RecuGAN: A Novel Generative AI Approach for Synthesizing RF Coverage Maps

Sopan Sarkar*, Mohammad Hossein Manshaei*,†, Marwan Krunz*, and Hamid Ravaee†
*Department of Electrical and Computer Engineering, University of Arizona, Tucson, USA
† Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran
{sopansarkar, manshaei, krunz}@arizona.edu

Abstract—**Radio-frequency coverage maps (RF maps) are essential in wireless communication, but obtaining them through site surveys can be labor-intensive and sometimes impractical. To address this challenge, we propose RecuGAN, a generative adversarial network (GAN)-based approach for generating RF maps. RecuGAN leverages the principles of information maximizing GAN (InfoGAN) to capture latent properties of RF maps, enabling unsupervised categorization and generation of new and diverse RF maps. Unlike traditional methods, RecuGAN does not require labeled data or conditional input, reducing complexity, time, and cost. We enhance the RecuGAN objective function with a customized gradient penalty-based Wasserstein GAN (WGAN) function and a gradient-based loss function for stable training and accurate map generation. We also provide the option to incorporate multiple generators in RecuGAN, enabling high-resolution RF map generation. As demonstrated through extensive training with both experimental and simulation data, RecuGAN can synthesize diverse high-quality RF maps and categorize them based on the RSS distribution. Compared to a UNet-based conditional GAN (cGAN), RecuGAN achieves a mean average percentage error (MAPE) of 1.18%, outperforming the cGAN model, which achieves a MAPE of 2.5%.**

Index Terms—**Generative Adversarial Networks, RF Mapping, Deep Neural Networks, Coverage, AI.**

I. Introduction

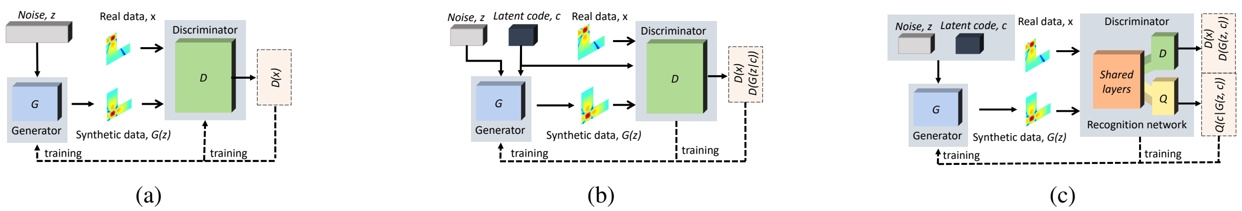
Radio-frequency coverage maps (RF maps) are of great importance in wireless communication applications like radio network planning, resource allocation, coverage estimation, localization, and handover management [1]. Collecting RF data requires site surveying, during which engineers consider different aspects of the coverage area, such as buildings, large objects, and walls, measure nearby cell interference, and generate RF maps. Site surveys are labor-intensive and complicated as they require carrying instruments around the measurement area and taking multiple measurements of the received signal strength (RSS) or signal-to-noise ratio (SNR) [2], [3]. For example, as stated in [4], it takes about 11.9 hours to survey a building of an area of 4500 m^2 using only 475 reference points (RP) for omnidirectional communications. It becomes even more complicated at millimeter-wave (mmW) and sub-Terahertz (sub-THz) bands. The transmissions at these frequencies suffer from high atmospheric attenuation and limited penetration [5], making cell coverage highly dependent on the environment. Moreover, they require directional antennas to compensate for the atmospheric attenuation. As a result, it is generally impossible to conduct comprehensive surveys that address all possible scenarios at a given site, e.g., considering all possible BS locations, antenna configurations, and beam attributes. Accordingly, there is a critical need for data synthesis approaches that augment measured datasets from a site survey with additional coverage maps.

To tackle the aforementioned challenges, a possible approach is to employ full-length Maxwell equation solvers such as ray tracing-based computer simulations for generating RF maps. Ray tracing methods can accurately solve electromagnetic equations in complex environments [6], [7], although this comes at the cost of computational resource [8]. Furthermore, the accuracy of ray tracing hinges on precise representations of the propagation environment, encompassing 3D models of objects and obstacles alongside their respective electromagnetic properties [8]. Consequently, there is a demand for a simple, fast, and cost-efficient alternative to generate RF maps.

