Argus: Predictable Millimeter-Wave Picocells with Vision and Learning Augmentation

HEM REGMI, University of South Carolina, USA
SANJIB SUR, University of South Carolina, USA

We propose *Argus*, a system to enable millimeter-wave (mmWave) deployers to quickly complete site-surveys without sacrificing the accuracy and effectiveness of thorough network deployment surveys. *Argus* first models the mmWave reflection profile of an environment, considering dominant reflectors, and then use this model to find locations that maximize the usability of the reflectors. The key component in *Argus* is an efficient machine learning model that can map the visual data to the mmWave signal reflections of an environment and can accurately predict mmWave signal profile at any unobserved locations. It allows *Argus* to find the best picocell locations to provide maximum coverage and also lets users self-localize accurately anywhere in the environment. Furthermore, *Argus* allows mmWave picocells to predict device's orientation accurately and enables object tagging and retrieval for VR/AR applications. Currently, we implement and test *Argus* on two different buildings consisting of multiple different indoor environments. However, the generalization capability of *Argus* can easily update the model for unseen environments, and thus, *Argus* can be deployed to any indoor environment with little or no model fine-tuning.

CCS Concepts: • **Networks** → **Network management**; • **Computing methodologies** → **Neural networks**.

Additional Key Words and Phrases: Millimeter-Wave; Picocells; Convolutional Neural Network; Transfer Learning

ACM Reference Format:
Hem Regmi and Sanjib Sur. 2022. Argus: Predictable Millimeter-Wave Picocells with Vision and Learning Augmentation. *Proc. ACM Meas. Anal. Comput. Syst.* 6, 1, Article 2 (March 2022), 26 pages. https://doi.org/10.1145/3508022

1 INTRODUCTION

Millimeter-Wave is the core technology for the new wireless LAN and cellular standards, such as IEEE 802.11ay [1] and 5G NR [2], and the key enabler for many high throughput and ultra-low latency wireless applications. Millimeter-Wave (mmWave) networks offer a substantially higher data rate than the traditional wireless networks, but the communications are limited to Line-Of-Sight (LOS) path and very few reflections in Non-LOS (NLOS) paths [3–10]. So, the network relies on light-weight, short-range, and densely deployed base-stations called "*picocells*," which use electronically steerable beams and communicate on very high frequency, on the order of 10s of GHz, and wide bandwidth. Due to the short wavelength, each picocell can host multiple palm-sized antenna arrays that can create hundreds of beams to serve mobile users. With such capabilities, the picocells and mobile devices can also function as high-precision environment sensors.

Authors' addresses: Hem Regmi, hregmi@email.sc.edu, University of South Carolina, USA; Sanjib Sur, sur@cse.sc.edu, University of South Carolina, USA.

2476-1249/2022/3-ART2 $15.00
https://doi.org/10.1145/3508022

However, the short wavelength, high signal attenuation, and environmental obstructions of mmWave links often yield unavailability or misalignment of the paths, which makes the performance of the picocells unpredictable [11–13]. Picocells can electronically steer/adapt beams to track their paths and coordinate among neighboring picocells to enable robust connectivity. But the effectiveness of coordination and adaptation depends on whether the neighbors can support reliable connectivity since their links are also sensitive to the environmental structure [14–16]. While it is not always feasible to transform the environment to aid the picocells (*e.g.*, by adding more reflectors), a network deployer can place the picocells smartly to improve the NLOS paths availability and thereby improve the predictability of mmWave links. Full site surveys may achieve this goal by war-driving a mmWave transceiver (co-located transmitter and receiver) and measuring the *Signal Reflection Profile* (SRP) from every *nook and cranny* [17–22], but they are costly and time-consuming [23, 24]. Ray propagation-based simulators may reduce the cost and time [25–28], but they are frequency-specific since the NLOS signal reflectivity is frequency-dependent. So, it is either costly or challenging to identify the mmWave SRP in a given environment.

We propose Argus, which explores a low-cost, visual data and deep learning based approach to predict the SRPs in indoor mmWave picocell networks.[1] Prior approach based on channel sparsity and geometrical propagation aimed to predict the reflection profiles in 60 GHz networks [15, 29], but the design has been tested and validated only in a single indoor environment. However, the extreme density of mmWave picocells, ultra-wide bandwidth of links, lack of coherency in hardware, and under-explored models of mmWave channels across multiple environments limit the applicability of sparsity or traditional signal processing. On the other hand, visual data can extract higher resolution environmental information, and deep learning can reveal complex models to tackle hard optimization problems. At a high level, *Argus* builds a framework to identify the mmWave SRP in an environment enabling network deployers to quickly and efficiently complete site surveys without sacrificing the accuracy and effectiveness of a thorough deployment survey. Our approach is intuitive: *Argus* identifies deployment locations that maximize a set of picocells' likelihood of having reflection paths; so, the network could be more effective and predictable in a dynamic environment by virtue of not being dependent on only the LOS path. The key idea is to first identify reflectors' properties, *i.e.*, reflectivity, location, and orientation, to model the SRP of an environment, and then use it to find the locations that maximize the usability of the reflectors.

To identify the reflectors' properties, *Argus* combines a visual 3D Point Cloud Data (PCD) and a very few mmWave SRPs measured sparsely inside the environment (see Figure 1) to build a deep learning model. Intuitively, *visually similar objects likely produce similar reflections; so, the learned model could predict the signal*

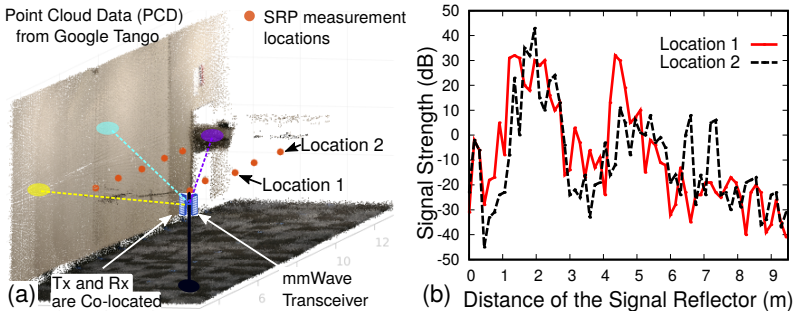

Figure 1. (a) Visual Point Cloud Data (PCD) of an indoor environment, and a mmWave transceiver that measures the reflection profiles. (b) Signal Reflection Profile (SRP) from two locations in the environment.

[1] *Argus* was the Roman god of surveillance and watch, and the great vision and wisdom of *Argus* is analogous to our proposed model with visual data and deep learning.

reflection patterns from any other viewpoint within the environment, even if the deployer has not measured them. However, training the model with the entire PCD will unlikely work since it will learn the noise from random pixel colors and distances. Besides, there are only a few objects in the environment that contribute strongly to the mmWave SRP [15]. So, *Argus* extracts the objects from the PCD within a limited *Field of View* (FoV) and uses prior knowledge of object labels as training input. The trained model can then be transferred to other environments which have similar structures, such as walls, ceilings, beams, columns, floors, *etc.*, with little to no fine-tuning. Besides identifying effective deployment locations, *Argus* can also use the predicted SRPs to enable several applications: (a) Identifying user's location or aiding robot navigation based on the pre-characterized SRPs; (b) Classifying or tagging objects under low-light conditions in VR/AR; (c) Adapting picocell's data transmission rate for different users; and (d) Uncovering "*signal holes*" in the environment and facilitate mounting intelligent surfaces on the walls to improve the SRP distributions [30].

We implement and evaluate *Argus* by building a custom platform for data collection. The setup uses an ASUS Zenfone AR smartphone [31] to collect the PCD and poses of the device and a co-located 24 GHz mmWave transceiver [32] to collect the SRPs (see Figure 8). Since it is hard to trigger the mmWave transceiver and smartphone at the same time due to various software-level delays, *Argus* post-processes the SRP and visual data in software to achieve synchronization. Our experiments across 16 indoor environments in two buildings over a period of 5 months, with 11 GB of data (~ 1.1 million samples), show that by re-training *Argus* for individual environments, it can predict the SRP with a median error of 1.5 dB and 90^{th} percentile error of 4.2 dB only with a base learning model. But the base model, which only considers the distance of the reflecting objects from the transceiver, fails to generalize over other environments. When transferred and tested in untrained environments, the median error is close to 12 dB, and the 90^{th} percentile error could be up to 35 dB! Fortunately, by incorporating the prior knowledge of the environment during training, *Argus* is able to contain the error to only 6.2 dB on the median. Furthermore, by predicting only important points of SRP, *Argus* is able to identify the SRP with a median error of 4.5 dB for completely unseen environments. For picocell deployment, *Argus* is able to reduce the link outage probability in multiple environments by almost 1.55× compared to random and common-sense deployment strategies. For localization and orientation, *Argus*'s predication errors are less than 35 cm and 1.7°, respectively, on all axes for 90^{th} percentile of measurements in diverse environments. For object tagging, *Argus* can classify objects and retrieve them with more than 98% accuracy.

In summary, we have the following contributions: (1) We design a framework for visual data and deep learning augmented mmWave signal reflection profile prediction. It includes the semantic understanding of the environment to make the model robust and effective across multiple environments. To the best of our knowledge, *Argus* is the first system to enable such accurate prediction for practical mmWave picocells. (2) We design and evaluate methods for picocell deployment, device's location and orientation prediction, and object classification for VR/AR applications under poor visibility. Our results demonstrate that *Argus* generalizes well across diverse environments, and it enables reliable and versatile mmWave networks and applications. To catalyze the vision and learning augmented mmWave picocell networking and applications research, we will open-source the measured dataset and deep learning implementation through our project repository [33].

2 MOTIVATION, BACKGROUND, AND CHALLENGES

Motivation and Problem Statement: Effective deployment of picocells for better mmWave indoor wireless networks is necessary for reliable, robust connectivity. Existing manual wardriving techniques to deploy picocells are time-consuming and often infeasible because Signal Reflection Profile (SRP) changes along with location and orientation of the user, requiring deployers to collect

SRP at every possible location and device orientation in the environment. Besides, a change in the environment, such as installation of new large furniture, changing the design and layout of couch, tables, chairs, *etc.*, might require another exhaustive SRP collection. *Argus* aims to reduce the time for effective picocells deployment by exploring a visual data and deep learning based approach to predict the SRPs in indoor mmWave picocell networks; the predicted SRP, in turn, allows the deployer to find effective picocells deployment locations for better indoor mmWave connectivity.

Picocell Fundamentals and Signal Reflection Profile: MmWave networks rely on picocells to provide indoor wireless connectivity within a range of 10-12 m only; this is due to the use of low transmission power and the lack of LOS path to the user for the majority of the time [1]. So, the links rely mostly on the NLOS paths, and thus, identifying the SRPs, determined by the available reflectors, is of utmost importance. To estimate the SRP, a transceiver sends a wireless signal from its transmitter at multiple frequencies towards a scene and receives the reflected time-delayed signals back at its receiver from objects at different distances. In *Argus*, we formulate this as the ranging problem where a transceiver at $(x_r, y_r, 0)$ sends a wide bandwidth signal $p(t)$ towards a scene with K reflecting points with variable reflectivities, each at coordinate (x_k, y_k, z_k) with reflectivity σ_k. Then, the reflected signal received at $(x_r, y_r, 0)$ is the sum of all time-delayed signals, represented as $s(x_r, y_r, t) = \sum_{k \in K} \sigma_k \cdot p[t - 2d_k/c]$, where c is the wireless propagation speed ($\sim 3{\times}10^8$ m/s), and $2d_k$ is the round-trip distance between the k^{th} reflecting point and the measurement location. After applying the Fast Fourier Transform (FFT) on the received signal, we get the SRP, which shows the signal intensity at different distances [34]. Figures 1(a–b) show an example indoor environment and the corresponding SRPs from two locations. Note that the SRP shows a 1D distance to the signal reflectors, even if the target scene is in 3D, and the SRP changes significantly depending on the measurement location. In *Argus*, instead of measuring SRP from every location within the environment, we propose a learning based approach that predicts SRPs based on a few measurements.

Challenges with SRP Prediction: SRP prediction for mmWave picocell networks is challenging because of the lack of proper modeling and availability of open-source datasets at very high wireless frequencies. Besides, different objects interact differently with the mmWave frequencies, and their reflection strengths are not only influenced by the distance to the transceiver but also the orientation and object materials [4, 6, 8, 35–37]. There exist multiple wireless propagation simulators [25–28] for the low-frequency networks, such as Wi-Fi and LTE, however, their approaches are frequency-specific since the signal reflectivity is frequency-dependent. So, they will still require actively measuring reflections from various objects to estimate the SRPs. Besides, mmWave signals are highly specular due to their small wavelength, *i.e.*, many objects introduce mirror-like reflections [5, 10]. So, even some strong signal reflections may not arrive at the receiver due to angle mismatch. Moreover, the mmWave signal strength depends on only a few strong reflectors, such as walls, ceilings, floors, *etc.* Detecting these items and isolating their contributions manually also poses a problem. Thus, in *Argus*, we leverage the combination of 3D visual data and SRPs measured at a few locations to build a deep learning model that can predict the SRPs of the entire environment.

