted in a typical lecture-room environment. Namely, the LiDAR and panoramic camera were installed on the measurement equipment. The channel sounder and measurement environment are introduced in Sections II-A and II-B, respectively. Section II-C presents a synchronized fusion method for the point cloud and image, aiming to compress the size of the raw data.

A. CHANNEL SOUNDER

In this work, a channel sounder with the center frequency of 60.5 GHz was used for the measurements. Fig. 1 shows the channel sounder at the receiver (Rx) side. The Rx composed of a signal digitizer, antenna array, LiDAR and panoramic camera were placed on a mobile robot, as shown in Fig. 1(a). The Rx antenna array consists of 16 horn antennas, as shown in Fig. 1(b); the gain and the beamwidth of each horn antenna is 18.1 dBi and 22.5° (3 dB beamwidth), respectively. As a result, the Rx antenna array covers 360° in the azimuth domain and 45° in the elevation domain. The LiDAR and 360° panoramic camera were mounted above the array antenna. The LiDAR we use is the OS0 from Ouster [46], which is a 128-line LiDAR. It scans 128 and 2048 angles in elevation and azimuth, respectively, and generates point clouds with a resolution of 128 samples × 2048 samples. The panoramic camera we use is the iSTAR Pular from NCTech [47]. The channel sounder, LiDAR and panoramic camera were synchronized in both the temporal and spatial domains through Rubidium clocks and metallic sphere markers respectively. Therefore, the channels, point clouds, and images measured and captured at the same time can be effectively corresponded.

The Tx antenna array consists of eight horn antennas, and the parameters of Tx horn antennas are the same as those of Rx. Since there are only eight Tx antennas in the array, its field-of-view (FoV) is reduced to 180° in the azimuth with respect to the Rx, while the elevation FoV is still 45°. Antenna measurements and calibrations in an anechoic chamber were performed to extract MPCs parameters precisely. The parameters related to the measurement configuration are summarized in Table 1, and more detailed descriptions about the channel sounder can be found in [48], [49].

An arbitrary waveform generator was used as the transmitter (Tx). It generates a repeating 2047 b pseudorandom (PN) codeword that has a chip rate of 2 GHz (0.5 ns chip length). The code is upconverted to 60.5 GHz center frequency and electronically switched through the 128 (16×8) Rx-Tx antenna pairs, requiring only 262 μs for a full channel sweep. At the Rx, the received signal is downconverted and then directly digitized. For each antenna pair, the digitized signal is correlated with the known code, yielding a train of pulses, each corresponding to a distinct path. The advantage of direct digitization is that the correlation is done in post-processing. This means that the channel can be sampled at D_t corresponding to a maximum measurable Doppler shift of

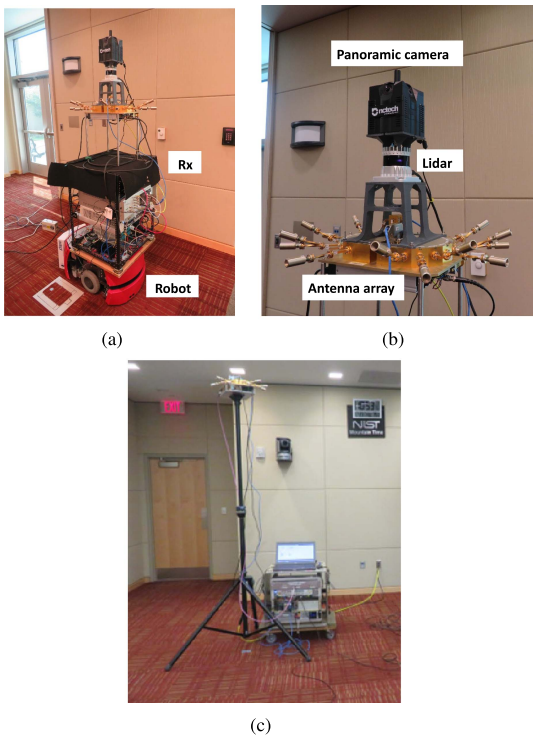
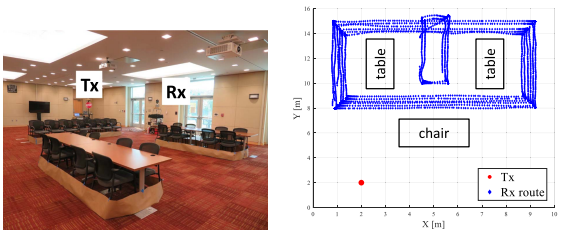


FIGURE 1. (a) The channel sounder at Rx, mounted on a robot. (b) Multisensor structure at Rx. (c) The Tx side of the sounder.

TABLE 1. Configurations of Measurement System

Parameter	Value
Center frequency	60.5 GHz
Bandwidth	2 GHz
Tx antenna array	8 horn antennas
Rx antenna array	16 horn antennas
FoV of Tx antenna array	180° in azimuth, 45° elevation
FoV of Rx antenna array	360° in azimuth, 45° elevation



(a) (b)

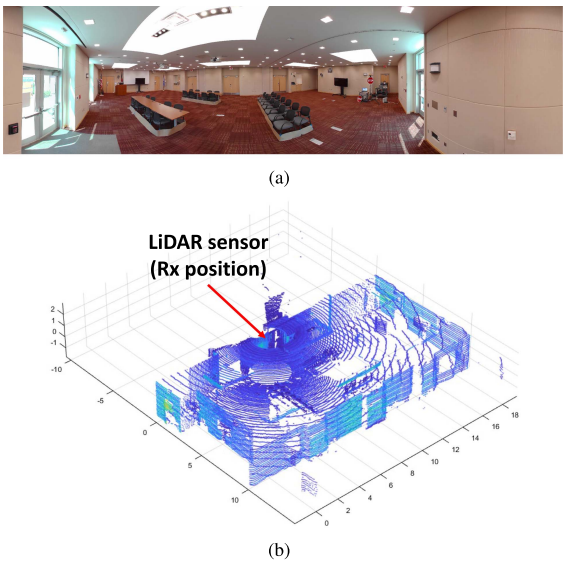
FIGURE 2. (a) The measurement environment. (b) The positions of Tx and Rx (the blue dots are Rx positions during measurement).

1.9 MHz. Thus, scenarios up to a closing speed of 34 km/h can be handled when investigating V2V scenarios.

B. MEASUREMENT CAMPAIGN

The measurement campaign was conducted in a typical lecture room, as shown in Fig. 2(a). Two tables and some chairs were present in the room. The Tx was stationary in one corner of the room. The Rx was mounted on a robot and traversed a predetermined route in the lecture room. The robot was equipped with a laser-guided navigational system, and reported the Rx positions in the environment through a simultaneous localization and mapping (SLAM) algorithm. The coordinates of the Tx and the route traversed by the Rx are shown in Fig. 2(b). The maximum velocity of the robot was 0.2 m/s (much smaller than 34 km/h), and the interval between each Rx position was about 10 cm. At this roughly every 10 cm, a burst of eight small-scale measurements (snapshots) was collected while the robot was in motion, specifically eight measurements were collected in rapid succession (every 24.9 ms), which is equivalent to sampling at about a wavelength (0.5 cm) for a nominal robot speed of 0.2 m/s. Over the total burst, the robot moved a total of about 4 cm. The heights of the Tx and Rx antenna arrays were 2 m and 1.48 m, respectively. This height is consistent with the location of conventional base stations and smart devices in indoor environments. Therefore, it is possible to regard the Tx and Rx as a base station and a mobile user, respectively. At Tx and each Rx position, electronic switches were used to switch each pair of Tx and Rx horn antennas in sequence, and 128 channel impulse responses (CIRs) were recorded for each snapshot. When the robot turns, it needs to stop and turn its body at the same position. Meanwhile, the channel sounder is still sampling at same rate. Therefore, Rx positions are denser at the corner of the route.