Different methods were proposed in the literature to address this issue, such as crowd-sourcing [9], [10] and machine learning [11]. In crowd-sourcing, it is challenging to aggregate data from heterogeneous user devices, as this may result in erroneous decisions. Moreover, traditional machine learning techniques such as generalized linear models and k-nearest neighbors cannot provide fast and accurate results and may require precise engineering of complex networks.

Recently, generative models [12] have attracted significant interest in generating additional measurements with improved diversity, thus expanding the training dataset, reducing data collection time, and saving human effort. In [13], Zou et al. used a mobile robot to collect RSS data of Wi-Fi devices in free-moving spaces for an indoor floor plan and estimated the RSS for constrained spaces by using a Gaussian process regression conditioned least-squares-based GAN. In [14], Njima et al. used GAN to generate RSS for new RPs based on a small set of experimentally collected labeled RSS for some given RPs within a floor plan. Jiang et al. proposed RMRec [15], an RF map reconstruction framework that reconstructs a high-resolution RF map from a low-resolution RF map constructed using a small number of samples. The above methods solely focus on reconstructing an RF map based on incomplete

This research was supported in part by NSF (grants # 2229386 and 1822071) and by the Broadband Wireless Access & Applications Center (BWAC). Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of NSF. The work of H. Manshaei was conducted while he was working at the University of Arizona.



(a) (b) (c)

Fig. 1: Three different types of GAN: (a) original GAN, (b) conditional GAN, and (c) information maximizing GAN.

data. However, they lack the capability to generate new RF maps when changes occur in the BS or access point (AP) configuration and position, requiring new measurements.

Conditional GAN (cGAN) can be employed to address this issue and produce RF maps for various BS configurations, utilizing the floor plan and other relevant factors as the condition. In [16], Kim et al. introduced a cGAN-based approach, which partitions the indoor space and fuses RF maps from multiple APs to create comprehensive maps. To achieve this, they divide the floor plan into AP-centered windows and utilize them as conditional inputs for the cGAN, which is built using fully connected layers. Although this technique enables the generation of RF maps for new AP locations and expedites the process, it falls short of capturing the RSS distribution for the entire floor plan associated with a specific AP. The authors in [17] proposed SURIMI, a cGAN-based model constructed with convolutional layers. This model utilizes the floor plan as a condition to effectively generate new RF maps. Seong et al. [18] proposed an unsupervised dual-radio mapping algorithm that intelligently selects between an autoencoder or GAN to generate an RF map, relying on measurements from a single reference floor. For the GAN, they utilized a 2D floor plan and the AP's position as inputs to generate RF maps for new floor plans and AP locations. The authors constructed the GAN using fully connected layers similar to that of [16]. In [19], the authors proposed a UNet-based cGAN to generate RF maps for outdoor scenarios. They conducted a comparison with RadioUnet [20], a UNet-based convolutional neural network model, and demonstrated significant performance enhancements in generating new RF maps. The aforementioned works utilize cGAN to generate new RF maps, necessitating accurate knowledge and modeling of the RF environment to guide the decision-making process. These models offer usability by enabling the generation of new RF maps for different floor plans or RF environments. However, this approach demands a substantial amount of labeled data, which is often scarce and challenging to obtain, leading to increased complexity, time, and cost. Additionally, it is important to note that none of the mentioned works present results for RF maps at the mmW and higher frequencies or with directional antennas.

This paper presents RecuGAN, an innovative approach for **R**F map **E**stimation and **C**ategorization through **U**nsupervised feature extraction using **GAN**. The advantage of RecuGAN lies in its unsupervised nature, eliminating the need for labeled data or conditional input during the RF map generation process. Our main contributions are as follows:

- Our RecuGAN framework leverages the principles of information maximizing GAN (InfoGAN) to effectively capture latent properties, such as transmitter location, antenna configuration, and center frequency, of the given RF maps, enabling unsupervised categorization and generation of new RF maps. As a result, RecuGAN does not need labeled training data, reducing the time, cost, and complexity associated with generating diverse RF maps for a given floor plan.
- We improve upon the RecuGAN objective function by integrating a customized Wasserstein GAN (WGAN) function with a gradient penalty. Moreover, we also propose a gradient-based loss function that captures the signal propagation from the BS by computing the magnitude and direction of the changes of RSS at each point in the given floor plan. This strategic combination effectively addresses challenges such as vanishing and exploding gradients during training, ensuring stability and yielding superior performance. In essence, this results in smoother convergence and improved accuracy in RF coverage map estimation and categorization tasks.
- We provide the flexibility to incorporate multiple generator networks in RecuGAN. While a single generator suffices for low-resolution RF maps, larger high-resolution RF maps benefit from the utilization of multiple generators. These generators work together, each responsible for generating a specific part of the RF map, preserving its spatial integrity. The beauty of this approach lies in the fact that the generators can have distinct neural network configurations and produce RF maps with varying shapes, offering adaptability and versatility.
- We train RecuGAN using RF maps obtained through both experimental measurements and ray tracing simulations and subsequently generate synthetic RF maps for previously unknown coverage scenarios. We evaluate the performance of RecuGAN in generating diverse RF maps and report the results in Section V. Moreover, we show the categorization capability of RecuGAN by comparing RF maps for each catergory using the structural similarity index metric (SSIM) [21]. Furthermore, we conduct a comparative evaluation with the UNet-based cGAN model described in [19], demonstrating that the RF maps produced by RecuGAN achieve a mean average percentage error (MAPE) of 1.18%, outperforming the 2.5% MAPE achieved by the UNet-based cGAN model.

The subsequent sections of this paper are structured as follows. Section II provides an overview of various architectures of GANs. Section III introduces the RecuGAN model. The training procedure for the RecuGAN is presented in Section IV. Section V presents the performance evaluation of RecuGAN, followed by conclusions in Section VI. Throughout the paper, we use the notation $X \sim P_X$ to refer to a random variable X whose probability density distribution is P_X.

II. Preliminaries: An Introduction to GANs

The original GAN [12] comprises two neural networks: a generator (G) and a discriminator (D), as depicted in Fig. 1(a). G and D are trained simultaneously via an adversarial learning technique. The idea behind this technique is a two-player zero-sum minimax game played between G and D. During the learning phase, the game reaches a nash equilibrium, where each player cannot unilaterally increase their payoff. Considering the Jensen-Shannon divergence metric, we can formalize the objective of this minimax game as follows:

$$\min_G \max_D V(G, D) = \min_G \max_D \Big(\mathbb{E}_{x \sim P_{data}}[\log(D(x))] + \mathbb{E}_{z \sim P_z}[\log(1 - D(G(z)))] \Big) \quad (1)$$

where x is real data samples and z is noise. $G(z)$ takes a prior noise distribution (P_z) as input and maps it to an approximate training data distribution (P_g). $D(x)$ is a mapping of the input data distribution P_{data} into the interval of $[0, 1]$, which represents the probability that the sample is real and has not been produced by the generator. Using the Jensen-Shannon divergence metric, the game is played at the nash equilibrium if both D and G are optimal. At this point, the minimax and maximin values are equal to $-\log 4$ [12].

With a cGAN (see Fig. 1(b)), extra information c is provided to the generator along with z [22]. c is a discrete variable that represents attributes of the generated data. In other words, a cGAN learns disentangled representations of data by utilizing c as a conditioning factor for the generator model. The objective of a cGAN model is given as:

$$\min_G \max_D V(G, D) = \min_G \max_D \Big(\mathbb{E}_{x \sim P_{data}}[\log(D(x|c))] + \mathbb{E}_{z \sim P_z}[\log(1 - D(G(z|c)))] \Big). \quad (2)$$

InfoGAN (see Fig. 1(c)) is an information-theoretic extension of GANs. The objective for an InfoGAN is given by:

$$\min_G \max_D \Big(V(G, D) - \lambda I(c; G(z, c)) \Big) \quad (3)$$

where λ is a hyperparameter and V is the objective function of a GAN as presented in Eq. (1), except that in this case, the InfoGAN discriminator does not take c as input. InfoGAN can learn disentangled representations of the data distribution in an unsupervised fashion [23]. This is done by maximizing the mutual information between the conditional variable c and the data generated with this conditional variable, i.e., $I(c; G(z, c))$. In practice, we cannot easily maximize the mutual information I as we need to know the posterior probability of the latent code c. As a result, InfoGAN uses a third network (besides G and D) called a recognition network (or Q-network) to determine an auxiliary distribution $Q(c|x)$, which is used to approximate $P_{(c|x)}$. In an InfoGAN, Q and D share similar convolutional layers, except for a fully connected final layer. The final layer provides the output parameters for the conditional distribution $Q(c|x)$. As a result, InfoGAN has slightly more computational cost than the original cGAN.