3 *ARGUS* DESIGN

3.1 Overview

Argus facilitates an effective way of SRP prediction in an environment to allow the network deployers place picocells optimally and achieve predictable performance. Many applications could also leverage these SRPs to enable various services, *e.g.*, localizing devices or tagging objects. To predict the SRPs, *first*, a deployer uses an AR device, such as Google Tango or ARCore [38, 39], to quickly

create a visual map of the environment by walking around (Figure 1[a]). *Second*, as the deployer is walking around, a co-located mmWave transceiver measures the SRPs by steering the mmWave beam rapidly (Figure 1[b]). *Finally*, *Argus* leverages the visual data and SRPs to learn a *mapping* between objects and their reflections, which allows predicting the full SRPs of the environment. To this end, *Argus* uses a Convolutional Neural Network that maps depth and object types to the reflections through a supervised learning. Figure 2 shows an overview of the *Argus* system.

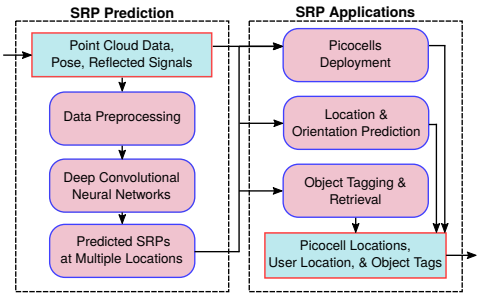

Figure 2. System overview of *Argus*.

The visual map of the environment, stored as the PCD, is first synchronized with the measured SRPs collected at random locations during the measurement. Since the FoV of the mmWave transceiver is smaller than the RGB-D camera, the PCDs are processed to generate a Local PCD (LPCD) by removing the data points falling outside the transceiver's FoV. These LPCDs are then projected onto a 2D plane, masked with the transceiver's beam pattern, and paired with the measured SRP and device's pose, to train a SRP prediction network. The network, from thousands of example data-pairs, learns the association between the visual data and the measured SRP, and then, can predict SRPs at various locations within the environment, even if it has not been trained before for those locations. This model is also generalizable to predict the SRPs for other environments with similar objects and structures. Finally, these predicted SRPs are used for identifying picocell locations for predictable performance and for various applications, including localizing devices and tagging objects during poor optical visibility. We now describe these design components in detail.

3.2 Visual Data to SRP Relationship

Before building the model, *Argus* first analyzes the relationship between the visual data to the SRPs: To explore the validity of the hypothesis that, *visually similar objects likely produce similar mmWave reflections*, and to identify a suitable model to capture such relationship. *First*, we segment the visual data into different semantic labels, such as wall, ceiling, floor, *etc.*, using the exisitng indoor PCD segmentation technique [40]. *Second*, we group the regions based on their semantic labels, and select the regions that have a single label in majority of the area. *Finally*, we compute the visual similarity by calculating the Structural Similarity Index Measure (SSIM) [41] between the projected 2D depth images obtained from visual data. The SSIM is between 0 and 1, where 1 represents the perfect similarity. Besides, we find the similarity between two SRPs by calculating the Mean Squared Error (MSE) between them, where lower MSE value represents a closer match. We compute the SSIM and MSE between data pairs across 16 different environments from two different buildings to test our hypothesis (see Table 1 for the environments' dimensions and properties).

Figure 3 shows the relationships between the SRPs and visual data through scatter plots, where each point represents the SSIM and MSE pair between two projected images and corresponding SRPs, respectively. Each scatter plot consists of 20,000 pairs of SSIM and MSE values for each environment, so Figure 3 represents 320,000 pairs. While we observe that there is a general trend in the relationship, it is hard to capture it using a straightforward model which work across various environments. For example, the relationships in Env A.6, Env B.1, and Env B.4 show somewhat simpler non-linearity between the visual data and SRP; but all other environments show the complex non-linear correlation, which is difficult to model mathematically. To this end, *Argus* aims to learn the relationship between the visual data and SRPs using a non-linear, data-driven model.

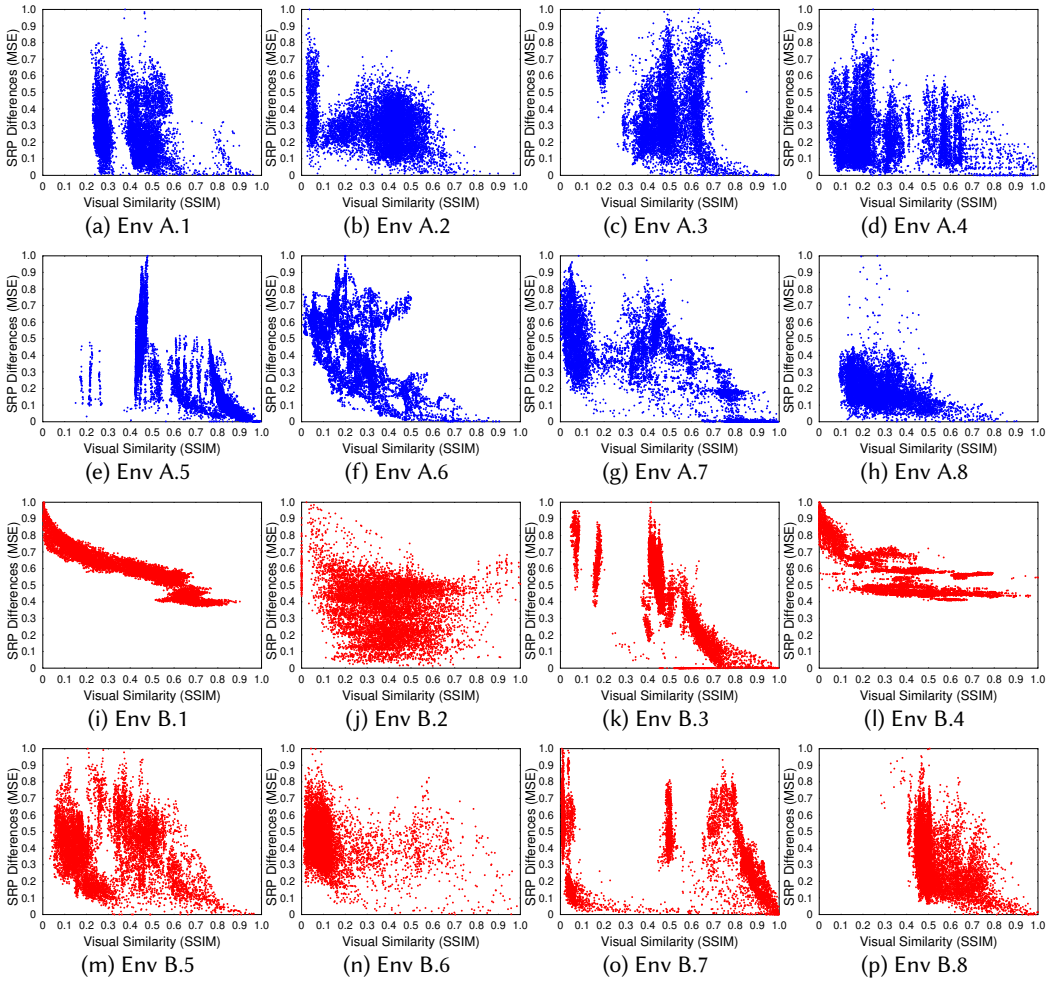

Figure 3. Relationships between SRPs and visual data across multiple environments in two buildings.

3.3 Signal Reflection Profile Prediction

Argus proposes to learn the mapping between the visual data and SRPs, enabling network deployers to quickly identify the full SRPs of an environment without sacrificing the accuracy and effectiveness of thorough site surveys. The key idea is to first identify reflectors' properties, *i.e.*, reflectivity, location, and orientation, and then use the properties to model the SRPs of the environment.

► **Problem Statement**: To be more concrete, let us formalize the problem. Consider that we have N SRPs $\{S_1, S_2, \ldots, S_N\}$ measured in an environment with K strong reflectors. The reflectivity, location, and orientation of the reflectors can be characterized by a vector, $p_k = \{R_k, L_k, O_k\}$. Suppose dT_j, dR_j denote the distance between the reflectors and j^{th} transceiver's location. Then, we can find the reflectors' number and properties (K^*, p_k^*) based on this optimization function:

$$(K^*, p_k^*) = argmin \sum_{j=1}^{N} \left\| S_j - M_K(p_k, dT_j, dR_j) \right\|^2 \quad (1)$$

Here, $||\cdot||^2$ is the L^2-norm, and $M_K(\cdot)$ can use the Saleh-Valenzuela method to model the relationship between the reflectors and the measurement locations [14, 42]. *But the optimization problem is hard*

for two reasons: (1) The minimization is non-convex and often underdetermined, and (2) $M_K(\cdot)$ relies on a geometrical channel model at mmWave that may not generalize to many indoor environments [42]. So, instead of relying on a fixed model, we propose a deep learning based approach.

3.3.1 Deep Convolutional Neural Network for SRP Prediction. *Argus* uses a Deep Convolutional Neural Network (DCNN) to learn the mapping between depth and object types to the reflections through supervised learning. The problem of predicting environmental structures and signal strength is similar in spirit to the image super-resolution [43–47], where the measured reflections are similar to low-resolution images captured from a few vantage points, which can be fed to a DCNN with locations as labels to reconstruct high-resolution images. So, from the 3D visual data, we project a 2D depth image from a viewpoint where the reflections were collected and then feed the reflections as the ground truth for training the DCNN. We further amend the network with antenna pose since SRP is also affected by the way the deployer holds the device, and we use the MSE between the predicted and ground truth SRPs as the DCNN loss function.

► **Data Preprocessing**: To prime the measured dataset for training, *Argus* first preprocesses them to filter out the noise and spurious information (see Figure 4). Recall that a deployer uses an AR device, coupled with a mmWave transceiver, to quickly gather the visual map of the environment, which is stored as the PCD (Section 3.1). As the deployer is walking around, the transceiver captures thousands of back-to-back mmWave SRPs from different poses. Hence, a single scan consists of an entire PCD with all the reflective objects and thousands of poses and SRPs. Asking the network to learn from the entire PCD will not only introduce randomness and noise but also may affect the network convergence. Besides, the transceiver has a limited FoV and specific signal emission (beam) patterns [48, 49], which suggests that at any pose, it only receives reflections from a fraction of the environment. Also, the mmWave signal strength decays steeply with distance [50], and thus, objects from a far distance will unlikely contribute to the measured SRPs.

Therefore, for each pose, we first obtain a Local PCD (LPCD) by removing all objects which are outside the transceiver's FoV and at distances greater than d_{max}. Then, the 3D locations of the LPCD are projected onto the XY-plane to obtain the 2D depth image, where the intensity represents the depth value in meters. We consider a wider visual FoV to ensure *Argus* does not miss any reflections that contribute to the SRP. We then get the Inverse Depth Image (IDI) to explicitly instruct DCNN to learn that closer objects provide stronger reflections. IDI is the pixel-wise inversion of depth value. For example, if a pixel value in the depth image is 0.5 (m), *i.e.*, the point is 0.5 m away from the mmWave transceiver, then the pixel value in the IDI is (1/0.5) = 2. We select the FoV (θ) [2] slightly higher than the transceiver's FoV to include not only the beam's main lobe but also side lobes [52]. We then include the beam pattern of mmWave transceiver [53], which is a 2D matrix with normalized transmit and receive power at different azimuth and elevation angles, to mask (dot multiplication) the IDI and obtain a *Masked Inverse Depth Image* (MIDI). MIDI not only embeds the

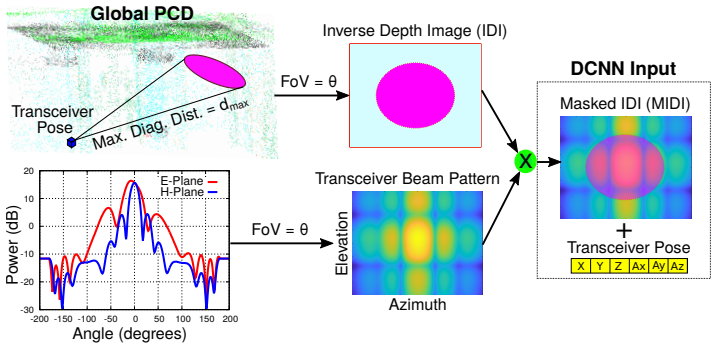

Figure 4. *Argus*'s data preprocessing to generate input for its DCNN.

[2]In *Argus* implementation, we use $d_{max} \sim 10m$ and $\theta \sim 120°$ to match realistic mmWave network and device properties [51]

information of the reflective objects but also the properties of the mmWave transceiver. Finally, MIDI is combined with the corresponding transceiver's pose and SRP to form an input and output pair for the DCNN.