The farfield distance of the horn antennas on the Tx and Rx arrays as defined by the Fraunhofer distance is 9 cm (given the 15 mm aperture width of the horns and the 5 mm wavelength at 60 GHz). The minimum Tx-Rx distance across the two areas used in the paper is 6 m, and neither the Tx nor the Rx were closer than 1 m to any other objects in the room. Therefore all measurements were well in the farfield. With calibrated radiation pattern of the Tx and Rx arrays, the space-alternating generalized expectation-maximization (SAGE) super-resolution algorithm [50] was used to extract



(a) (b)

FIGURE 3. Raw images and point clouds. (a) Raw panoramic image for one frame. (b) Raw point cloud for one frame.

propagation MPCs and their parameters for each snapshot. In this paper, the LOS condition was maintained throughout the measurement campaign. The power percentage of LOS paths in all MPCs ranges from 50% to 92% at different Rx positions. It should be noted that NLOS measurements were not conducted in this paper. This is because LiDAR under NLOS conditions cannot capture the entire environment due to occlusion.

C. DATA SYNCHRONIZATION AND FUSION

As mentioned in Section II-A, LiDAR, camera and channel sounder are time-space synchronized. Therefore, each frame point cloud from LiDAR and image from camera corresponded to the channel of each snapshot. Fig. 3 shows the raw image and the point cloud at the same position. Since the camera generates panoramic images, all objects in the environment can be recorded, as shown in Fig. 3(a). The raw point cloud is shown in Fig. 3(b). The color of each point indicates the intensity of laser reflection at that point. The panoramic image shown in Fig. 3(a) corresponds to the depth map in Fig. 5, where each point (i.e., color) of the depth map is the distance in meters from the camera to the objects. The details for obtaining the depth map in Fig. 5 from the point cloud example in Fig. 3(b) are further discussed in the next paragraphs in this Section II-C. From Fig. 3, it was found that some unreasonable points appear outside the scope of the environment. This is because the laser may penetrate objects such as glass, and these points need to be removed.

The resolution of the raw image is 3000 pixels $\times$ 11 000 pixels, and contains RGB values at each pixel. The resolution of the raw point cloud is 128 elevation angle samples $\times$ 2048 azimuth angle samples, and contains three (X, Y, Z) spatial coordinates information at each sample point. Point clouds and images with such a high resolution are difficult

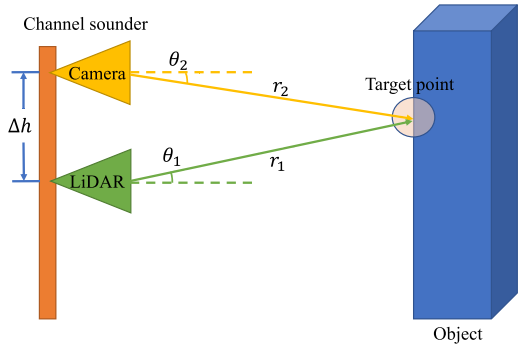


FIGURE 4. Synchronization between camera and LiDAR.

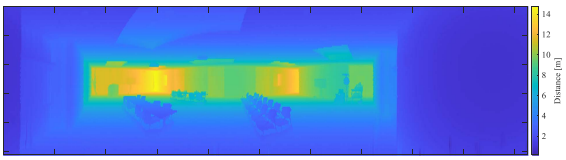


FIGURE 5. The depth map from camera's perspective.

to be used as input at the same time due to the large data size. Therefore, fusion of point clouds and images to reduce the data size is necessary.

Since the camera and LiDAR are mounted at different heights, their distances to a point on the target object are different, as shown in Fig. 4. In order to spatially synchronize the point cloud with the image, the distance between LiDAR and target point needs to be converted to the distance between camera and target point, i.e. the distance of r_2 in Fig. 4 need to be calculated. According to the geometric relationship, r_2 is obtained as

$$r_2 = \frac{r_1 \cos\theta_1}{\cos\theta_2}, \tag{1}$$

where r_1 and θ_1 represent the distance between LiDAR and target point and elevation angle for LiDAR, respectively. θ_2 denotes the elevation angle for camera, and it is calculated as

$$\theta_2 = \arctan\left(\frac{|\Delta h - r_1 \sin\theta_1|}{r_1 \cos\theta_1}\right), \tag{2}$$

where Δh denotes the difference of height between camera and LiDAR. The depth map from camera's perspective is obtained after synchronizing the LiDAR and the camera, as shown in Fig. 5. The color indicates the distance in meters from the camera to the object as encoded in the color bar to the right of the image. Meanwhile, each pixel in the depth map corresponds to a point in the point cloud. Therefore, the connection between point clouds and images is established through depth maps. After down-sampling the image to the resolution of the point cloud, the RGB data is assigned to each point in the point cloud. The flowchart for the fusion of the point clouds and images is shown in Fig. 6. The fusion result is shown in the right figure of the flowchart. The point cloud fused with the RGB data can distinguish more clearly the objects in the environment compared to the raw point cloud. Therefore, the data size is greatly compressed without losing much information through this fusion method.

III. LEARNING-BASED NETWORK ARCHITECTURE FOR POINT CLOUD

In this section, the simple and efficient network architecture of PointNet [51] is introduced in Section III-A. As a classic network for processing point clouds, PointNet has good performance especially in the classification and segmentation of 3D point clouds. Therefore, PointNet will be adapted and used to process the point cloud mentioned in Section II-C and predict multiple channel parameters by decoding features. The original PointNet network performs classification tasks, while channel parameter prediction requires regression functions. Therefore, we modify the network architecture based on PointNet to adapt it to the regression task of channel parameter prediction in Section III-B. In addition, an area-by-area training and testing method is proposed under the limited amount of measurement data and presented in Section III-C.

A. POINTNET FOR 3D CLASSIFICATION AND SEGMENTATION - A RECAP

Both point clouds and images are important information sources and have specific data structures. However, point clouds have two following unique characteristics that differ from images. The first characteristic is permutation invariance. Unlike images, where pixels are arranged in a particular order, most point clouds are unordered. This means that the points describing the same object will have different orders and arrangements in point clouds. The second characteristic is transformation invariance. Both rotating and translating points cannot modify the global point cloud characteristics. Due to the two above characteristics of the point cloud, conventional machine learning architectures such as supporting vector regression and random forest cannot be used to process point clouds. Meanwhile, operations such as convolution cannot be directly used on the raw point cloud. Therefore, most researchers transform such point cloud data to regular 3D voxel grids or collections of images, which can be processed using a convolutional neural network (CNN), etc. However, the processing complexity of the point cloud using the above methods is high and cannot be used in certain scenarios. PointNet solves these problems, and makes deep learning directly applicable to 3D point cloud tasks [51].