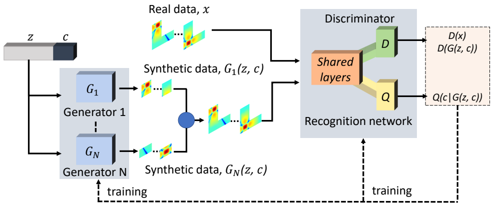


Fig. 2: General architecture of RecuGAN.

In this paper, RecuGAN utilizes InfoGAN's ability to extract distinct features from data distributions. By representing the RSS coverage of a given BS location as latent properties within the latent code space, we can categorize synthetic RSS maps into various coverage regions. Consequently, we can generate diverse RF maps for any specified region by manipulating the values of the latent code.

III. RecuGAN: Architecture and Design

A. RecuGAN Architecture

Fig. 2 presents a general architecture of RecuGAN, which is composed of three types of neural networks: (i) N generators $G_i, i = 1, \ldots, N$, (ii) a discriminator D, and (iii) a recognition network Q. In general, each generator G_i is associated with a set of trainable hyperparameters θ_i. Likewise, D and Q are associated with the sets of hyperparameters ϕ and ψ, respectively. For simplicity, our notation does not show the dependence on the hyperparameters. All N generators take a common input, which is a concatenation of a noise vector z and a discrete latent code c of length L, and pass it through stacks of layers to generate a synthetic RF map. In general, each generator can be assigned to synthesize a portion of the RF map, e.g., the coverage for one room in a floor plan, and the N synthetic outputs can then be combined and presented as input to both D and Q. With some abuse of notation, we denote the combined outputs of the N generators as $G(z, c)$. The vector z is sampled from a multi-variant Gaussian distribution (i.e., $z \sim \mathcal{N}_k(0, 1)$, where k is the dimension of z and c is sampled from a categorical distribution. Recall that the length of c (i.e., L) equals the number of categories i.e., coverage regions the Q-network would detect for these synthetic RF maps. The discriminator D tries to differentiate between real and synthetic RF maps. It provides feedback to the generator regarding the similarity of the generated data to real samples.

Fig. 3 depicts the architecture of G, D, and Q-networks utilized in RecuGAN with a single generator. All 2D convolutions are implemented using a 5×5 kernel size. As seen in the figure, the Q-network has six common convolutional