► **Base DCNN Model**: While data preprocessing allows keeping only important features, finding an optimal model to map visual data to SRP is non-trivial due to the complex non-linear relationship (see Figure 3). Machine learning models have been widely used to build such non-linear models by extracting hidden features, and recent advancements in deep learning for wireless [54–56] motivate us to build a suitable model for *Argus*. Since SRP is a vector of signal strengths, we build a DCNN that takes MIDI and pose as inputs and generates SRP at the output. *Argus* can use multiple Convolutional layers [57] to extract features from MIDI, which has an image-like structures, and pass the extracted features to the Fully Connected (FC) layers [58] to predict the SRP. However, DCNN with multiple Convolution and FC layers will have millions of parameters and impose memory and computational constraint in *Argus* [59]. So, we propose to select a model that is not only accurate but also has lower computational and memory requirements.

To find the DCNN model with smaller memory and training footprint and better SRP prediction accuracy, we empirically evaluate different popular deep learning models as Convolution layers: *VGG16* [60], *ResNet152V2* [61], *InceptionV3* [62], *DenseNet121* [63], *MobileNet* [64], and *MobileNetV2* [65], and ranked them based on their MSE loss on test data samples. We select *MobileNetV2* as the Convolution layer in *Argus* because it shows a similar MSE loss as the other models, but has the smallest number of training parameters, smallest memory requirement (14 MB), and fastest training time. *MobileNetV2* uses the depthwise separable convolutions, which allows the full convolution layers to be replaced by a combination of lightweight filters and pointwise convolution, significantly reducing the computational cost by a factor of $\mathbb{F}^2$, where $\mathbb{F}$ is the filter size. Besides, *MobileNetV2* uses the inverted residual blocks instead of normal residual blocks; so it requires less number of tensors (compared to normal residual blocks) during implementation to store the intermediate convolution results, and thus, it is more memory efficient [65]. We customize the traditional *MobileNetV2* to work with *Argus* since the standard input type is 3-channel 224×224 RGB image, but the MIDI in *Argus* is a single channel 122×122 monochrome image. So, we concatenate MIDI in the channel dimension to generate 122×122×3 and supply it to the *MobileNetV2*. However, *MobileNetV2* is designed and trained on ImageNet dataset [66] for image classification tasks, but *Argus*'s goal is to predict SRP, which is a regression task. Thus, we trim *MobileNetV2* up to its convolution layers to extract features only and feed them to customized FC layers for regression. In addition to the MIDI, we also provide the pose in the second last FC layer as complementary information because the SRP also depends on the transceivers' location and orientation (*i.e.*, pose): MmWave signals are sensitive to the device's location and orientation since both properties affect how signals hit the reflectors and bounce them off of them. Hence, adding pose directly into the DCNN FC layer helps the network to generalize on robust settings. Figure 5 shows the architecture of the base DCNN model in *Argus*.

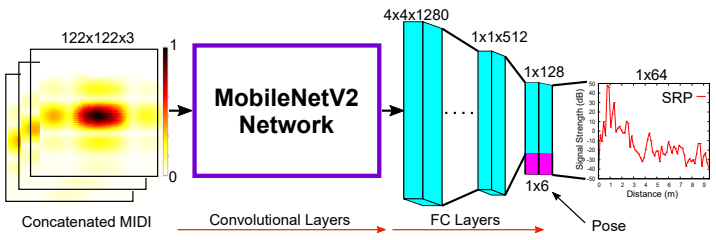

Figure 5. Base DCNN model of *Argus*.

3.3.2 *Transfer Learning via Semantic-Aware DCNN.*

The base DCNN model allows *Argus* to learn the mapping between the MIDI and pose to SRP. Although the base model performs well for

individual environments, it is not transferable to another similar-looking environment due to the lack of semantic information, like the object types. In indoor settings, many structural components are similar among multiple environments. For example, the hallway space on the first floor of building A (Table 1) consists of large structures, such as floor, wall, and ceiling made of certain materials, which remains similar in the second floor hallway too. Besides, many other environments in the same building also consist of similar large structures made of similar materials. However, we observe that the base model performs poorly when trained on the first floor hallway space and tested on the second floor hallway space. This is because the base model fails to incorporate the semantic structure information and thus, lacks a generalization capability. Thus, *Argus* proposes to augment the base model for more general-purpose, transferable model training.

Figure 6 shows the semantic-aware DCNN model in *Argus*, which designs a two-way feature extraction pipeline: (1) The local feature extraction using the base DCNN model and (2) The global feature extraction using a semantic segmentation architecture [67]. The base DCNN model allows *Argus* to efficiently extract the local features from MIDI that are within the environment and are contributing to the SRP. In addition, the semantic segmentation allows *Argus* to learn the high-level global features to transfer the model across multiple environments. Here, we first select all the points within the transceiver's FoV from a particular pose to find the LPCD and pass it to the semantic segmentation to get the corresponding semantic labels [67]. We follow [40] to label each point into one of the following common items found in the indoor environment: *ceiling*, *floor*, *wall*, *beam*, *column*, *window*, *door*, *table*, *chair*, *sofa*, *bookcase*, *board*, and *clutter*. Semantically segmented labels are then used to construct the point features vector, where it concatenates the location, RGB color, and one-hot encoding of semantic labels [68] to form the point features vector; [X, Y, Z, R, G, B, 1×13 one-hot coded class]. Thus, the size of point features vector is [Number of Points × 19].

Each point on this point feature vector then goes through multiple 2D convolution and pooling layers of PointNet architecture [67] to extract the global features. Global features carry the properties of the whole environment to represent it. Since the similar looking environments will likely generate matching global features, irrespective of their location and building, they help *Argus* to find the association between multiple environments. Finally, we concatenate the local features from the base model and the global features from the semantic segmentation and pass them into the FC layers to predict the SRP. During concatenation, we provide higher weightage to the local features by sampling more from there. Since local features are closely related to the SRP values, the higher weightage on them will enforce DCNN's primary goal, *i.e.*, accurate SRP prediction. Furthermore, the base model tries to predict the entire SRP vector and doesn't take into account that there exist only a few strong reflectors in the environment which contribute significantly to the mmWave SRP, and in turn, the network performance [4, 5, 42, 51]. From analyzing different environments, we found that there are only a few peaks in the received signal, which are significant and need to be learned in *Argus*. So, in the semantic-aware DCNN model, we ignore the points in the SRP, which has very low magnitude and doesn't provide any significance in describing the environment in the wireless domain.

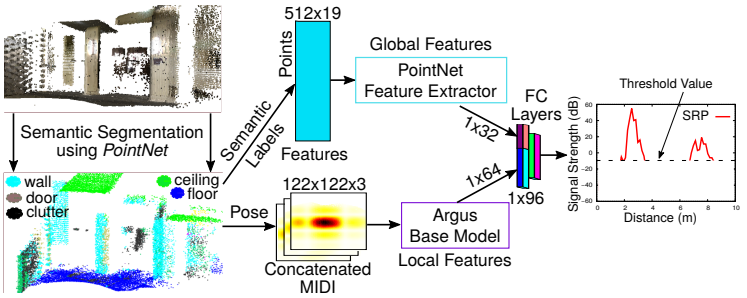

Figure 6. Semantic-aware DCNN model of *Argus*.

► **Network Loss Functions**: Loss functions allow the network to tune the convolution weights appropriately and train themselves. Base DCNN model uses the MSE loss to train the network. However, in the sematic-aware DCNN model, we use a custom loss function to train the model to consider only those reflecting objects that are contributing strongly to the SRP. To design the custom loss function, we ignore the SRP points based on their value to obtain the mask shown in Equation (2). We calculate the threshold value (γ) from the empirical analysis of the dataset with all environments. The mask obtained is used to define a custom loss function, L_C.

$$L_C = \sqrt{\frac{1}{N}\sum_{i=1}^{N} M \times (y_i - t_i)^2}; \qquad where, M = \begin{cases} 1 & \text{if SRP} >= \gamma \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where y_i is the predicted output, t_i is the ground truth, and N is the total data samples used to update the DCNN network in a single batch, *i.e.*, *BatchSize*. The customized loss function helps *Argus* learn the important feature points of the environment that contributes strongly to the SRP.

In sum, the semantic-aware DCNN makes Argus environment-agnostic, i.e., it helps to achieve better performance within a trained environment as well as in other similar-looking untrained environments.

3.4 Applications of SRP Prediction

The ability to predict the mmWave SRP at any location in an environment opens the avenue for multiple opportunities, such as finding the picocell deployment locations to maximize the network capacity and area coverage [69, 70], predicting location and orientation of devices in run-time [71–74], and tagging and retrieving objects for VR/AR applications under poor visibility [75]. Next, we describe the design of these 3 applications using the predicted SRPs.

3.4.1 Picocell Deployment. Since picocells' range is limited, and they are susceptible to obstructions, coordination among neighboring picocells is necessary to achieve predictable performance. However, the effectiveness of coordination depends on the neighbors' reliable connectivity, but their links are also sensitive to the location and environmental structure, and minor location changes can drastically alter the network capacity [15, 16]. Said differently, these picocells need to be deployed in the environment carefully and efficiently to get optimal coverage. [15] predicts the picocells' deployment locations using the Ray-tracing method [76] and dwelling probability of users' location in the environment, but fails to include the actual received signal strength at various locations. In contrast, *Argus* uses the predicted SRPs in the environment to include realistic reflectivity for picocells' locations estimation. But recall that *Argus* leverages a mmWave transceiver to collect the SRPs sparsely to build the DCNN model; but in practical networking, the picocell (transmitter) and the user (receiver) are separately located. Hence, *Argus* implements the Ray-tracing method [76] using picocells' predicted SRP to predict the user's SRP. This, in turn, allows *Argus* to find the picocells' that will likely generate better users' SRPs and higher network performance.

The Ray-tracing method works as follows. *First*, using the picocell's predicted SRP, we virtually extend the wide bandwidth mmWave wireless signal across the environment and towards the user's location. The signal bounces off of different objects to arrive at the receiver. *Second*, we record and add the reflected signals coming back from all objects that are inside the user's FoV to emulate realistic mmWave hardware. Additionally, we only consider those points with matching angles of departure and arrival to simulate the scattering property of mmWave [77]. *Finally*, we apply a 1D FFT on the reflected signals collected over time to obtain the predicted SRP at the user's location. To predict the *Coverage Profile* (CP), *i.e.*, how well a picocell will likely perform at a specific deployment location (see Figure 14 for an example CP), we first divide the 3D environment into a fixed number of voxels, equally spaced on each axis, and measure SRPs at different users' locations by placing a single picocell at one voxel location. Since the actual user's location during

the run-time is unknown, we assume a uniform location distribution. This gives the CP for the specific picocell location. We now repeat this process by virtually placing the picocell at all possible voxel locations and recording all CPs. Then, we rank each picocell at different locations and choose a set of picocells that have the maximum combined CPs.

Algorithm 1 *Argus* picocell placement for an environment

Argus trained model for SRP prediction; Number of available picocells = Q;
for LT_j = 1 to M, Total Tx Locations **do**
  Estimate SRP S_{jk} from *Argus* Ray-tracing; $1 \leq LR_k \leq N$, at all Rx locations for j^{th} Tx location
  Where, $(X_p, Y_q, Z_r) \leftarrow LT_j$; $(X_m, Y_n, Z_o) \leftarrow LR_k$; $M = p * q * r$, $N = m * n * o$;
  Calculate the coverage score $C(x, y, z)$ based on selection criteria:
    "Average," or "Variation," or "Link-outage"
end for
Sort locations LT_1, LT_2, ..., LT_M in descending order based on the coverage score C
Retrieve first Q locations from M possible locations: $(LT_1, LT_2, .., LT_Q) \leftarrow (LT_1{:}LT_M), (Q << M)$

Algorithm 1 shows the steps to find the picocell locations in *Argus*. *Argus* uses the user's SRP to calculate picocell locations. These user's SRP implements Ray-tracing method using picocells' predicted SRP from the deep learning model. In addition, *Argus* allows the deployer to select between the "average," "variation," or "link-outage" method on CPs for picocell ranking: (1) Ranking based on high CP average allows better mean network throughput; (2) Ranking based on low CP variation allows performance fairness across users; (3) Ranking based on the smallest link-outage reduces the fraction of area with no network coverage. For the average strategy, we compute the mean of all SRPs for a CP, while variation computes the variance of SRPs. In the link-outage, we select a signal strength threshold T_{SRP} and measure the % of area above that threshold. Finally, we select the top Q locations from total M ($Q << M$) possible picocell locations.

3.4.2 Localization and Orientation Prediction. We now employ the *Argus* model and measured SRPs to predict the location and orientation of the user during run-time. While there exist many localization techniques based on Wi-Fi, IMUs, vision, deep learning, *etc.* [78–82], we show that the predicted SRPs and *Argus* model could seamlessly upgrade the mmWave picocells to provide accurate location and orientation estimation. The localization model includes the base DCNN model as the feature extractor and FC layers with an appropriate number of neurons for regression. Figure 7 shows the network for location and orientation prediction, which takes the MIDI and SRP in the input layers and predicts 3D pose at the output layer. We train the network with thousands of samples of *<PCD, SRP, Pose>* pairs to learn input to output mapping for the given environment and use MSE loss to tune the network parameters. During the run-time, a device can then measure the SRP, and *Argus* can predict its pose to enable multiple location-based applications.