The original PointNet architecture is shown in the yellow box of the classification PointNet in Fig. 7. Most of its modules are composed of a multilayer perceptron (MLP) and fully connected (FC) layers. Therefore, compared to a CNN-based image or voxel network, the structure of PointNet is simpler and more efficient. In order to solve the transformation invariance and permutation invariance of point clouds, PointNet uses a general function f defined as

$$f(\{x_1, \ldots, x_n\}) \approx g(h(x_1), \ldots, h(x_n)), \tag{3}$$

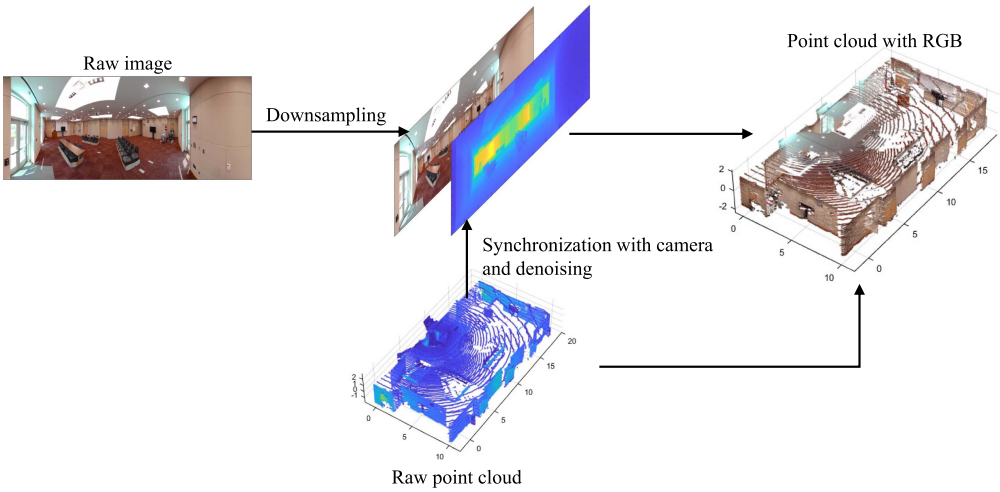


FIGURE 6. The flowchart and result for the fusion of point cloud and image.

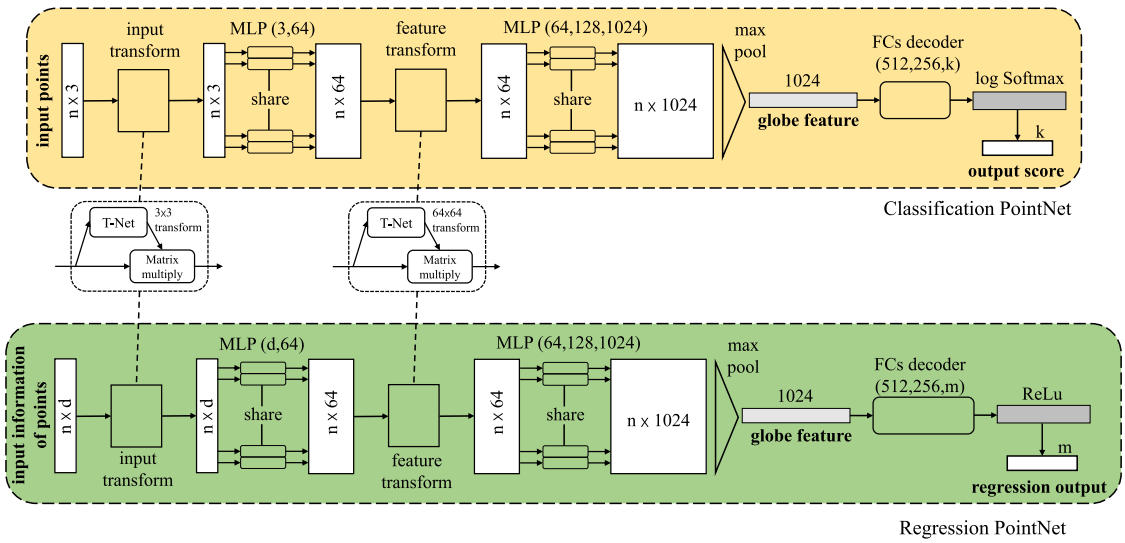


FIGURE 7. PointNet architecture for traditional classification with input of block point cloud (top); Modified PointNet architecture for proposed regression with input of fused data of entire environment (bottom).

where $\{x_1, \ldots, x_n\}$ is the input point or feature vector. Function h is approximated by an MLP network to deal with the transformation invariance of point clouds. Due to the constraint of loss function during the training, the MLP parameter of h will learn the transformation that is most conducive to the final prediction, such as transforming the point cloud to the same direction. Function g is a symmetric function, and can deal with the permutation invariance of point clouds. The symmetric function is not sensitive to the order of the input, such as addition, dot product and max pooling. Among them, g is the max pooling function in PointNet. The input dimension of PointNet is $n \times 3$, where n represents the number of points. After the input is subjected to the max pooling layer, the maximum value in each dimension will be obtained. Each dimension's feature is not related to its order, which ensures the robustness to the order of point clouds. The general function f corresponds to the two T-Net structures in Fig. 7, which are defined in [51]. Among them, the first T-Net transforms the input $n \times 3$ dimensional point cloud. The transformed results are promoted to $n \times 64$ dimensional feature vectors by an MLP network. Then the feature vector is transformed by the second T-Net. The transformed $n \times 64$ dimensional feature vector is generated as the input of the MLP network with 3 layers. Each point has 1024 dimensional features. Then the max pooling is applied to aggregate point features, and the global feature is obtained. A decoder composed of FC layers decodes the global features and outputs the classification scores for k classes after the log Softmax activation function.

The focus of this paper is to predict four channel LSPs, that is, path loss, delay spread, angular spread and Rician

K factor. Therefore, the PointNet model needs to be adapted from classification to regression.

B. REGRESSION POINTNET FOR CHANNEL PREDICTION

The green box at the bottom of Fig. 7 shows the proposed architecture of regression PointNet. In this paper, regression PointNet will be applied to predict channel LSPs. In the input layer, the point cloud with a fixed number of points is used as input. Meanwhile, the original point cloud data needs to be downsampled due to limited GPU memory. The original classification PointNet divides the point cloud of environment into 3D blocks with side lengths of 1 m, and randomly downsamples the point cloud in blocks to 4096 points. The downsampled point cloud blocks are then inputted into PointNet. However, wireless channels are associated with the entire environment. Hence partitioning of the environment will not be suitable for channel prediction. In addition, due to the sparse feature of point cloud, the points far away from the LiDAR sensor will be sparser than the nearby points. Random downsampling will make sparse point clouds further away from LiDAR even sparser. This will lose some of the environment characteristics. Therefore, the farthest point sampling (FPS) is used in this paper. The central idea of FPS is to always sample the point farthest from the current sampling point. In the point cloud data of this paper, FPS can preferentially sample the sparser part of the point cloud to preserve the characteristics of the environment as much as possible. In this paper, 32 768 points are sampled and inputted into the regression PointNet. After the downsampling of data fusion and point cloud, the storage size of each frame of point cloud and image was compressed from 15.7 MB to 1.46 MB. The total data size was compressed from 16.6GB to 1.5 GB, which is a reasonable storage size to input into the GPU for training. In addition, other information will also be given about the point cloud, such as Rx coordinates, the color and segmented masks of the points. Therefore, the input layer dimension in the regression PointNet is $n \times d$, where n is the number of points ($n = 32768$) and d is the dimension of input information. The impact of input information on prediction will be analyzed in Section IV-B. In the output layer, we change the log Softmax activation function into ReLu to adapt the regression task. The FCs decoder decodes m predicted channel parameters and outputs them after the ReLu activation function. In terms of network complexity, the proposed regression PointNet is close to the conventional classification PointNet. Its space complexity (number of parameters processed in the network) is 3.5 e^6 [51], while the scale of point cloud models based on 3D CNN, as mentioned in the Related Work in Introduction, is usually 10 times that of PointNet. Meanwhile, PointNet is much more scalable compared to the 3D CNN, because it's space and time complexity is $O(N)$, which is linear in the number of input points.