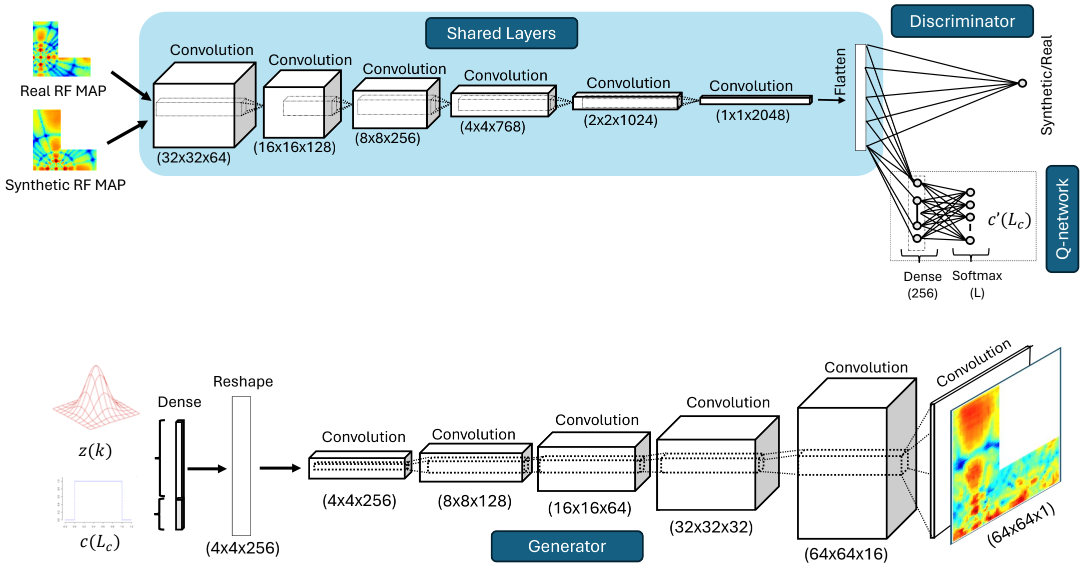


Fig. 3: Generator (G), Discriminator (D), and recognition (Q-network) models in the RecuGAN.

layers with the discriminator. In other words, the Q-network extended from D with the same shared hidden layers. In this approach, we feed the output of the flattened layer of the discriminator into the last layer of the Q-network. A softmax activation function is then applied to generate the output of the Q-network, representing the latent code c. For RecuGAN with multiple generator networks, we modify the architecture of G based on the size of the output RF map. Consequently, we can produce RF maps with diverse shapes and complexities with high resolutions.

B. RecuGAN Objective Function

To train RecuGAN, we train each component of the neural network models described above. As with all GAN models, we use an objective function to train the generator and discriminator. As RecuGAN incorporates a recognition network aimed at maximizing mutual information, we establish the loss objective following a similar formulation as defined in Eq. (3), where λ represents the weight of mutual information in the loss function and is set to a value close to one. We conducted a series of simulations to determine the optimal $V(G,D)$ in our experiments. Based on our experiments, the Wasserstein GAN [24] with a gradient penalty [25] exhibited the best performance for the RecuGAN model. This new objective function applies a soft version of the constraint through a penalty function on the gradient norms for random samples [25]:

$$\begin{aligned} V^*(G,D) &= \mathbb{E}_{\tilde{x}\sim P_g}[D(\tilde{x})] - \mathbb{E}_{x\sim P_{data}}[D(x)] \\ &+\beta\mathbb{E}_{\hat{x}\sim P_{\hat{x}}}[(\| \bigtriangledown_{\hat{x}} D(\hat{x}) \|_2 - 1)^2] \end{aligned} \tag{4}$$

where $P_{\hat{x}}$ is computed by interpolating the real and synthetic images. Specifically, to calculate $P_{\hat{x}}$, we first compute the difference between real and synthetic images and then multiply it with normal noise. The resulting value is then added to the real image, yielding $P_{\hat{x}}$. In our experiments, we found that setting β to 10 worked well for generating RF maps. It is important to note that the modifications mentioned above lead to more stable gradients, preventing them from either vanishing or exploding during the training of our complex network [25].

The RecuGAN loss function incorporates the mutual information between the latent code c and the generated RF map, $G(z,c)$ [23]. This means that the objective function takes into account the mutual information between the two, with the aim of minimizing the overall loss by minimizing the GAN loss V and maximizing the mutual information I. By maximizing the mutual information, the generator ensures that the information in the latent code is not lost during the generation of a new synthetic RF map. However, directly maximizing the mutual information is not feasible since we cannot access the posterior $P_{(c|x)}$, as noted in [23]. Therefore, the authors in [23] suggested using a lower bound:

$$\begin{aligned} I(c;G(z,c)) &= H(c) - H(c|G(z,c)) \\ &\geq \mathbb{E}_{x\sim P_{G(z,c)}}[\mathbb{E}_{y\sim P_{(c|x)}}[\log Q(y|x)]] + H(c) \end{aligned} \tag{5}$$

where $Q(y|x)$ is an auxiliary distribution to approximate $P_{(c|x)}$. This approach is named Variational Information Maximization. In RecuGAN, we also modify the mutual information function. Specifically, we calculate the mutual information I between c and c' (i.e., the output of the Q-network) as the sum of the entropy of c, denoted by $H(c)$, and the conditional entropy of c' given c, denoted by $H(c'|c)$, as shown in the following equation:

$$I(c,c') = H(c) - H(c'|c). \tag{6}$$

Since the entropy of c remains constant, $H(c)$ can be omitted from the above equation. The conditional entropy, $H(c'|c)$ can be then defined as $\mathbb{E}_{c\sim P_{(c)}}[\log(c')]$ where c' is the output of