3.4.3 Object Tagging and Retrieval. The predicted SRPs could also be used to tag and retrieve nearby objects for applications, such as VR/AR. For example, Overlay [83] uses the annotated images and database to build an AR application that provides the guidance for visitors, but requires a large amount of visual datasets and fails under poor or no light conditions. *Argus* uses the previously collected data samples for classifying the objects in the environment with DCNN model. Figure 7 shows the network for object tagging and retrieval, and the network shares convolution structures with the localization and orientation prediction module. To get data for training and testing,

we locate N different objects at various locations in the environment and label them to corresponding classes. We create thousands of *<MIDI, SRP, Object Category>* pairs, and shuffle and feed the samples for training a Deep Classification Network (DCN). DCN is trained with categorical cross-entropy loss [84], defined as, $\mathbf{L}_{DCN} = -\sum_{i=1}^{N} t_i log(c(s_i))$, where $c(s_i)$ and t_i are the predicted and actual probabilities of i^{th} class. Post training, we pass the SRPs to get the object classes as probabilities. The category with the highest probability will be classified as the predicted class. This allows *Argus* to classify and tag objects in the environment with the help of SRPs even under poor lighting conditions.

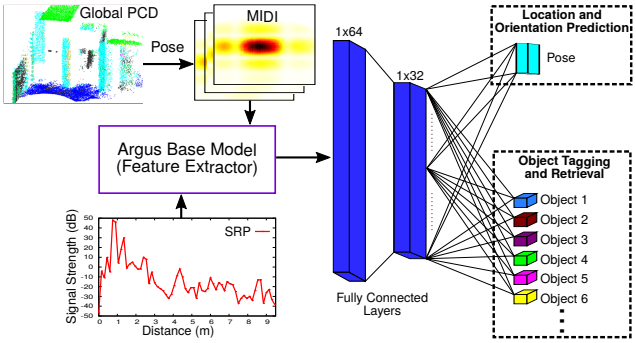

Figure 7. Pose prediction and object classificaton network.

4 IMPLEMENTATION AND EXPERIMENTAL SETUP

Hardware Platform: Due to the unavailability of AR functionality and lack of access to the SRPs in current 5G mmWave devices, we implement and evaluate *Argus* using real data collected from our custom-made platform with a 24 GHz mmWave transceiver [32] and a Google Tango device, ASUS Zenfone AR [31] (see Figure 8[a]). The mmWave transceiver is equipped with 24 transmit and 24 receive phased-array antennas, each arranged in 4×6 format, and can generate different beam patterns. It can collect SRPs as data frames in real-time, with a sampling rate of ~ 67 milliseconds (ms) per frame, and is connected to a laptop via a USB cable for storing the data. The transceiver operates on a 1 GHz bandwidth at the center carrier frequency of 24.625 GHz. We use the following parameters for SRP measurement: Start frequency, 24.125 GHz; frequency ramp slope, 1.43 MHz/μS; number of complex ADC samples, 256; ADC sampling rate, 487 Msps; sweep duration, 0.70 ms; pulse repetition rate, 15 Hz; and maximum receive antenna gain, 56 dBi. The measured SRP at a particular pose is a 256 element vector, which corresponds to the number of ADC samples. Since the transceiver operates at 1 GHz bandwidth, each SRP element has a resolution of ~ 0.1499 m [85]; hence, the device can gather reflections up to ~ 38.40 m. However, the reflection strength at such a long distance is below the noise floor, and the picocells are supposed to operate within 10s of m in indoor settings (Section 2); so, we limit the maximum range to ~ 10 m, which corresponds to the first 64 elements. The distance between the transceiver and AR device is kept fixed during all the data collection, and we use the known distance offset to calibrate the transceiver's pose. We implement *Argus* in Matlab and Python environments running on a host PC, which uses the 3D PCD and a few mmWave SPRs as input and generates full SRPs of the environment as output.

Real Data Collection: Since a real-time, tight synchronization between the transceiver and AR device is currently unavailable, we post-process the collected dataset in software to achieve the synchronization. A Matlab program running in the host PC first starts the AR device to collect the visual data using the RTAB-SLAM app [86] and waits for scene stabilization, while the transceiver starts recording the reflected signals. After the scene is stabilized, we move the setup around to construct the visual map of the environment and gather SRPs from various poses. A single scan of ~ 4 min. can generate a rough PCD of a typical indoor environment and can gather reflected signals from around 3700 unique transceiver poses. We then apply 1D FFT on the reflected signals to obtain the SRPs. Since the AR device takes ~ 10 - 15 secs for scene stabilization and has a lower sampling

Table 1. Properties of 16 environments in two buildings.

Environments	L (m)×W (m)×(H (m)	Purpose of Usage	Wall Material	Type of Clutters
A.1	9.4 × 6.1 × 2.9	Lobby area to enter building	Drywall	TV mointors, glass doors, metal stripes, benches
A.2	14.4 × 3.1 × 2.9	Space near first floor elevator	Drywall	Paintings in wall, glass doors, elevator doors
A.3	10.4 × 4.3 × 3.1	Second lobby area	Drywall	Wooden benches, glass doors, metal stripes
A.4	37.4 × 2.0 × 2.9	First floor long hallway	Drywall	Wooden doors for offices, glass regions on wall
A.5	11.3 × 3.2 × 3.0	Space near second floor elevator	Drywall	Glass doors, elevator doors
A.6	45.9 × 1.9 × 2.7	Second floor long hallway	Drywall	Wooden doors, wall with glass regions
A.7	28.1 × 1.9 × 2.7	Short hallway	Drywall	Wooden doors, wall with glass regions
A.8	11.5 × 6.0 × 3.0	Large office	Drywall	Computer screens, chairs, couch, wooden door
B.1	48.5 × 2.4 × 2.4	General hallway	Concrete wall	Woodend doors, wall with glass regions
B.2	17.2 × 9.6 × 3.0	Common hangout area	Concrete wall	Chairs, tables, wall paintings
B.3	13.2 × 3.5 × 2.4	Lobby area to enter building	Concrete wall	Wooden doors, glass doors
B.4	7.9 × 7.2 × 3.2	Area near staircases	Concrete wall	Wooden doors, glass doors, stairs, tables, chairs
B.5	17.7 × 2.4 × 2.4	Third floor common hangout area	Concrete wall	Couch, glass doors, notice board, wall paintings
B.6	56.1 × 2.4 × 2.4	Large hallway	Concrete wall	Notice board, wooden door, trash cans
B.7	9.6 × 2.0 × 2.4	Exit area of the building	Concrete wall	Wood doors, glass doors
B.8	7.9 × 7.1 × 2.7	Connecting area near staircases	Concrete wall	Circular hollow in middle, stairs, glass doors

rate than the transceiver, we post-process the pose and SRPs to align them and interpolate the poses to match the SRP sampling rate. The consecutive SRPs will be *similar* to each other during the scene stabilization; so, we can identify the SRP's local timestamp of movement by correlating each SRP with the first SRP. Similarly, we can identify the AR device's local starting timestamp by identifying the self-pose change. Since the transceiver and AR device begin moving at the same time, we can calibrate the local timestamps and obtain the synchronized start of SRP and poses.

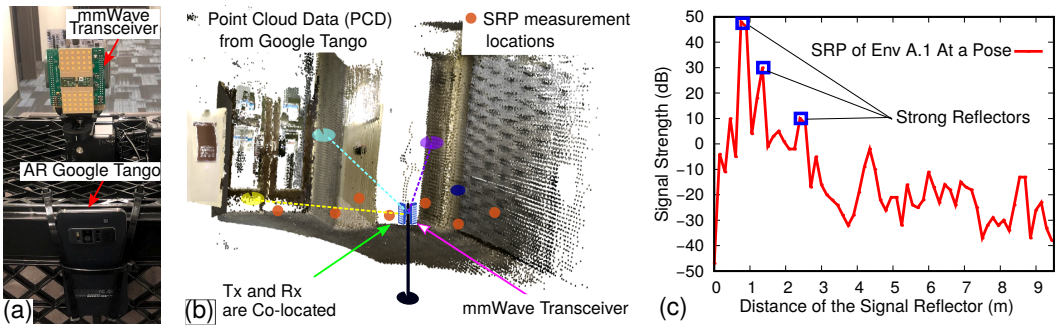

Figure 8. (a) 24 GHz mmWave transceiver and AR device setup. (b) A PCD with strong reflective objects and the mmWave transceiver. (c) Received signal strength (dB) at a pose (blue circle of Figure 8[b]).

Figures 8(b–c) show an example environment and the measured SRP from one of the locations (marked in blue circle). There are three strong reflectors at ∼ 0.7 m, 1.3 m, and 2.4 m from this location, which corresponds to three strong peaks in the SRP. We collect the PCDs, poses, and SRPs from 16 different environments across two distinct buildings (whose wall materials are different) to understand *Argus*'s spatial performance. To compare *Argus* with a full site survey and understand the optimal performance, we scan each environment thoroughly for ∼ 2 hours. But *Argus* uses only the first few minutes of dataset to mimic a quick scan from the deployer. We also repeat this process multiple times over a period of 5 months to understand *Argus*'s temporal performance. These datasets are collected during regular business hours; so, there could be background noise and occasional disruptions. Table 1 summarizes the dimensions and properties for each environment across the two buildings, A and B. Our dataset is diverse, consisting of environments primarily used as walking hallway, near elevator spaces, common hangout areas, offices, *etc.*, from both buildings, and they have different sizes, wall materials, and types of objects. These properties affect the way signals are reflected back to the transceiver. In total, we have collected and analyzed over 11 GB of dataset with ∼ 1.1 million data samples. We have used only ∼ 280 K samples for training, and the rest of ∼ 812 K pairs are used for testing and benchmarking all our design components. *Such data*

diversity and scale allow us to not only evaluate the performance of Argus but also understand its generalizability across multiple, real environments.

Neural Network Training: The SRP prediction model in *Argus* is trained and tested with both the MSE and MAE losses, but we found that MSE loss performs better across all cases. During the training, we allow the network to explore up to 1000 epochs, but we also implement an additional hook so that the training would stop if there is no improvement for 10 consecutive epochs. We explore different optimizers and learning rate for our DCNN models following [87] and observed that the "Adam" optimizer performed the best among all with a learning rate of 0.002; it combines the *AdaGrad* and *RMSProp*, which helps for better model convergence. We follow a similar strategy to train and test location and orientation prediction and object classification models since they also have similar architecture to extract features from MIDI and only vary at the final FC layer. Finally, all the DCNN models are designed and implemented with Python programming language and TensorFlow 2.3 [88] package on host PC with Intel Xeon CPU @ 3.5 GHz, 32 GB RAM with Nvidia's GeForce GTX 1070 GPU [89] to speed up the training process. All of our networks require at least 24 hours to train, but the training time can be reduced significantly by uploading the data to the Cloud TPUs [90] or using more powerful GPUs [91].

5 PERFORMANCE EVALUATION

We evaluate *Argus* using 3 standard metrics commonly used to quantify regression error, classification error, and network link outage.

► **Absolute Error**: Error between the ground truth and predicted values of SRP (in dB), location (in m), and orientation (in °), and it compares the performance with the optimal scheme.

► **Link-Outage Probability**: It represents the probability of falling on the zone without any network service, *i.e.*, SRP values are below a certain threshold value. We use this to rank the picocell's location during deployment. The values are between 0% and 100%.

► **Classification Confusion Matrix**: The probability of classifying objects, where the matrix diagonal represents the probability of correct classification. The values are between 0% and 100%.

Evaluation Summary: (1) *Argus*'s base DCNN model can predict SRP with a median error of 1.5 dB only, but fails to generalize over similar-looking environments. Adding semantic labels reduces the cross-environment SRP prediction error by ~3×; it can also predict SRP with a median error of 1.36 dB when fine-tuned with just 5 mins. of scanning data. *Argus* works well even when the visual datasets are incomplete and sparse, and its performance remains stable over time. (2) *Argus*'s picocell deployment model achieves a near-optimal performance, and its "link-outage" strategy reduces users' link-outage probability by 1.55× compared to the *Random* and *Common-Sense* methods. (3) *Argus* can localize users within 35 cm on median and predict device's orientation with at most 1.7° median error on all axes. Finally, *Argus*'s classification model achieves more than 98% average accuracy on all object classes allowing precise object tagging and retrieval.

5.1 SRP Prediction

Base DCNN Model: To evaluate the effectiveness of *Argus*'s base DCNN model, we use the datasets collected from 4 environments from the first floor of building A: A.1, A.2, A.3, and A.4. We build, train, and test individual base DCNN models for each environment, where the trained network parameters could potentially vary across environments. We preprocess the dataset to generate *<MIDI, Pose, SRP>* pairs, and then randomly select training and testing samples. We then shuffle and sequentially feed all the training samples in the model. After training, we feed MIDI and pose pairs from the test samples to the trained DCNN model to predict their SRP. We then find the absolute

SRP prediction error of the base DCNN model by calculating the difference in the predicted and ground truth SRPs for different distances across 2000 test samples per environment.