C. AREA-BY-AREA TRAINING AND TESTING METHOD

The conventional deep learning tasks, such as image classification, usually randomly divide the entire dataset into a training set, a validation set, and a test set according to a certain proportion, which is called "random shuffling". The datasets of such tasks are usually large and diverse. This makes their models more generalizable. However, channel measurement is difficult to cover a wide area due to the high cost. Training the network with limited channel data is likely to lead the model overfitting by randomly shuffling the dataset. Furthermore, most of the samples in the testing dataset can be found in the training dataset in a close position using random shuffling datasets. Since the close positions have similar channel parameters, this has a negative impact on the generalization of the model. To address this issue, an area-by-area training and testing method is proposed in this paper. We divide the measurement area in Fig. 2(b) into 5 areas according to the route, as shown in Fig. 8. The rule of thumb of dividing the measurement into areas is that in each area the propagation path composition is distinctive from the other areas. In each model training, four areas are used as the training dataset, and the remaining 1 area is used as the testing dataset. Based on the proposed area-by-area dataset division, the regression PointNet model will be trained 5 times and it will generate 5 different models using different datasets.

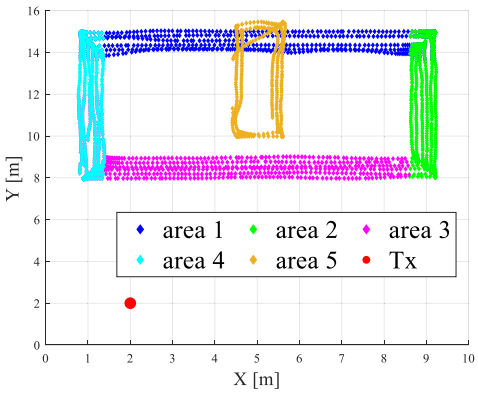


FIGURE 8. The area-by-area division with 5 areas.

We implement the regression PointNet model in PyTorch [52]. For other hyperparameters, we keep the adaptive moment estimation (Adam) optimizer and initial learning rate of 0.001 with original PointNet. The learning rate is decayed by 0.7 times every 60 epochs. In addition, the negative log-likelihood loss function is used in the classification PointNet, while we use the mean square error loss function for our regression task. Meanwhile, to ensure model convergence, we increase the number of training epochs from the original 200 to 500. The batch size is set to 10 due to limited GPU memory. Each dimension information of points in the input layers and labels in the output layers are normalized to between 0 and 1 using the min-max scaling as follows

$$z' = \frac{z - z_{\min}}{z_{\max} - z_{\min}}, \tag{4}$$

where z is the input information or output labels. $z_{\min}$ and $z_{\max}$ are the minimum and maximum values in the data set $\{z\}$, respectively. The $z_{\min}$ and $z_{\max}$ of each parameter need to

TABLE 2. The Hyperparameters of Training Networks

Hyperparameters	Value
Initial learning rate	0.001
Learning rate decay	Decay 0.7 every 60 epochs
Epoch number	500
Batch size	10
Loss function	Mean square error
Optimizer	Adam

be recorded in order to recover the original values from the normalized values after prediction. The hyperparameters of training networks are summarized in Table 2.

IV. LARGE-SCALE PARAMETERS PREDICTION

In this section, we predict channel LSPs using the proposed regression PointNet model and training method. Root mean square error (RMSE) is used to evaluate the prediction accuracy. The RMSE is defined as follows

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(\hat{y}_i - y_i)^2}, \tag{5}$$

where $\hat{y}_i$ and y_i is the prediction and measurement value, respectively. M is the number of test data. According to (5), it is found that RMSE has the same unit as the channel parameter to be evaluated. Therefore, it can visually show the error between measured and predicted channel parameters.

We first show the prediction results using the all information of points, and compare with the classical path loss models. Then the impact of point information on prediction accuracy is analyzed and insights are given.

A. PREDICTION RESULTS AND EVALUATION

1) PATH LOSS

The path loss can be calculated from the path gain of MPCs estimated from SAGE as follows

$$\mathrm{PL} = -10\log_{10}\left(\sum_{i=1}^{N} P_i\right), \tag{6}$$

where P_i is the path gain of the i-th MPC, and N is the number of MPCs. In the evaluation of path loss, two benchmarks are used to compare with the proposed model. The first benchmark is the close-in (CI) freespace reference distance path loss model [11], which is defined as follows

$$\begin{aligned}\mathrm{PL}^{\mathrm{CI}}(f, d_{\mathrm{TR}})[\mathrm{dB}] &= \mathrm{FSPL}(f, d_0)\,[\mathrm{dB}] \\ &+ 10n_{\mathrm{PL}}\log_{10}\left(\frac{d_{\mathrm{TR}}}{d_0}\right) + \chi_\sigma^{\mathrm{CI}},\ d_{\mathrm{TR}} \geq d_0\end{aligned} \tag{7}$$

where f is the center frequency in GHz. d_{TR} is the distance between Tx and Rx in meters. d_0 is the reference distance and is set to 1 m. n_{PL} denotes the path loss exponent (PLE), and $\chi_\sigma^{\mathrm{CI}}$ is a zero-mean Gaussian random variable with a standard deviation σ in decibels. In the evaluation, the n_{PL} and σ of CI model are obtained by fitting the test datasets. The second benchmark is the 3GPP indoor hotspot (InH) path loss model in LOS condition [53], which is defined as follows

$$\begin{aligned}\mathrm{PL}_{\mathrm{InH\text{-}LOS}}(f, d_{\mathrm{TR}})[\mathrm{dB}] &= 32.4 \\ &+ 17.3\log_{10}(d_{\mathrm{TR}}) + 20\log_{10}(f),\end{aligned} \tag{8}$$

where the meaning of the variables is consistent with the CI model. The 3GPP InH model can directly obtain the path loss according to the distance between transceivers without fitting any parameters.

The RMSE between measurements and predictions using different methods is summarized in Table 3. In this table, the CI model serves as a reference, because it is directly fitted to the measurement data of test datasets (thus also has the smallest RMSE); the 3GPP InH model does not fit to the measurement data and is a prediction method to be compared to our proposed approach. The average value of RMSE represents the overall performance of the model on the dataset. The proposed regression PointNet significantly outperforms the 3GPP InH model among models that do not use prior information on the test datasets. In some areas, the prediction error of the regression PointNet model is even close to that of the CI model. Table 3 also shows that all the models perform the worst in area 1, including the proposed method. Fig. 9 shows the measured and predicted path loss for different models. It is found that the 3GPP InH model differs significantly from the measured path loss in both the best and worst cases. In the best case of Fig. 9(a), the variation in the trend of path loss is smooth in area 2, thus having the smallest RMSE among all the models. Meanwhile, the proposed regression PointNet model has similar values to the CI model, and both are close to the measured path loss in Fig. 9(a). In the worst case of Fig 9(b), the measured path loss fluctuates significantly in area 1. It is difficult for the slope-intercept model to capture this variation in this test area, resulting in the largest RMSE in all tested areas. If the fitted CI model is used as a baseline in Fig. 9(b), it is found that some measured samples deviate significantly from the baseline. This is due to the birth-death behavior of dominant MPCs. In addition, the prediction of regression PointNet is as close as possible to the measured samples that deviate significantly. This shows that the proposed model can characterize this variation caused by birth-death behavior to some extent.