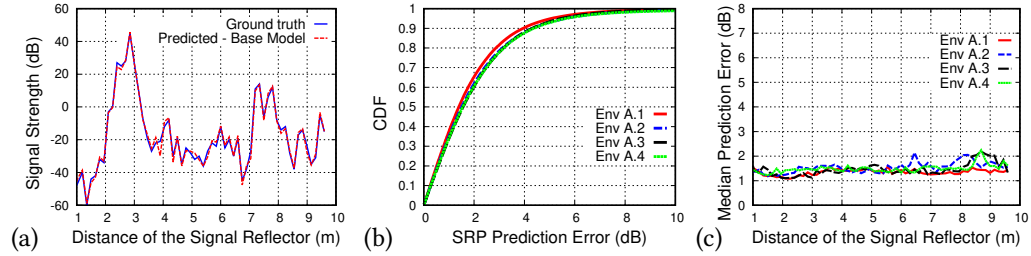

Figure 9. (a) An example of SRP prediction using base model at one location in A.1. (b) CDF of SRP prediction error in 4 environments (2000 samples/environment). (c) Effect of reflector distances on SRP prediction error.

Figure 9(a) shows an example of ground truth and predicted SRP from one of the locations in environment A.1, and clearly, the prediction accurately follows the ground truth. Figure 9(b) further shows the distribution of SRP prediction error across 4 environments: The base models have similar performance, irrespective of the environment. For example, for A.1, the median prediction error is only 1.39 dB and the 90^{th} percentile error is 4.04 dB, which are tolerable for practical networking [1, 92]. Similarly, for A.2, the median and 90^{th} percentile errors are 1.57 dB and 4.34 dB, respectively. Overall, the average of median errors across the 4 environments is 1.5 dB and the standard deviation is 2.03 dB. Figure 9(c) further zooms in on the distribution and sorts the prediction error for different distances. It shows that the base models' performance is unaffected by the position of the reflectors, indicating no biasness up to 10 m. *Therefore, the base model, when tuned and customized for individual environments, performs very well consistently.* But it lacks the generalization capability since it is only trained on object depths and transceiver's poses without the knowledge of true environmental semantics. Figure 10(a) (red line) shows the effect of such poor generalization when it is trained in one set of environments and tested in unseen environments with similar-looking structures. The median and 90^{th} percentile SRP prediction errors are more than 12 dB and 34 dB, respectively. *Clearly, such a high prediction error would make the base model unusable in practice unless it is tediously re-trained per individual environments.*

Semantic-aware DCNN Model: Semantic labels as global features are crucial to recognize the similar-looking objects and build models that are transferable across environments. Recall that *Argus* augments the base model by first segmenting the PCD data, and then processing them to generate <*Global Features, MIDI, Pose, SRP*> pairs. Furthermore, *Argus* applies the masking to consider only those objects that are contributing strongly to the SRP (Section 3.3.2). We intend to build a model for each building, so to train for building A, we use samples from its first floor (Env A.1 to A.4), and to test, we use samples from its second floor (Env A.5 to A.7). We

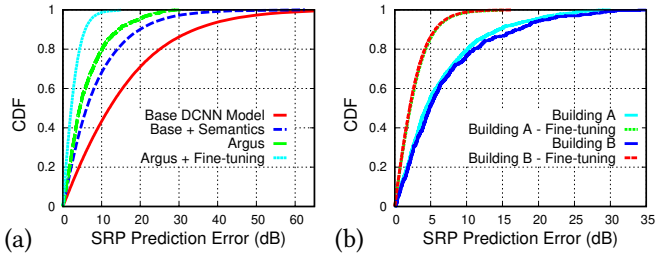

Figure 10. SRP prediction error comparison: (a) Base model, base model with semantic segmentation, *Argus*, and *Argus* with fine-tuning; training the models with 4 environments (Envs A.1 to A.4) and testing with 3 environments (Envs A.5 to A.7). (b) For buildings A and B, performance of *Argus* when trained and tested on their respective environments.

also compare the results with the base model and base model with only the semantic segmentation. All these networks are trained and tested with identical sets of samples for the same number of epochs used in *Argus* training. Finally, we apply fine-tuning to *Argus* by employing a few samples (collected within 4 minutes) from the second floor environments. [3]

Figure 10(a) shows the semantic-aware DCNN model results for building A. While the base model performs poorly with a median error of more than 12 dB, with the inclusion of global features and semantic labels, the median error is reduced by more than 2×. The prediction error is further reduced in *Argus*, which shows a median error of 4.35 dB only. Furthermore, Figure 10(b) shows that *Argus* (and fine-tuning results) translate across different buildings made of mostly different materials. Under buildings A and B with unseen test environments from different floors, the model shows the median SRP errors of 4.35 dB and 4.85 dB, respectively, which further reduces to 2.19 dB and 2.07 dB, respectively, after fine-tuning. *These results show that Argus is well generalizable across multiple environments, and its core semantic segmentation can easily transfer models from one environment to another with little to no fine-tuning.*

Scanning Time Requirement for Fine-Tuning: We now evaluate *Argus*'s performance with fine-tuning by scanning an environment for a different amount of time. Intuitively, fine-tuning with more scanned samples should reduce the prediction error further. To this end, *Argus* is trained with datasets from Env A.1 to A.4, and then tested with datasets from Env A.5 to A.7. Figure 11(a) shows the prediction results as a function of scanning time. We first test *Argus* with no fine-tuning data samples from Env A.5 to A.7, which corresponds to 0 min. of scan time. The median error is 4.35 dB and 1 standard deviation could be 10.03 dB. But these errors drop quickly as we add data samples to fine-tune *Argus*. By fine-tuning *Argus* with only 5 mins. of scanned samples, the error is reduced to 1.36 dB on median. But longer scanning times do not always improve the SRP prediction accuracy significantly: There is hardly 0.36 dB prediction accuracy improvement for an additional 25 mins. of scanning. *These results show that Argus adapts very well in unknown environments without requiring the deployer to spend a lot of time scanning and collecting new data samples.*

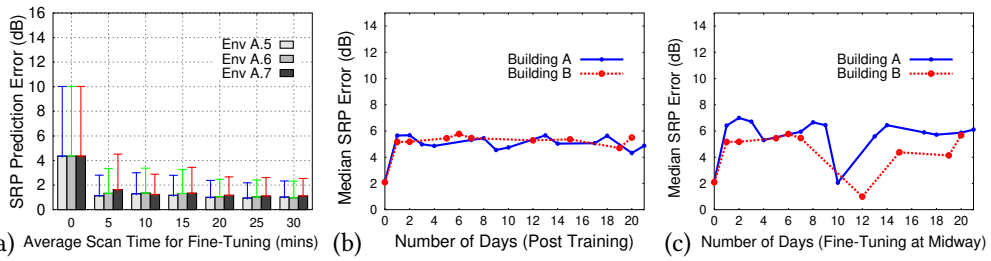

Figure 11. (a) Effect of scan time on model fine-tuning for building A. (b–c) Temporal SRP prediction error of Building A and B: (b) Without fine-tuning; (c) With fine-tuning Building A at 10^{th} day and B at 12^{th} day.

Temporal Performance of SRP Prediction: So far, we have analyzed the spatial performance of *Argus* across multiple environments. We would like to now understand how long the models remain stable after they are trained. To this end, we use the trained models from Building A and Building B and test with samples that are collected several days after training. For this analysis, we include data samples that are collected up to 21 days after model training, and Figure 11(b) shows that models' performances are stable over time across both the buildings. Furthermore, Figure 11(c) shows that by fine-tuning the models with only ∼ 1 min. of scanned data samples around 10^{th} day

[3]It does not require rebuilding a separate network for separate environments; only fine-tuning a single network.

in building A and 12^{th} day in building B, the average prediction error reduces below 2 dB. Note that such fine-tuning is only required when deployers plan to add new picocells in the network or re-deploy existing picocells due to large structural changes in the environment.

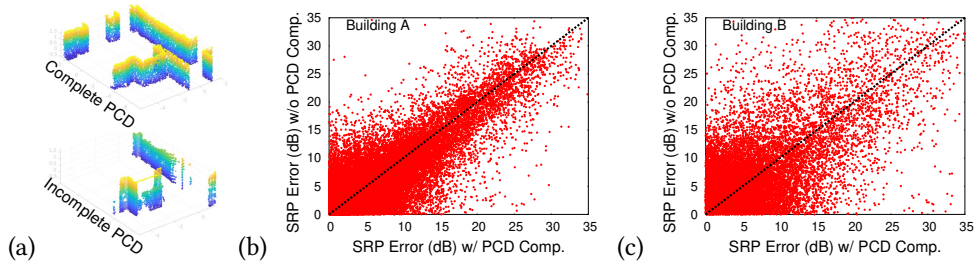

Figure 12. (a) PCD obtained from 1 min. and 10 min. of scans. (b–c) Analysis of SRP prediction errors with and without PCD completion: (b) for building A and (c) for building B.

Effect of PCD Completion in SRP Prediction: Due to the limited FoV of the AR camera, arbitrary scan trajectory, and random noise from deployer's quick scans, the measured PCDs could be incomplete and sparse (see Figure 12[a]). Since the DCNN model relies on the depth images, we would like to evaluate the effect of such PCD artifacts on *Argus*. To this end, we collect PCDs from multiple environments with a quick 1 min. scan and build *Argus*'s model. Then, we test SPR prediction in different environments without any fine-tuning. We repeat this process by re-scanning the environment for a longer duration to obtain a higher quality PCD. Figures 12(b–c) show that the higher scanning time (better quality PCD) has a negligible effect on SPR prediction. Each dot in the figures represents the SRP prediction error from an environment with low-quality (Y-axis) and high-quality (X-axis) PCDs, and the high-quality PCDs do not perform significantly better than the low-quality ones. The result supports the hypothesis that there are only a couple of strong reflectors in the environment which contributes to the SRP, and a low-quality PCD, obtained from a quick scan, captures the majority of the unique environmental information, and by using convolution layers, *Argus* is able to extract necessary features. *So, deployers do not have to spend a significant time scanning the environment to ensure the visual data has very high quality and is noise-free.*

5.2 Application Results with Predicted SRP

Picocell Deployment: Accurate SRP prediction allows us to build a practical Ray-tracing method that not only simulates separately located transmitter and receiver but also considers realistic mmWave reflections. Due to the lack of SRP access in practical 5G mmWave picocells, we emulate the picocell deployments in various indoor environments. We first simulate the full SRPs of an environment by passing its PCD to the Ray-tracing method, and then, apply the mean and standard deviation from *Argus*'s prediction error on the simulated SRPs. This will inject realist errors in prediction, as well as enable us to compare *Argus* with an *Optimal* method where the knowledge of accurate SRPs is available. Furthermore, we follow the 3 deployment strategies in

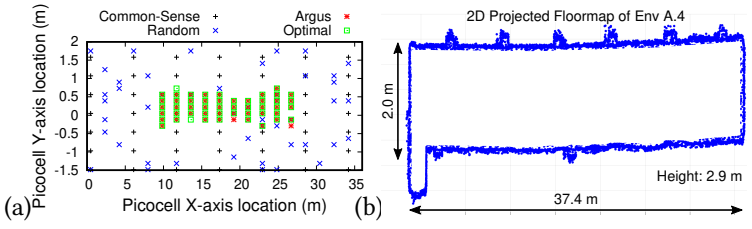

Figure 13. (a) Picocell locations estimated by Optimal, *Argus* (with "link-outage" strategy), Random, and Common-Sense. (b) Top-view of Env A.4.

Argus, "Average," "Variation," and "Link-outage", (Algorithm 1) to understand their performance variations. We also simulate two other deployment methods: *Random*, which places the picocells randomly within the environment; and *Common-Sense*, *a.k.a.*, corner deployment, which places the picocells at the corner locations. Figure 13(a) shows the picocells' locations predicted via the "Link-outage" strategy in *Argus* for Env A.4 (Figure 13[b]), and compare the result with the *Optimal*, *Random*, and *Common-sense* deployments. Results indicate that *Optimal* and *Argus*'s locations are mostly overlapping, *i.e.*, *Argus*'s network will likely perform similar to an optimal deployment.

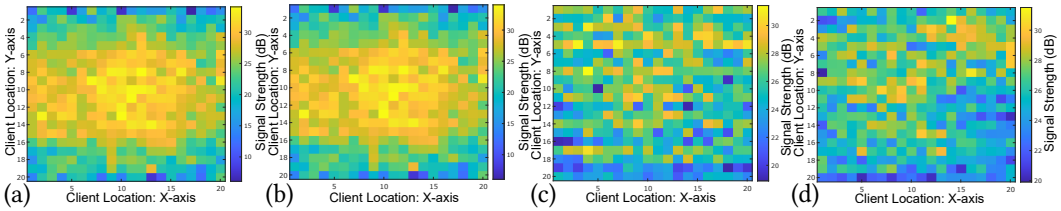

Figure 14. Example Coverage Profiles (CP) of the environment A.4 for four deployment methods: (a) Optimal; (b) *Argus* (with "link-outage" strategy); (c) Common-Sense (*a.k.a.* corner deployment); (d) Random.