2) RMS DELAY SPREAD

As one of the important characteristics of wireless channels, RMS delay spread is commonly used to measure MPCs dispersion in the delay domain. It is calculated as the second-order central moment of the power delay profile. In this paper, we use the path gain and delay of MPCs to calculate the RMS delay spread as follows [12]

$$\tau_{RMS} = \sqrt{\overline{\tau^2} - \bar{\tau}^2}, \tag{9}$$

with

$$\overline{\tau^2} = \frac{\sum_{i=1}^{N} P_i\tau_i^2}{\sum_{i=1}^{N} P_i}, \quad \bar{\tau} = \frac{\sum_{i=1}^{N} P_i\tau_i}{\sum_{i=1}^{N} P_i}, \tag{10}$$

TABLE 3. RMSE of Fitted or Predicted Path Loss Using Different Methods

Path loss model (Path loss ranging from 81.4~92.3 dB)	Area of test datasets						
	Area 1	Area 2	Area 3	Area 4	Area 5	Mean	Standard deviation
CI model fitting error	1.24 dB	0.53 dB	1.19 dB	1.26 dB	1.21 dB	1.09 dB	0.31 dB
3GPP InH model prediction error	2.77 dB	2.35 dB	2.38 dB	2.51 dB	2.76 dB	2.55 dB	0.20 dB
Regression PointNet prediction error	1.77 dB	1.23 dB	1.71 dB	1.57 dB	1.51 dB	1.56 dB	0.21 dB

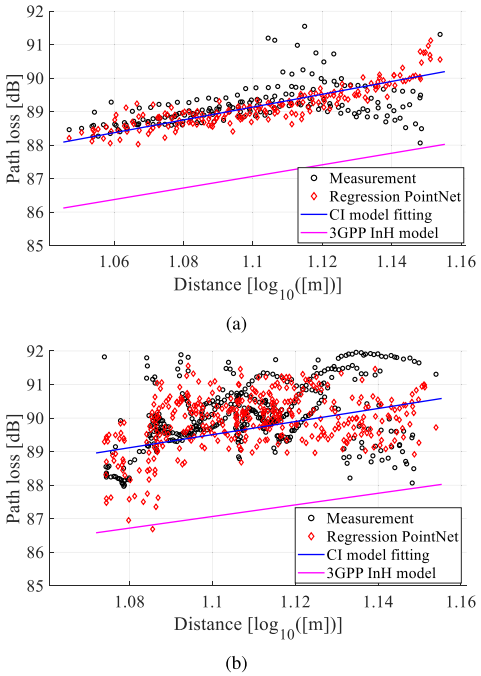


FIGURE 9. The results of path loss between measurements and predictions using different models. (a) The best case for regression PointNet (area 2). (b) The worst case for regression PointNet (area 1).

where P_i and τ_i represent the path gain and delay of the i-th MPC, respectively. N is the number of MPCs. The RMSE of RMS delay spread predicted by regression PointNet in all test areas are listed in Table 4. Fig. 10 shows the prediction results of RMS delay spread in the best and worst cases. In Fig. 10(a) and (c), each sample represents the measured and predicted results at a certain position. It is found that the predictions of regression PointNet largely agree with the measured trends in both the best and worst cases. Fig. 10(b) and (d) show the cumulative distribution functions (CDF) of the measured and predicted RMS delay spread. It is found that the proposed model is close to the measured samples in CDF curves.

3) ANGULAR SPREAD

Similar to the time-delay domain, the channel has angular spread characteristics in the spatial domain. The angular spread is defined in 3GPP 38.901 [53] as follows

$$\mathrm{AS} = \sqrt{-2\ln\left(\left|\frac{\sum_{i=1}^{N}\exp(j\phi_i)\cdot P_i}{\sum_{i=1}^{N}P_i}\right|\right)}, \tag{11}$$

where P_i is the path gain for the i-th MPC, and ϕ_i is the angle of the i-th MPC given in radians. N is the number of MPCs. The angle includes azimuth angle of arrival, elevation angle of arrival, azimuth angle of departure and elevation angle of departure. In this paper, we focus on the analysis of azimuth angle of arrival. Therefore, the angular spread in this paper defaults to the azimuth spread of arrival. In addition, the unit of angular spread is converted from radians to degrees, which is more intuitively displayed in the prediction results and RMSE evaluation. The RMSE between measurement and prediction using the proposed model in all test areas are listed in Table 4. The average RMSE of angular spread across all areas is 7.39°. The minimum and maximum RMSE of angular spread corresponds to area 2 and area 1, respectively. Fig. 11 shows the prediction results and corresponding CDFs of angular spread in the best and worst cases. In the best case, it is found from Fig. 11(a) that the predicted trends of the proposed model are consistent with the measurements. Meanwhile, the proposed model can also cope with the rapid change of angular spread caused by the birth and death of MPCs. This is also confirmed in Fig. 11(b). The CDF curves of measurement and prediction almost coincide for most samples. In addition, it is found from Fig. 11(a) and (b) that some samples have large and rapid changes in some continuous index. This is because some unreasonable measurement samples are filtered out in the data sets, which make the channel parameter values discontinuous.

4) RICIAN K FACTOR

Rician K factor is used to describe the power relationship between the dominant MPCs and other MPCs. In this paper, the LOS condition is always maintained throughout the measurement campaign. Therefore, the Rician K factor is defined as follows

$$KF = 10\log_{10}\left(\frac{P_{LOS}}{\sum_{i=1}^{N}P_i - P_{LOS}}\right), \tag{12}$$

where P_{LOS} is the path gain of LOS path. N is the number of MPCs, and P_i is the path gain of the i-th MPC. The prediction

TABLE 4. RMSE of Predicted LSPs Using Regression PointNet

LSPs (Parameters range)	Area of test datasets						
	Area 1	Area 2	Area 3	Area 4	Area 5	Mean	Standard deviation
RMS delay spread (0.38~5.5 ns)	0.95 ns	0.91 ns	0.94 ns	0.88 ns	0.82 ns	0.90 ns	0.05 ns
Angular spread (6.44°~41.38°)	9.05°	5.61°	8.78°	7.56°	5.94°	7.39°	1.58°
Rician K factor (0.41~16.39 dB)	2.16 dB	2.28 dB	3.77 dB	2.48 dB	2.65 dB	2.67 dB	0.64 dB

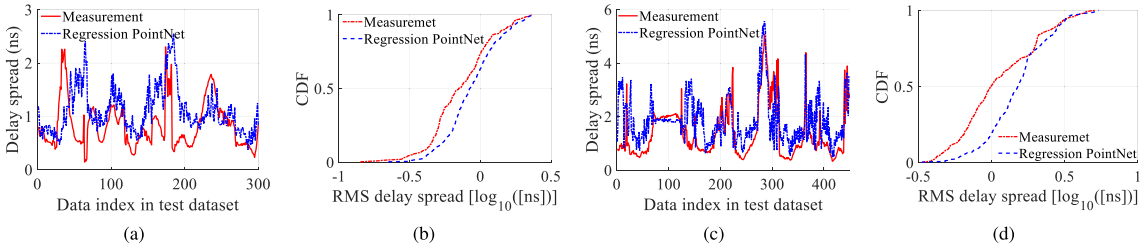


FIGURE 10. The results of RMS delay spread between measurements and predictions. Among them, one index in the horizontal axis in Fig. (a) and Fig. (c) represents the measurement and corresponding prediction results at one position. (a) The best case for regression PointNet (area 5). (b) The CDFs of measurement and prediction for the best case. (c) The worst case for regression PointNet (area 1). (d) The CDFs of measurement and prediction for the worst case.

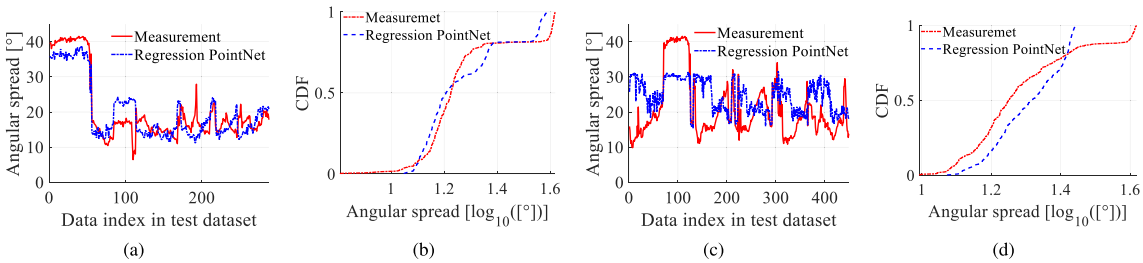


FIGURE 11. The results of angular spread between measurements and predictions. Among them, one index in the horizontal axis in Fig. (a) and Fig. (c) represents the measurement and corresponding prediction results at one position. (a) The best case for regression PointNet (area 2). (b) The CDFs of measurement and prediction for the best case. (c) The worst case for regression PointNet (area 1). (d) The CDFs of measurement and prediction for the worst case.