Figures 14(a–d) further plots the *Coverage Profiles* under different methods. The signal strength distribution for Random and Common-Sense deployment show high variations and sporadic high performance regions; but *Argus* closely matches with the distribution under the optimal method. Figures 15(a–c) shows the results with these strategies; Figure 15(a) shows that we can achieve the median signal strength of -55 dB with *Argus*, which is close to the Optimal method (-52 dB). The average signal strength could be as low as -75 dB with *Random* and *Common-Sense* deployment methods. Similarly, Figure 15(b) shows that *Argus* limits the SRP variation to 1 dB for all receiver's locations while it grows up to 3 dB with *Random* and *Common-Sense* deployment methods. Finally, Figure 15(c) reveals that area without active link is reduced by ~1.55× with *Argus* as compared to *Random* and *Common-Sense* deployment methods. *These results emphasize the benefit of accurate SRP prediction in planning and deploying dense mmWave picocells in an environment.*

Location and Orientation Estimation: We now evaluate *Argus*'s ability to predict a device's location and orientation. To this end, we leverage the datasets from for multiple environments of building A, and train the model with thousands of samples randomly selected from the 7 different environments. For testing, we used the previously available original PCD and SRPs to predict the location and orientation in unknown environments to understand *Argus*'s performance. Figure 16(a) shows the distribution of location prediction error across different axes.

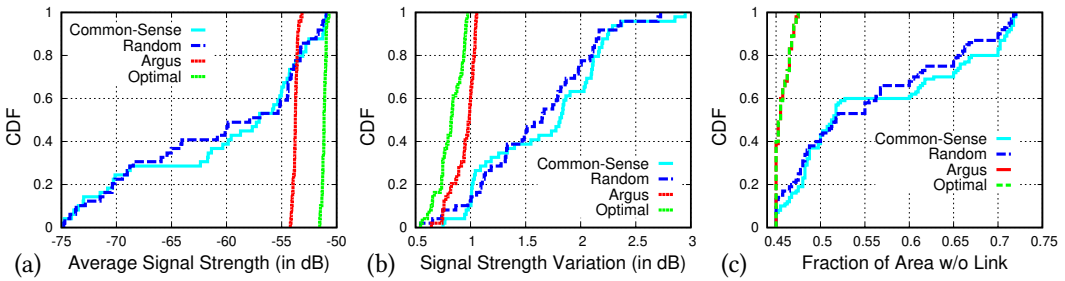

Figure 15. Deployment metrics with; (a) Average signal strength in users' locations. (b) Variation of signal strength across the environment. (c) Distribution of fraction of area w/o any link.

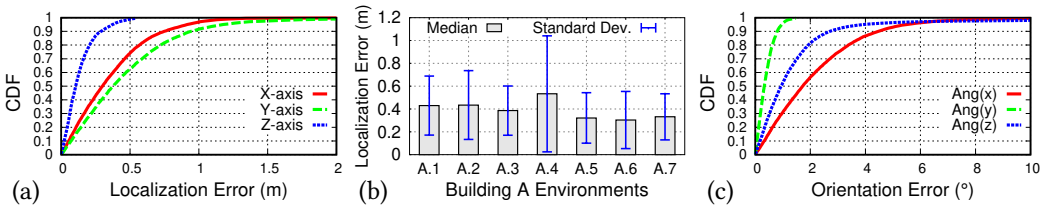

Figure 16. Evaluation of location and orientation estimation model; (a) Localization error in different axes. (b) Absolute localization error in different environments. (c) Orientation error in different axes.

The median prediction errors are less than 30 cm and 35 cm across X and Y axes, respectively. The prediction error across Z axis is the smallest, only 10.5 cm in median; this is because significant device movements are mainly limited to X-Y plane. Figure 16(b) further zooms in on the localization error for individual environments and plots the Euclidean distance between the predicted and ground truth locations. The results show that the prediction works consistently across multiple environments, and the median error is always lower than 45 cm, except for A.4. It turns out A.4 is a long hallway, which can act as a physical waveguide for mmWave signals [5, 6] and distort their features from the trained model. Still, the median prediction error is less than 50 cm, which is tolerable for many ubiquitous applications. *Argus* can predict device's orientation too, and Figure 16(c) shows that the orientation prediction has less than 1.7°, 0.3°, and 0.9° median errors for X, Y, and Z orientation estimation, which is highly accurate. In sum, mmWave devices could leverage *Argus*'s accurate location and orientation prediction for many indoor navigation applications.

Object Tagging and Retrieval: Finally, we evaluate *Argus*'s ability to tag objects during training and retrieve it during testing without relying on visual cues. This could be useful for VR/AR applications or object access via a Robot under poor lighting conditions. We use *Argus*'s DCN (Section 3.4.3) as the categorical classifier for the datasets of Env A.1, where we supply MIDI and SRP as inputs and get object category as output. We select 6 major objects for tagging, and during data preprocessing, we manually assign their object class labels. After training, we use the predicted SRP to classify multiple objects. We have used total of 18,000 data samples to train and 2,000 data samples to test. Table 2 shows the confusion matrix of categorical classification for the 6 objects in A.1, which shows more than ~ 99% accuracy in class label prediction for all objects, except *Object 6*. This is due to the training data imbalance for *Object 6*, which could be easily resolved by introducing more samples with *Object 6*.

Table 2. Confusion matrix of object classification of *Argus* trained and tested on Env A.1 datasets.

Actual/Predicted	Obj1	Obj2	Obj3	Obj4	Obj5	Obj6
Obj1	**99.48**	0	0.26	0	0.26	0
Obj2	0	**99.74**	0	0	0	0.26
Obj3	0	0	**99.42**	0.58	0	0
Obj4	0	0	0	**100**	0	0
Obj5	0	0	0.69	0	**99.31**	0
Obj6	2.7	2.3	0	0	0.86	**94.14**

In summary, these results demonstrate that Argus is well generalizable across diverse environments for SPR prediction, and it enables reliable and versatile mmWave networks and applications.

6 DISCUSSION AND FUTURE WORKS

Extending *Argus* to Outdoor Environments and Picocells Redeployment: Currently, *Argus* has been designed and validated in indoor environments, but we believe the deep learning approach will extend to outdoor environments too. For example, a network deployer could sparsely measure the SRPs from a few vantage points of an outdoor street and use the outdoor 3D visual data, *e.g.*, a Google street view [93], to predict the full SRPs in the environment. We leave this extension as future work. In the outdoor environment, there might be no reflection due to the unavailability of

reflectors at near distance and *Argus* can warn deployer to take extra measures such as increasing transmission power and picocell density, install intelligent surfaces, *etc.* Besides, *Argus* can also provide hints for future redeployment or picocell provisioning by continuously monitoring for a change in the SRPs distribution. A key challenge is to find out the right time for redeployment and provisioning. Intuitively, whenever there is a large structural change, some picocells could lose their reflection paths. But we need to distinguish between transient and permanent changes. We can leverage an outage risk assessment based on temporal learning, similar to [7], by correlating the changes in the SRPs with the link outages and long-term throughput to reassess for provisioning or redeployment periodically. We leave it for future work.

Picocell Deployment Density: When fully operational, 5G mmWave enterprise and cellular deployments are expected to be much denser than traditional Wi-Fi or LTE [94–96]; but the exact density requirement is still an open question. Higher density brings challenges with higher cost and power, complex cabling, and scalable backhauling. In the future, fixed wireless between picocells could potentially eliminate wired backhauling [92, 97, 98], and mmWave hardware is expected to be more inexpensive and power-efficient [99, 100]. Still, we believe the densifications would be limited by factors beyond technical issues: Deployers' incentives and management cost.

To enable a cost-benefit analysis of deployment, *Argus* can limit the number of picocells and compare its effectiveness *w.r.t.* the optimal deployment. Specifically, after the visual data to SRP prediction, we can virtually place the picocells at different locations in the area. Then, for each case, we can simulate the network with users' average mobility, obstruction, and traffic patterns to predict the long-term network behavior and power consumptions and identify the set of locations that optimizes the performance. We can then compare the effectiveness of both the optimal and limited-picocell deployments by estimating their *probability of link outages, long-term throughput, and area spectral efficiency* [7, 101, 102]. This allows greater flexibility for the deployers to select the number of picocells beforehand, based on their power budget and management cost-benefit analysis. Besides, *Argus* can also showcase the benefits of increasing the number of picocells to inform about better effectiveness with higher cost. Currently, we train and test *Argus* offline. However, in the future, we plan to upload the trained model in the cloud and run tests on mobile platforms in real-time. Overall, *Argus* can become be a versatile "*what-if*" deployment analysis tool to help the deployers make better decisions before upgrading an area with mmWave technology. We leave this "*what-if*" analysis with *Argus* as future work.

7 RELATED WORK

Millimeter-Wave Signal Reflection Prediction: Even though knowledge of signal strength map is critical for network planning and operation, accurate retrieval of this information is expensive and sometimes error-prone [103]. Signal strength prediction is closely related to path loss prediction in the urban and rural areas, and a simple model, like Friis path loss [50], has been shown to be effective for lower frequencies, but fails for higher frequencies, like mmWave [104]. One of the major reasons is the diffraction [105] which needs to be accounted for frequencies around 10–26 GHz. Existing works have used simulation models due to the lack of extensive datasets to predict signal strength for 5G networks [106]. However, such models are error-prone, and simulation might not capture accurate reflectivity properties of objects. [15] first reconstructs the environment using measurements and mmWave parameters and then uses the Ray-tracing simulation to predict the signal strength. Even though [15] could achieve a prediction accuracy with a median error of 2.8 dB, it requires tedious measurement and modeling across individual environments. Recently, Lumos5G implemented the RNN based deep learning framework to predict the 5G outdoor signal

strength and developed a Google Map like 5G coverage map, but they only use client features, such as moving speed, geolocation, compass direction, *etc.*, and position of existing 5G picocells to give a rough performance prediction [107]. Different from all existing works, *Argus* proposes a first-of-a-kind vision and learning augmented system to predict mmWave signal reflections, which is not only accurate but also generalizes well across multiple environments.

Applications of Millimeter-Wave Signal: Accurate prediction of signal reflection profile opens various opportunities for 5G network assisted technologies. For example, [108, 109] use pose information of mobile clients to manage multi-link and spatial reuse based on traffic density. [110, 111] builds the 5G throughput coverage map during mobility with the help of unmanned aerial vehicles. Localization services in 5G are extensively used in various scenarios, such as bike sharing, targeted advertisement, traffic flow management, public safety, *etc.* [112, 113], and SRP is used as the key component to help in localization. [114] uses SRP to localize Wi-Fi users within the network using machine learning approaches. On other hand, [115] uses the signal arrival angle to fingerprint and localize devices within the network. Besides, there are numerous computational approaches, such as iterative stochastic gradient descent [116] or deep learning [117–119] that exhibit better results. Furthermore, wireless network planning has relied on empirical solutions for decades, *e.g.*, using thorough site surveys or common-sense deployments. Site survey, despite being tedious, is still widely used by enterprise and cellular network deployers. AR applications based on the 5G networks have been widely used in Mobile Augmented Reality (MAR) [75, 120], and 5G based AR has been used in the classroom for education [121] and to improve operator's safety in complex and dangerous industrial sites [122]. In *Argus*, we show that the predicted and measured SRPs could be leveraged to design three applications, optimal picocell deployment, accurate location and orientation prediction, and correct object and tagging and retrieval, and we believe our approach opens up the opportunity for many mmWave SRP based applications in the future.

8 CONCLUSION

We present *Argus*, a first-of-a-kind system that accurately predicts the mmWave signal reflection profiles at unobserved locations by building a deep learning model from sparse and randomly collected measurements. The model exploits semantic information of an environment to make it more generalizable. *Argus* uses the information gathered from SRP prediction to precisely estimate picocell locations and provide optimal network coverage. *Argus*'s SRP prediction and picocell deployment further help users to localize themselves accurately, and it also enables tagging and retrieving surrounding objects. Experimental results demonstrate that *Argus* is well generalizable across diverse environments for SPR prediction, and it enables reliable and versatile mmWave networks and applications.

ACKNOWLEDGMENTS

We sincerely thank the reviewers and our Shepherd, Prof. Shaileshh Bojja Venkatakrishnan, for their comments and feedback. We also thank Timothy Dayne Hooks, Stephen Baione, Lance Kevin, and Edward Sitar for their help with the data collection process. This work is partially supported by the NSF under grants CNS-1910853 and MRI-2018966.

REFERENCES

[1] IEEE P802.11 - Task Group ay, "Status of Project IEEE 802.11ay," http://www.ieee802.org/11/Reports/tgay_update.htm, 2020.

[2] 3GPP: A Global Initiative, "The Mobile Broadband Standard: Release 17," 2021. [Online]. Available: http://www.3gpp.org/release-17

[3] Mathew K. Samimi and Theodore S. Rappaport, "3-D Statistical Channel Model for Millimeter-Wave Outdoor Mobile Broadband Communications," in *IEEE International Conference on Communications (ICC)*, 2015.