RMSE of Rician K factor in all test areas are list in Table 4. It is found that most of the test areas can achieve a small RMSE except for area 3. Similarly, Fig. 12 shows the prediction results and corresponding CDFs of the Rician K factor in the best and worst cases. In the best case of Fig. 12(a) and (b), it is found that most of the predicted samples are good matches to the measured samples. Meanwhile, the proposed model can still predict well for some rapidly changing samples and extreme values in the best case of Fig. 12(a). In the worst case, there is a relatively large RMSE in area 3 due to the decrease in the prediction performance of the proposed model for extreme values samples. But the predicted value is still within a reasonable range in Fig. 12(c). Thus the proposed regression PointNet model is reliable.

B. IMPACT OF INPUT INFORMATION

In Section III-B, we mentioned that the proposed regression PointNet can modify and select the dimensionality of the input information. Therefore, the impact of input information on prediction accuracy will be evaluated in this section. In this paper, the input information that can be used are point cloud (X, Y, Z), Rx position (X_{Rx}, Y_{Rx}, Z_{Rx}), the color of point (R, G, B) and segmented masks. Among them, the segmented mask is the category of objects using semantic segmentation for the image, as shown in Fig. 13(b). The segmented masks are fused into the point cloud using the same method as the image, which is described in Fig. 6 and Section II-C. Thus, a total of 10 dimensional channels ($d = 10$) can be used as input. Among them, the point cloud is considered as the basis

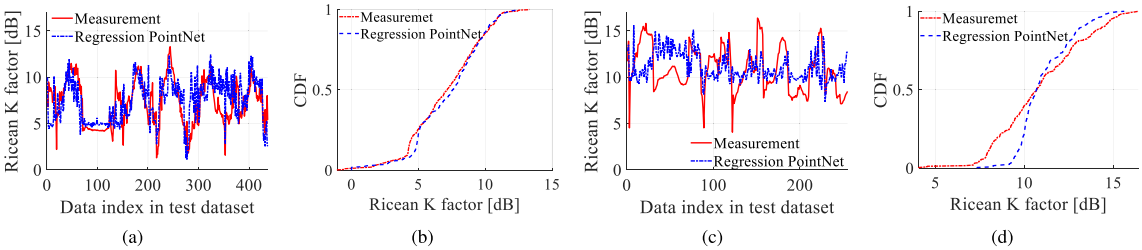


FIGURE 12. Results of Rician *K* factor between measurements and predictions. Among them, one index in the horizontal axis in Fig. (a) and Fig. (c) represents the measurement and corresponding prediction results at one position. (a) The best case for regression PointNet (area 1). (b) The CDFs of measurement and prediction for the best case. (c) The worst case for regression PointNet (area 3). (d) The CDFs of measurement and prediction for the worst case.

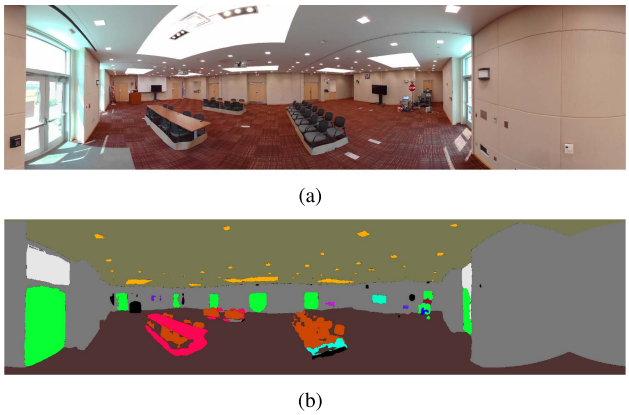


FIGURE 13. (a) Raw panoramic image. (b) Segmented masks from image.

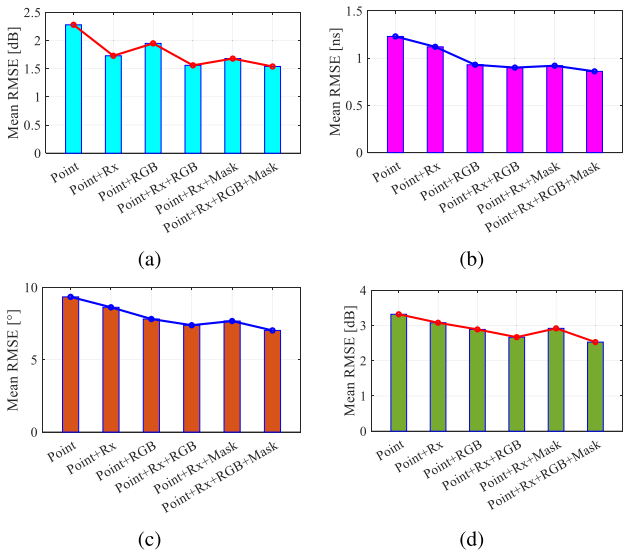


FIGURE 14. The mean RMSE using different input information on the LSP prediction. (a) Path loss. (b) RMS delay spread. (c) Angular spread. (d) Rician *K* factor.

since it describes the geometric characteristics of the environment. We will add other information to the point cloud and evaluate their impact on prediction accuracy.

Fig. 14 shows the impact of using different input information on the prediction of different LSPs. On the X-axis, the Point, Rx, RGB and Mask denote point cloud, Rx position, the color of the point and segmented mask of the points, respectively. The Y-axis is the mean RMSE among the 5 test areas. It is found that the Rx position and the color of the point improve predictions for all LSPs. Among them, the improvement of the path loss prediction by Rx position is significantly greater than that of the other three LSPs. This is because the addition of the Rx position enriches the distance information in the input, which is more important for the prediction of path loss. Please note that path loss and other LSPs can still be predicted without Rx positions. This is due to the fact that the point cloud scanned by the LiDAR contains the feature of Rx position. For example, the object above the Rx cannot be scanned due to the limited FOV of LiDAR, and it creates a void without points. This is shown at the Rx position in Fig. 3(b). The feature of Rx position can be learned by the network to a certain extent. However, the LOS path contributes most of the received power under LOS conditions, which depends more on the Rx position and the distance between the transceivers. Therefore, accurate Rx positions can reduce the learning cost of the model, and improve the prediction performance of path loss.

In addition, color information also improves path loss prediction. Point clouds can construct geometric information of objects, but cannot distinguish objects. Objects can be classified and segmented through color information, and associated with the material electromagnetic parameters of the object. For example, the door or window embedded in the wall shown in Fig. 13(a) is difficult to distinguish only through the geometric information provided by the point cloud. But they can be classified and distinguished by the difference in color. Object properties can be related to material parameters, which is beneficial for the prediction of power, especially for non-dominant MPCs. Therefore, compared with Rx position, the addition of color information has a more significant improvement in RMS delay spread, angle spread and Rician *K* factor. This is because these three LSPs evaluate the impact of non-dominant MPCs on channel characteristics. Therefore, color information improves their predictions more significantly. In addition, the segmented mask can be regarded as the extraction and compression for color information. It more precisely describes the object category for each point in the point cloud. Thus, the segmented mask can replace the color of point to a certain extent. For example, for the prediction of RMS

TABLE 5. RMSE of Predicted LSPs Using Different Parameter Models and Mean Model in Area 5

LSPs	Model 1	Model 2	Model 3	Model 4	Mean model
Path loss	1.98 dB	1.28 dB	1.42 dB	1.65 dB	**1.17 dB**
RMS delay spread	1.23 ns	**0.84 ns**	0.85 ns	0.93 ns	0.87 ns
Angular spread	7.59°	6.14°	**5.45°**	6.87°	5.85°
Rician K factor	3.21 dB	2.89 dB	3.19 dB	**2.67 dB**	2.78 dB

delay spread in Fig. 14(b), the fourth bar (Point+Rx+RGB) and fifth bar (Point+Rx+Mask) have similar mean RMSE. Compared to using the raw color, this means that using only the segmented mask can achieve similar performance on the prediction of some channel parameters. In addition, it is found from the sixth bar (Point+Rx+RGB+Mask) in Fig. 14, using all information exhibits the best prediction performance for all LSPs. On the one hand, more input information has a greater improvement on the prediction of LSPs. On the other hand, the addition of segmented mask reduces the learning cost of the network in the classification for objects. This is also beneficial for the prediction of LSPs.

C. MODEL DEPLOYMENT TRAINED AREA-BY-AREA

In conventional deep learning tasks, usually one set of model parameters is trained and finally deployed. In this paper, however, the regression PointNet model will be trained five times using different training and testing datasets, and five sets of model parameters will be generated, as mentioned in Section III-C. There are two deployment methods that can be used for the proposed model. The first method is to select a single model to deploy in all models. For example, the model that has the best performance can be selected, i.e. the model with the smallest mean RMSE in the test area. The second method is that the mean of all model predictions is considered the final prediction results, which is called the "mean model". To compare the two deployment methods, we use area 5 in Fig. 8 as the comparison area predicted by the model to be deployed. The remaining four areas are trained using the area-by-area training method, and four sets of model parameters are generated, e.g. model 1 (trained by datasets in area 2, 3, 4 and to predict area 5), model 2 (trained by datasets in area 1, 3, 4 and to predict area 5). Table 5 lists the RMSE in area 5 predicted by the four models with different parameters and the mean model. Data in bold font indicate the smallest RMSE in all models. It is found that the single model has different performance on prediction for different LSPs. Therefore, it is difficult to select an optimal model that predicts all LSPs well. In addition, the mean model has the smallest RMSE in path loss prediction. Meanwhile, the RMSE of the mean model on other LSPs is also close to the best single model. This means that the mean model has a more balanced performance in the prediction of all LSPs. Therefore, it would be more appropriate to use the mean model when deploying.

V. CONCLUSION

In this paper, a novel LiDAR-assisted mmWave channel measurement campaign at 60 GHz was conducted in a typical lecture room. The data generated from channel sounder, LiDAR and panoramic camera were synchronized in time-spatial domain. Based on geometric relationship between LiDAR and camera, a novel data fusion method for point cloud and image has been proposed, and size of input data has been compressed. Then by adapting PointNet model, a regression PointNet model applied for channel parameters prediction has been proposed. Meanwhile, an area-by-area training and testing method has been proposed under the limited amount of measurement data. This protects the proposed model from overfitting and improve generalization performance by using different combinations of areas for training and testing. Based on the proposed model and the fused point cloud, the prediction performance of channel parameters in test areas have been evaluated. The mean RMSE of path loss, RMS delay spread, angular spread and Rician K factor are 1.56 dB, 0.90 ns, 7.39° and 2.67 dB, respectively.

In addition, the impact of different combinations of input information on prediction accuracy has been analyzed. The additional Rx position, color and segmented mask, in addition to point cloud data, improve prediction performance. This is because of more information being fed to learning network and thus reducing learning cost of the network. Finally, we compared different deployment methods for the proposed model and concluded that so far using the mean prediction of all models has the best performance.

REFERENCES

[1] T. S. Rappaport et al., "Millimeter wave mobile communications for 5G cellular: It will work!," *IEEE Access*, vol. 1, pp. 335–349, 2013.

[2] S. A. Busari, K. M. S. Huq, S. Mumtaz, L. Dai, and J. Rodriguez, "Millimeter-wave massive MIMO communication for future wireless systems: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 836–869, Secondquarter 2018.

[3] W. Tashan, I. Shayea, S. Aldirmaz-Colak, M. Ergen, M. H. Azmi, and A. Alhammadi, "Mobility robustness optimization in future mobile heterogeneous networks: A survey," *IEEE Access*, vol. 10, pp. 45522–45541, 2022.

[4] J. A. Zhang et al., "Enabling joint communication and radar sensing in mobile networks—A survey," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 306–345, Firstquarter 2022.

[5] C.-X. Wang, J. Huang, H. Wang, X. Gao, X. You, and Y. Hao, "6G wireless channel measurements and models: Trends and challenges," *IEEE Veh. Technol. Mag.*, vol. 15, no. 4, pp. 22–32, Dec. 2020.

[6] H. Mi et al., "Multi-scenario millimeter wave wireless channel measurements and sparsity analysis," *China Commun.*, vol. 19, no. 11, pp. 16–31, Nov. 2022.

[7] R. He, B. Ai, G. Wang, M. Yang, C. Huang, and Z. Zhong, "Wireless channel sparsity: Measurement, analysis, and exploitation in estimation," *IEEE Wireless Commun.*, vol. 28, no. 4, pp. 113–119, Aug. 2021.

[8] R. He et al., "A kernel-power-density-based algorithm for channel multipath components clustering," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7138–7151, Nov. 2017.

[9] M. Giordani, M. Polese, A. Roy, D. Castor, and M. Zorzi, "A tutorial on beam management for 3GPP NR at mmWave frequencies," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 173–196, Firstquarter 2019.

[10] Y. Cui, F. Liu, X. Jing, and J. Mu, "Integrating sensing and communications for ubiquitous IoT: Applications, trends, and challenges," *IEEE Netw.*, vol. 35, no. 5, pp. 158–167, Sep./Oct. 2021.

[11] S. Sun et al., "Investigation of prediction accuracy, sensitivity, and parameter stability of large-scale propagation path loss models for 5G wireless communications," *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 2843–2860, May 2016.

[12] X. Cai, G. Zhang, C. Zhang, W. Fan, J. Li, and G. F. Pedersen, "Dynamic channel modeling for indoor millimeter-wave propagation channels based on measurements," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5878–5891, Sep. 2020.

[13] M. Yang et al., "Dynamic V2V channel measurement and modeling at street intersection scenarios," *IEEE Trans. Antennas Propag.*, vol. 71, no. 5, pp. 4417–4432, May 2023.

[14] Z. Yun and M. F. Iskander, "Ray tracing for radio propagation modeling: Principles and applications," *IEEE Access*, vol. 3, pp. 1089–1100, 2015.

[15] Y. Miao, Q. Gueuning, M. Gan, and C. Oestges, "Adding diffuse scattering correlation to effective roughness models in ray tracing," in *Proc. IEEE 2017 11th Eur. Conf. Antennas Propag.*, 2017, pp. 828–830.

[16] V. Degli-Esposti, F. Fuschini, E. M. Vitucci, and G. Falciasecca, "Measurement and modelling of scattering from buildings," *IEEE Trans. Antennas Propag.*, vol. 55, no. 1, pp. 143–153, Jan. 2007.

[17] D. Bilibashi, E. M. Vitucci, and V. Degli-Esposti, "On dynamic ray tracing and anticipative channel prediction for dynamic environments," *IEEE Trans. Antennas Propag.*, vol. 71, no. 6, pp. 5335–5348, Jun. 2023.

[18] R. Charbonnier et al., "Calibration of ray-tracing with diffuse scattering against 28-Ghz directional urban channel measurements," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 14264–14276, Dec. 2020.

[19] T. Alwajeeh, P. Combeau, R. Vauzelle, and A. Bounceur, "A high-speed 2.5D ray-tracing propagation model for microcellular systems, application: Smart cities," in *Proc. IEEE 2017 11th Eur. Conf. Antennas Propag.*, 2017, pp. 3515–3519.