[4] Theodore S. Rappaport and Eshar Ben-Dor and James N. Murdock and Yijun Qiao, "38 GHz and 60 GHz Angle-Dependent Propagation for Cellular and Peer-to-Peer Wireless Communications," in *IEEE International Conference on Communications (ICC)*, 2012.

[5] Hao Xu and Vikas Kukshya and Theodore S. Rappaport, "Spatial and Temporal Characteristics of 60-GHz Indoor Channels," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 3, 2002.

[6] Sanjib Sur and Vignesh Venkateswaran and Xinyu Zhang and Parmesh Ramanathan, "60 GHz Indoor Networking through Flexible Beams: A Link-Level Profiling," in *Proc. of ACM SIGMETRICS*, 2015.

[7] Sanjib Sur and Xinyu Zhang and Parmesh Ramanathan and Ranveer Chandra, "BeamSpy: Enabling Robust 60 GHz Links Under Blockage," in *Proceedings of USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2016.

[8] Christopher R. Anderson and Theodore S. Rappaport, "In-Building Wideband Partition Loss Measurements at 2.5 and 60 GHz," *IEEE Transactions on Wireless Communications*, vol. 3, no. 3, 2004.

[9] Theodore S. Rappaport and Felix Gutierrez and Eshar Ben-Dor and James N. Murdock and Yijun Qiao and Jonathan I. Tamir, "Broadband Millimeter-Wave Propagation Measurements and Models Using Adaptive-Beam Antennas for Outdoor Urban Cellular Communications," *IEEE Transactions on Antennas and Propagation*, vol. 61, no. 4, 2013.

[10] Peter F. M. Smulders, "Statistical Characterization of 60-GHz Indoor Radio Channels," *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 10, 2009.

[11] Ahmed M. Al-Samman and Tharek A. Rahman and Marwan H. Azmi and M.N. Hindia, "Large-scale path loss models and time dispersion in an outdoor line-of-sight environment for 5G wireless communications," *AEU - International Journal of Electronics and Communications*, vol. 70, no. 11, 2016.

[12] Virk, Usman Tahir and Haneda, Katsuyuki, "Modeling Human Blockage at 5G Millimeter-Wave Frequencies," *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 3, 2020.

[13] Moltchanov, Dmitri and Ometov, Aleksandr and Kustarev, Pavel and Evsutin, Oleg and Hosek, Jiri and Koucheryavy, Yevgeni, "Analytical TCP Model for Millimeter-Wave 5G NR Systems in Dynamic Human Body Blockage Environment," *Sensors*, vol. 20, no. 14, 2020.

[14] Sanjib Sur and Xinyu Zhang, "Scoping Environment for Robust 60 GHz Link Deployment," in *Invited paper at Information Theory and Applications*, 2016.

[15] Wei, Teng and Zhou, Anfu and Zhang, Xinyu, "Facilitating Robust 60 GHz Network Deployment by Sensing Ambient Reflectors," in *Proceedings of the 14th USENIX Conference on Networked Systems Design and Implementation*, 2017.

[16] Timothy Dayne Hooks and Hem Regmi and Sanjib Sur, "Poster: VisualMM: Visual Data and Learning Aided 5G Picocell Placement," in *Proceedings of ACM International Workshop on Mobile Computing Systems and Applications (HotMobile)*, 2021.

[17] Cisco, "Site Survey Guidelines for WLAN Deployment," 2021. [Online]. Available: https://www.cisco.com/c/en/us/support/docs/wireless/5500-series-wireless-controllers/116057-site-survey-guidelines-wlan-00.html

[18] iBwave Solutions, "iBwave Design: A Single Solution to Streamline the Design of All Your Indoor Wireless Network Projects," 2021. [Online]. Available: https://www.ibwave.com/ibwave-design

[19] SecureEdge Networks, "Software that Makes WiFi More Intelligent," 2021. [Online]. Available: https://www.securedgenetworks.com/software

[20] Transition Products Inc., "Wireless Site Surveys," 2021. [Online]. Available: https://www.tpi1.com/services/wireless-site-surveys/

[21] ExterNetworks Inc., "Wireless Site Survey," 2021. [Online]. Available: https://www.extnoc.com/wireless-site-survey

[22] VisiWave, "Visualize Your Wireless Network," 2021. [Online]. Available: https://www.visiwave.com/

[23] Made by WiFi Inc., "How Much Does A Wireless Site Survey Cost And Is It Worth It For My Business?" 2021. [Online]. Available: https://www.madebywifi.com/blog/how-much-does-a-wireless-site-survey-cost-and-is-it-worth-it-for-my-business/

[24] ExterNetworks Inc., "Wireless Site Survey — Outdoor Survey for upto 100,000 Sq. Ft," 2021. [Online]. Available: https://www.extnoc.com/store/wireless-outdoor-survey-100000sqft/

[25] Remcom, "Wireless InSite: 3D Wireless Prediction Software," 2021. [Online]. Available: https://www.remcom.com/wireless-insite-em-propagation-software

[26] Altair Engineering, Inc., "Altair Feko Applications," 2021. [Online]. Available: https://www.altair.com/feko-applications/

[27] Siradel, "Software for Wireless Network and Smart City Planning," 2021. [Online]. Available: https://www.siradel.com/solutions/software/

[28] SonicWall, "WiFi Planner: Elevate Your WiFi User Experience with the Right Design," 2021. [Online]. Available: https://www.sonicwall.com/products/secure-wireless/wifi-planner/

[29] Sanjib Sur and Xinyu Zhang, "Scoping Environment to Assist 60 GHz Link Deployment," in *Proceedings of ACM International Conference on Mobile Computing and Networking (MobiCom) Poster*, 2015.

[30] Basar, Ertugrul and Di Renzo, Marco and De Rosny, Julien and Debbah, Merouane and Alouini, Mohamed-Slim and Zhang, Rui, "Wireless Communications Through Reconfigurable Intelligent Surfaces," *IEEE Access*, vol. 7, 2019.
[31] Gsmarena, "Asus Zenfone AR ZS571KL," 2021. [Online]. Available: https://www.gsmarena.com/asus_zenfone_ar_zs571kl-8502.php
[32] Silicon Radar GmbH, "24 GHz Products," 2020. [Online]. Available: https://siliconradar.com/products/#24ghz-radar-chips
[33] Sanjib Sur, "Software-Hardware Reconfigurable Systems for Mobile Millimeter-Wave Networks," 2022. [Online]. Available: https://cse.sc.edu/~sur/projects/nsf1910853
[34] M. Soumekh, *Synthetic Aperture Radar Signal Processing*. John Wiley & Sons, Inc., 1999.
[35] Nozhan Hosseini and Mahfuza Khatun and Changyu Guo and Kairui Du and Ozgur Ozdemir and David Matolak and Ismail Guvenc and Hani Mehrpouyan, "Attenuation of Several Common Building Materials: Millimeter-Wave Frequency Bands 28, 73, and 91 GHz," in *IEEE Antennas and Propagation Magazine*, 2021.
[36] Timothy A. Thomas and Huan Cong Nguyen and George R. MacCartney Jr. and Theodore S. Rappaport, "3D mmWave Channel Model Proposal," in *IEEE Vehicular Technology Conference*, 2014.
[37] Yaniv Azar and George N. Wong and Kevin Wang and Rimma Mayzus and Jocelyn K. Schulz and Hang Zhao and Felix Gutierrez and DuckDong Hwang and Theodore S. Rappaport, "28 GHz Propagation Measurements for Outdoor Cellular Communications using Steerable Beam Antennas in New York City," in *IEEE International Conference on Communications (ICC)*, 2013.
[38] Google, "Tango," 2014. [Online]. Available: https://www.youtube.com/watch?v=Qe10ExwzCqk
[39] ——, "ARCore," 2021. [Online]. Available: https://developers.google.com/ar
[40] Iro Armeni and Ozan Sener and Amir R. Zamir and Helen Jiang and Ioannis Brilakis and Martin Fischer and Silvio Savarese, "3D Semantic Parsing of Large-Scale Indoor Spaces," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2016.
[41] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, 2004.
[42] Alexander Maltsev and Roman Maslennikov and Artyom Sevastyanov and Alexey Lomayev and Alexey Khoryaev, "Statistical Channel Model for 60 GHz WLAN Systems in Conference Room Environment," in *In Proceedings of the Fourth European Conference on Antennas and Propagation (EuCAP)*, 2010.
[43] Chao Dong and Chen Change Loy and Kaiming He and Xiaoou Tang, "Image Super-Resolution Using Deep Convolutional Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, 2016.
[44] Wenming Yang and Xuechen Zhang and Yapeng Tian and Wei Wang and Jing-Hao Xue and Qingmin Liao, "Deep Learning for Single Image Super-Resolution: A Brief Review," *IEEE Transactions on Multimedia*, vol. 21, no. 12, 2019.
[45] Kevin de Haan and Yair Rivenson and Yichen Wu and Aydogan Ozcan, "Deep-Learning-Based Image Reconstruction and Enhancement in Optical Microscopy," *Proceedings of the IEEE*, vol. 108, no. 1, 2020.
[46] Saiprasad Ravishankar and Jong Chul Ye and Jeffrey A. Fessler, "Image Reconstruction: From Sparsity to Data-Adaptive Methods and Machine Learning," *Proceedings of the IEEE*, vol. 108, no. 1, 2020.
[47] Florian Knoll and Kerstin Hammernik and Chi Zhang and Steen Moeller and Thomas Pock and Daniel K. Sodickson, "Deep-Learning Methods for Parallel Magnetic Resonance Imaging Reconstruction: A Survey of the Current Approaches, Trends, and Issues," *IEEE Signal Processing Magazine*, vol. 37, no. 1, 2020.
[48] Sarabandi, K and Vahidpour, M and Moallem, M and East, J, "Compact Beam Scanning 240 GHz Radar for Navigation and Collision Avoidance," in *Micro-and Nanotechnology Sensors, Systems, and Applications III*, vol. 8031, 2011.
[49] Mosalanejad, Mohammad and Ocket, Ilja and Soens, Charlotte and Vandenbosch, Guy AE, "Multilayer Compact Grid Antenna Array for 79 GHz Automotive Radar Applications," *IEEE Antennas and Wireless Propagation Letters*, vol. 17, no. 9, 2018.
[50] Rappaport, Theodore S, *Wireless Communications: Principles and Practice*. Prentice Hall, 2002.
[51] Sanjib Sur and Ioannis Pefkianakis and Xinyu Zhang and Kyu-Han Kim, "Towards Scalable and Ubiquitous Millimeter-Wave Wireless Networks," in *Proc. of ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2018.
[52] Song, Rongguo and Wang, Zhe and Zu, Haoran and Chen, Qiang and Mao, Boyang and Wu, Zhi Peng and He, Daping, "Wideband and Low Sidelobe Graphene Antenna Array for 5G Applications," *Sci. Bull*, 2020.
[53] Huang, He and Li, Xiaoping and Liu, Yanming, "5G MIMO Antenna Based on Vector Synthetic Mechanism," *IEEE Antennas and Wireless Propagation Letters*, vol. 17, no. 6, 2018.
[54] Pandya, Sharnil and Wakchaure, Manoj Ashok and Shankar, Ravi and Annam, Jagadeeswara Rao, "Analysis of NOMA-OFDM 5G Wireless System Using Deep Neural Network," *The Journal of Defense Modeling and Simulation*, 2021.
[55] Li, Tian-Hao and Khandaker, Muhammad RA and Tariq, Faisal and Wong, Kai-Kit and Khan, Risala T, "Learning the wireless V2I channels using deep neural networks," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*,