[20] Z. Dai and R. J. Watson, "Accelerating a ray launching model using GPU with CUDA," in *Proc. IEEE 12th Eur. Conf. Antennas Propag.*, 2018, pp. 1–3.

[21] Z. Ma et al., "Impact of UAV rotation on MIMO channel characterization for air-to-ground communication systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12418–12431, Nov. 2020.

[22] Z. Ma, B. Ai, R. He, Z. Zhong, and M. Yang, "A non-stationary geometry-based MIMO channel model for millimeter-wave UAV networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 2960–2974, Oct. 2021.

[23] R. He, B. Ai, G. L. Stüber, G. Wang, and Z. Zhong, "Geometrical-based modeling for millimeter-wave MIMO mobile-to-mobile channels," *IEEE Trans. Veh. Technol.*, vol. 67, no. 4, pp. 2848–2863, Apr. 2018.

[24] C. Huang et al., "Artificial intelligence enabled radio propagation for communications—Part II: Scenario identification and channel modeling," *IEEE Trans. Antennas Propag.*, vol. 70, no. 6, pp. 3955–3969, Jun. 2022.

[25] C. Huang et al., "Machine learning-enabled LOS/NLOS identification for MIMO systems in dynamic environments," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 3643–3657, Jun. 2020.

[26] M. Yang et al., "Machine-learning-based scenario identification using channel characteristics in intelligent vehicular communications," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 3961–3974, Jul. 2021.

[27] C. Huang et al., "Artificial intelligence enabled radio propagation for communications—Part I: Channel characterization and antenna-channel optimization," *IEEE Trans. Antennas Propag.*, vol. 70, no. 6, pp. 3939–3954, Jun. 2022.

[28] H. F. Ates, S. M. Hashir, T. Baykas, and B. K. Gunturk, "Path loss exponent and shadowing factor prediction from satellite images using deep learning," *IEEE Access*, vol. 7, pp. 101366–101375, 2019.

[29] N. Kuno, M. Inomata, M. Sasaki, and W. Yamada, "Deep learning-based path loss prediction using side-view images in an UMa environment," in *Proc. IEEE 2022 16th Eur. Conf. Antennas Propag.*, 2022, pp. 1–5.

[30] S. P. Sotiroudis, K. Siakavara, G. P. Koudouridis, P. Sarigiannidis, and S. K. Goudos, "Enhancing machine learning models for path loss prediction using image texture techniques," *IEEE Antennas Wireless Propag. Lett.*, vol. 20, no. 8, pp. 1443–1447, Aug. 2021.

[31] K. Mao et al., "A2G channel measurement and characterization via TNN for UAV multi-scenario communications," in *Proc. IEEE 2022 IEEE Glob. Commun. Conf.*, 2022, pp. 4461–4466.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[33] A. Gupta, J. Du, D. Chizhik, R. A. Valenzuela, and M. Sellathurai, "Machine learning-based urban canyon path loss prediction using 28 Ghz manhattan measurements," *IEEE Trans. Antennas Propag.*, vol. 70, no. 6, pp. 4096–4111, Jun. 2022.

[34] A. Seretis and C. D. Sarris, "Toward physics-based generalizable convolutional neural network models for indoor propagation," *IEEE Trans. Antennas Propag.*, vol. 70, no. 6, pp. 4112–4126, Jun. 2022.

[35] A. Bharti, F.-X. Briol, and T. Pedersen, "A general method for calibrating stochastic radio channel models with kernels," *IEEE Trans. Antennas Propag.*, vol. 70, no. 6, pp. 3986–4001, Jun. 2022.

[36] A. Bharti, R. Adeogun, and T. Pedersen, "Learning parameters of stochastic radio channel models from summaries," *IEEE Open J. Antennas Propag.*, vol. 1, pp. 175–188, 2020.

[37] P. Zhang, C. Yi, B. Yang, H. Wang, C. Oestges, and X. You, "Predictive modeling of millimeter-wave vegetation-scattering effect using hybrid physics-based and data-driven approach," *IEEE Trans. Antennas Propag.*, vol. 70, no. 6, pp. 4056–4068, Jun. 2022.

[38] X. Zhao et al., "Semi-deterministic dynamic millimeter-wave channel modeling based on an optimal neural network approach," *IEEE Trans. Antennas Propag.*, vol. 70, no. 6, pp. 4082–4095, Jun. 2022.

[39] Y.-J. Wang, L.-X. Guo, Z.-Y. Liu, and X.-J. Li, "Ray-tracing-based indoor channel prediction using point cloud geometrical modeling," in *Proc. IEEE 2021 13th Int. Symp. Antennas, Propag. EM Theory*, 2021, pp. 1–3.

[40] H. Mi et al., "Cluster association for 3D environment based on 60GHz indoor channel measurements," in *Proc. IEEE 2023 17th Eur. Conf. Antennas Propag.*, 2023, pp. 1–5.

[41] J. Pascual-Garcia, J.-M. Molina-Garcia-Pardo, M.-T. Martinez-Ingles, J.-V. Rodriguez, and L. Juan-Llacer, "Wireless channel simulation using geometrical models extrated from point clouds," in *Proc. IEEE 2018 Int. Symp. Antennas Propag. USNC/URSI Nat. Radio Sci. Meeting*, 2018, pp. 83–84.

[42] P. Koivumäki and K. Haneda, "Point cloud ray-launching simulations of indoor multipath channels at 60GHz," in *Proc. IEEE 2022 33rd Annu. Int. Symp. Pers., Indoor Mobile Radio Commun.*, 2022, pp. 01–07.

[43] K. Saito, N. Keerativoranan, and J.-I. Takada, "Dynamic propagation simulation method from LiDAR point cloud data for smart office scenario," in *Proc. IEEE 2022 33rd Annu. Int. Symp. Pers., Indoor Mobile Radio Commun.*, 2022, pp. 1–6.

[44] K. Saito, C. Kang, and J.-I. Takada, "Angular characteristics prediction of radio propagation channel from point cloud data by aperture field integration method," in *Proc. IEEE 2022 16th Eur. Conf. Antennas Propag.*, 2022, pp. 1–5.

[45] S. Ohta, T. Nishio, R. Kudo, and K. Takahashi, "Millimeter-wave received power prediction using point cloud data and supervised learning," in *Proc. IEEE 2022 95th Veh. Technol. Conf.*, 2022, pp. 1–5.

[46] "OS0 Ouster." [Online]. Available: https://ouster.com/products/scanning-lidar/os0-sensor/

[47] "NCTech." [Online]. Available: https://www.nctechimaging.com/capture/istar-pulsar/

[48] R. Sun et al., "Design and calibration of a double-directional 60GHz channel sounder for multipath component tracking," in *Proc. IEEE 2017 11th Eur. Conf. Antennas Propag.*, 2017, pp. 3336–3340.

[49] S. Blandino, J. Senic, C. Gentile, D. Caudill, J. Chuang, and A. Kayani, "Markov multi-beamtracking on 60GHz mobile channel measurements," *IEEE Open J. Veh. Technol.*, vol. 3, pp. 26–39, 2022.

[50] K. Hausmair, K. Witrisal, P. Meissner, C. Steiner, and G. Kail, "SAGE algorithm for UWB channel parameter estimation," in *Proc. COST 2100 Manage. Committee Meeting*, 2010, pp. 1–7.

[51] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE 2017 Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 77–85.

[52] A. Paszke et al., "Automatic differentiation in PyTorch," in *Proc. Neural Inf. Process. Syst. Workshop Autodiff Submission*, Oct. 2017.

[53] 3rd Generation Partnership Project (3GPP), "Study on channel model for frequencies from 0.5 to 100GHz," 3GPP, Sophia Antipolis, France, Tech. Report 38.901, Nov. 2020.