2019.
[56] Ravi, Nagarathna and Rani, P Vimala and Shalinie, S Mercy, "Secure Deep Neural (SeDeN) Framework for 5G Wireless Networks," in *2019 10th International Conference on computing, communication and networking technologies (ICCCNT)*, 2019.
[57] Gu, Jiuxiang and Wang, Zhenhua and Kuen, Jason and Ma, Lianyang and Shahroudy, Amir and Shuai, Bing and Liu, Ting and Wang, Xingxing and Wang, Gang and Cai, Jianfei and others, "Recent Advances in Convolutional Neural Networks," *Pattern Recognition*, vol. 77, 2018.
[58] Zhang, Chen-Lin and Luo, Jian-Hao and Wei, Xiu-Shen and Wu, Jianxin, "In Defense of Fully Connected Layers in Visual Representation Transfer," in *Pacific Rim Conference on Multimedia*, 2017.
[59] Keras, "Keras Applications," 2021. [Online]. Available: https://keras.io/api/applications/
[60] Karen Simonyan and Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *International Conference on Learning Representations*, 2015.
[61] He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
[62] Szegedy, Christian and Vanhoucke, Vincent and Ioffe, Sergey and Shlens, Jon and Wojna, Zbigniew, "Rethinking the Inception Architecture for Computer Vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
[63] G. Huang and Z. Liu and L. Van Der Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
[64] Andrew G. Howard and Menglong Zhu and Bo Chen and Dmitry Kalenichenko and Weijun Wang and Tobias Weyand and Marco Andreetto and Hartwig Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017. [Online]. Available: https://arxiv.org/abs/1704.04861
[65] Sandler, Mark and Howard, Andrew and Zhu, Menglong and Zhmoginov, Andrey and Chen, Liang-Chieh, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.
[66] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
[67] Qi, Charles R. and Su, Hao and Mo, Kaichun and Guibas, Leonidas J., "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
[68] Scikit-Learn, "OneHotEncoder," 2021. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html
[69] Tsai, Chun-Wei and Cho, Hsin-Hung and Shih, Timothy K and Pan, Jeng-Shyang and Rodrigues, Joel JPC, "Metaheuristics for the deployment of 5G," *IEEE Wireless Communications*, vol. 22, no. 6, 2015.
[70] Xiao, Zhu and Liu, Hongjing and Havyarimana, Vincent and Li, Tong and Wang, Dong, "Analytical study on multi-tier 5G heterogeneous small cell networks: Coverage performance and energy efficiency," *Sensors*, vol. 16, no. 11, 2016.
[71] Im, Chaehun and Jung, Sunghoon and Lee, Chungyoung, "A deep autoencoder approach to received signal strength-based localization with unknown channel parameters," in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2020.
[72] Sathish, L and Bhuvaneswari, Y Satya and Devi, B Satya Sri and Nandan, Durgesh, "Analysis of Received Signal Strength Based on User Position Locating by Using ML Methods," in *International Conference on Emerging Trends and Advances in Electrical Engineering and Renewable Energy*, 2020.
[73] e Silva, Pedro Figueiredo and Richter, Philipp and Talvitie, Jukka and Laitinen, Elina and Lohan, Elena Simona, "Challenges and solutions in Received Signal Strength-based seamless positioning," in *Geographical and Fingerprinting Data to Create Systems for Indoor Positioning and Indoor/Outdoor Navigation*, 2019.
[74] Menta, Estifanos Yohannes and Malm, Nicolas and Jäntti, Riku and Ruttik, Kalle and Costa, Mário and Leppänen, Kari, "On the performance of AoA–based localization in 5G ultra–dense networks," *IEEE Access*, vol. 7, 2019.
[75] Siriwardhana, Yushan and Porambage, Pawani and Liyanage, Madhusanka and Ylianttila, Mika, "A Survey on Mobile Augmented Reality With 5G Mobile Edge Computing: Architectures, Applications, and Technical Aspects," *IEEE Communications Surveys Tutorials*, vol. 23, no. 2, 2021.
[76] Yang, Chang-Fa and Wu, Boau-Cheng and Ko, Chuen-Jyi, "A Ray-Tracing Method for Modeling Indoor Wave Propagation and Penetration," *IEEE transactions on Antennas and Propagation*, vol. 46, no. 6, 1998.
[77] Shihao Ju, Syed Hashim Ali Shah, Muhammad Affan Javed, Jun Li, Girish Palteru, Jyotish Robin, Yunchou Xing, Ojas Kanhere, Theodore S. Rappaport, "Scattering Mechanisms and Modeling for Terahertz Wireless Communications," in *IEEE International Conference on Communications (ICC)*, 2019.
[78] Amini, Navid and Sarrafzadeh, Majid and Vahdatpour, Alireza and Xu, Wenyao, "Accelerometer-based on-body sensor localization for health and medical monitoring applications," *Pervasive and mobile computing*, vol. 7, no. 6, 2011.

[79] Hsu, Ching-Hsien and Yu, Chia-Hao, "An accelerometer based approach for indoor localization," in *2009 Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing*. IEEE, 2009, pp. 223–227.
[80] Alam, Fakhrul and Faulkner, Nathaniel and Legg, Mathew and Demidenko, Serge, "Indoor visible light positioning using spring-relaxation technique in real-world setting," *Ieee Access*, vol. 7, 2019.
[81] Sarfraz, M and Rizvi, SM Ali J, "Indoor navigational aid system for the visually impaired," in *Geometric Modeling and Imaging (GMAI'07)*. IEEE, 2007, pp. 127–132.
[82] Ayyalasomayajula, Roshan and Arun, Aditya and Wu, Chenfeng and Sharma, Sanatan and Sethi, Abhishek Rajkumar and Vasisht, Deepak and Bharadia, Dinesh, "Deep Learning Based Wireless Localization for Indoor Navigation," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 2020.
[83] Jain, Puneet and Manweiler, Justin and Roy Choudhury, Romit, "OverLay: Practical Mobile Augmented Reality," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, 2015.
[84] "Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss, Logistic Loss, Focal Loss and all those confusing names," https://gombru.github.io/2018/05/23/, 2021.
[85] Sheen, David M and McMakin, Douglas L and Hall, Thomas E, "Three-dimensional millimeter-wave imaging for concealed weapon detection," *IEEE Transactions on microwave theory and techniques*, vol. 49, no. 9, 2001.
[86] "Real-Time Appearance-Based Mapping," http://introlab.github.io/rtabmap/, 2021.
[87] "Optimizers," https://keras.io/api/optimizers/, 2021.
[88] Open-Source, "TensorFlow," 2021. [Online]. Available: https://www.tensorflow.org/
[89] NVIDIA, "GEFORCE," 2021. [Online]. Available: https://www.nvidia.com/en-us/geforce/
[90] Google, "Cloud TPU," 2021. [Online]. Available: https://cloud.google.com/tpu
[91] NVIDIA, "RTX A6000," 2021. [Online]. Available: https://www.nvidia.com/en-us/design-visualization/rtx-a6000/
[92] Erik Dahlman and Stefan Parkvall and Johan Skold, *5G NR: The Next Generation Wireless Access Technology*. Elsevier, 2018.
[93] Google, "Google Street View," 2021. [Online]. Available: https://www.google.com/streetview/
[94] Verizon, "Explore 4G LTE and 5G Network Coverage in Your Area," 2021. [Online]. Available: https://www.verizon.com/coverage-map/
[95] T-Mobile USA, Inc., "5G & 4G LTE Coverage," 2021. [Online]. Available: https://www.t-mobile.com/coverage/coverage-map
[96] AT&T, "Nationwide 5G," 2021. [Online]. Available: https://www.att.com/5g/coverage-map/
[97] Stefan Parkvall and Erik Dahlman and Anders Furuskar and Mattias Frenne, "NR: The New 5G Radio Access Technology," *IEEE Communications Standards Magazine*, vol. 1, no. 4, 2017.
[98] Telefonaktiebolaget L. M. Ericsson, "5G Fixed Wireless Access," 2021. [Online]. Available: https://www.ericsson.com/en/fixed-wireless-access
[99] DARPA, "DARPA T-MUSIC," 2020. [Online]. Available: https://www.darpa.mil/news-events/2020-02-04
[100] VTT Technical Research Centre of Finland Ltd., "DREAM: D-Band Radio Solution Enabling Up To 100 Gbps Reconfigurable Approach for Meshed Beyond 5G networks," 2021. [Online]. Available: http://www.h2020-dream.eu/
[101] Fadhil Firyaguna and Jacek Kibilda and Carlo Galiotto and Nicola Marchetti, "Coverage and Spectral Efficiency of Indoor mmWave Networks with Ceiling-Mounted Access Points," in *IEEE Global Communications Conference*, 2017.
[102] Rony Kumer Saha, "On Maximizing Energy and Spectral Efficiencies Using Small Cells in 5G and Beyond Networks," *MDPI Sensors*, vol. 20, no. 6, 2020.
[103] Alimpertis, Emmanouil and Markopoulou, Athina and Butts, Carter and Psounis, Konstantinos, "City-Wide Signal Strength Maps: Prediction with Random Forests," in *The World Wide Web Conference*, 2019.
[104] Sulyman, Ahmed Iyanda and Nassar, Almuthanna T. and Samimi, Mathew K. and Maccartney, George R. and Rappaport, Theodore S. and Alsanie, Abdulhameed, "Radio propagation path loss models for 5G cellular networks in the 28 GHZ and 38 GHZ millimeter-wave bands," *IEEE Communications Magazine*, vol. 52, no. 9, 2014.
[105] Deng, Sijia and MacCartney, Geoge R. and Rappaport, Theodore S., "Indoor and Outdoor 5G Diffraction Measurements and Models at 10, 20, and 26 GHz," in *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016.
[106] Hossain, Ferdous and Geok, Tan Kim and Rahman, Tharek Abd and Hindia, Mohammad Nour and Dimyati, Kaharudin and Ahmed, Sharif and Tso, Chih P. and Abd Rahman, Noor Ziela, "An Efficient 3-D Ray Tracing Method: Prediction of Indoor Radio Propagation at 28 GHz in 5G Network," *Electronics*, vol. 8, no. 3, 2019.
[107] Narayanan, Arvind and Ramadan, Eman and Mehta, Rishabh and Hu, Xinyue and Liu, Qingxu and Fezeu, Rostand A. K. and Dayalan, Udhaya Kumar and Verma, Saurabh and Ji, Peiqi and Li, Tao and Qian, Feng and Zhang, Zhi-Li, "Lumos5G: Mapping and Predicting Commercial MmWave 5G Throughput," in *Proceedings of the ACM Internet Measurement Conference*, 2020.
[108] Wei, Teng and Zhang, Xinyu, "Pose Information Assisted 60 GHz Networks: Towards Seamless Coverage and Mobility Support," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. New York, NY, USA: Association for Computing Machinery, 2017.

[109] Tsukamoto, Yu and Hirayama, Haruhisa and Moon, Seung Il and Nanba, Shinobu and Shinbo, Hiroyuki, "Feedback Control for Adaptive Function Placement in Uncertain Traffic Changes on an Advanced 5G System," in *2021 IEEE 18th Annual Consumer Communications Networking Conference (CCNC)*, 2021.

[110] Mathew, Anitha P and Arthi, M and Babu, K Vinoth, "An uniform clustering based coverage and cost effective placement of serving nodes for 5G," in *2017 International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT)*, 2017.

[111] Khan, Shah Khalid and Naseem, Usman and Sattar, Abdul and Waheed, Nazar and Mir, Adnan and Qazi, Atika and Ismail, Muhammad, "UAV-aided 5G Network in Suburban, Urban, Dense Urban, and High-rise Urban Environments," in *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*, 2020.

[112] Bartoletti, Stefania and Conti, Andrea and Dardari, Davide and Giorgetti, Andrea, "5G localization and context-awareness," *University of Bologna, University of Ferrara*, 2018.

[113] Khan, Muhammad Alee and Saeed, Nasir and Ahmad, Arbab Waheed and Lee, Chankil, "Location Awareness in 5G Networks Using RSS Measurements for Public Safety Applications," *IEEE Access*, 2017.

[114] Belay, Abebe and Yen, Lei and Renu, Sakthidasan and Lin, Hsin-piao and Jeng, Shiann-Shiun, "Indoor Localization at 5 GHz Using Dynamic Machine Learning Approach (DMLA)," in *2017 International Conference on Applied System Innovation (ICASI)*, 2017.

[115] Meng, Jiayi and Sharma, Abhigyan and Tran, Tuyen X. and Balasubramanian, Bharath and Jung, Gueyoung and Hiltunen, Matti and Charlie Hu, Y., "A Study of Network-Side 5G User Localization Using Angle-Based Fingerprints," in *2020 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN*, 2020.

[116] Pandey, Ankur and Pinky, Pinky and Kumar, Sudhir, "Localization Using Stochastic Gradient Descent Method in a 5G Network," in *2018 15th IEEE India Council International Conference (INDICON)*, 2018.

[117] El Boudani, Brahim and Kanaris, Loizos and Kokkinis, Akis and Kyriacou, Michalis and Chrysoulas, Christos and Stavrou, Stavros and Dagiuklas, Tasos, "Implementing Deep Learning Techniques in 5G IoT Networks for 3D Indoor Positioning: DELTA (DeEp Learning-Based Co-operaTive Architecture)," *Sensors*, vol. 20, 2020.

[118] Klus, Roman and Klus, Lucie and Solomitckii, Dmitrii and Valkama, Mikko and Talvitie, Jukka, "Deep Learning Based Localization and HO Optimization in 5G NR Networks," in *2020 International Conference on Localization and GNSS (ICL-GNSS)*, 2020.

[119] Butt, M. Majid and Rao, Anil and Yoon, Daejung, "RF Fingerprinting and Deep Learning Assisted UE Positioning in 5G," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020.

[120] Jason Orlosky and Kiyoshi Kiyokawa and Haruo Takemura, "Virtual and Augmented Reality on the 5G Highway," *Journal of Information Processing*, vol. 25, 2017.

[121] Baratè, Adriano and Haus, Goffredo and Ludovico, Luca Andrea and Pagani, Elena and Scarabottolo, Nello, "5G technology for augmented and virtual reality in education," in *Proceedings of the International Conference on Education and New Developments*, 2019.

[122] Verde, Sebastiano and Marcon, Marco and Milani, Simone and Tubaro, Stefano, "Advanced Assistive Maintenance Based on Augmented Reality and 5G Networking," *Sensors*, vol. 20, no. 24, 2020.

Received October 2021; revised December 2021; accepted January 2022