Q-network. To obtain c', we feed z and c into the generator, and then its output is fed to the Q-network, which generates c'. Thus, the mutual information I can be expressed as follows:

$$I(c, c') = \mathbb{E}_{x \sim P_{G(z,c)}}[\mathbb{E}_{c' \sim P_{(c|x)}}[\log(c')]] \tag{7}$$

By maximizing the mutual information between c and c', RecuGAN can generate new RF maps and categorize them in an unsupervised manner based on RSS distribution for specific BS locations.

C. Gradient Loss Function

By inspecting the RF maps generated using directional antennas, we observe that the distribution of the RSS depends on the antenna pattern. The signal propagates smoothly along the antenna's boresight, but there are abrupt changes in the RSS between the boresight and the null direction. To capture these effects, we use a gradient loss (L_{GL}). Basically, we use Sobel filer to obtain the gradient of ground truth RF maps, x_i and synthesized RF maps, $\tilde{x}_i$, i.e., ∇x_i and $\nabla \tilde{x}_i$. The magnitude of the gradient tells us how quickly the RSS is changing, while the direction of the gradient tells us the direction in which the RSS is changing most rapidly. To measure L_{GL}, we determine the changes in the magnitude and direction between ∇x_i and $\nabla \tilde{x}_i$ using KL-divergence (KL) and cosine similarity (CS), respectively. The KL-divergence between the gradients will give us the changes in the magnitude, while the cosine similarity will provide us with the changes in the direction. Thus we can write L_{GL} as

$$L_{GL}(x, \tilde{x}) = \frac{1}{n} \sum_{i=1}^{n} \Big(\text{KL}(\nabla x_i, \nabla \tilde{x}_i) + \text{CS}(\nabla x_i, \nabla \tilde{x}_i) \Big). \tag{8}$$

Finally, given the above calculation, we can express the objective function of RecuGAN as:

$$\begin{aligned} V_{RecuGAN} = \min_{G} \max_{D} \Big(& \mathbb{E}_{\tilde{x} \sim P_g}[D(\tilde{x})] - \mathbb{E}_{x \sim P_{data}}[D(x)] \\ & + \beta \mathbb{E}_{\hat{x} \sim P_{\hat{x}}}[(\parallel \nabla_{\hat{x}} D(\hat{x}) \parallel_2 - 1)^2] \\ & + \lambda_{gl} \mathbb{E}_{x \sim P_{data},\ \tilde{x} \sim P_g}[L_{GL}(x, \tilde{x})] \\ & - \lambda \mathbb{E}_{x \sim P_{G(z,c)}}[\mathbb{E}_{c' \sim P_{(c|x)}}[\log(c')]] \Big) \end{aligned} \tag{9}$$

where λ_{gl} is the weight of L_{GL}.

IV. Training Challenges and Solutions

In each training step of RecuGAN, we randomly choose a batch of real RF images from our dataset to train the discriminator, generator(s), and recognition networks. We feed z and c to the generator to produce synthetic RF maps. Consequently, we will feed the real and synthetic RF maps to the discriminator to calculate the loss and the gradient penalty terms. The objective function of the discriminator, $V(G, D)$ is then computed by Equation (4), and in every step of training, the objective function is maximized, as shown in Fig. 4(b).

The generator and the recognition networks are trained simultaneously in the following steps. z and c are randomly produced again, and we feed them to the generator to make a new set of synthetic RF maps. In this step, the weights of the discriminator model are kept fixed. We compute the objective function using the discriminator outputs and the recognition network by Equation (9). Fig. 4(c) shows the loss function of the recognition network calculated by Equation (7). The generator loss is computed using Equation (9). As depicted in Figs. 4(a) and 4(d), we could minimize the loss in each epoch of the training process, and finally, the RecuGAN is trained successfully by a set of available RF maps. Note that in our experiment, for each update of the generator, the discriminator is trained three times on a combination of real and synthetic RF maps. This approach helped us to make the training more stable.

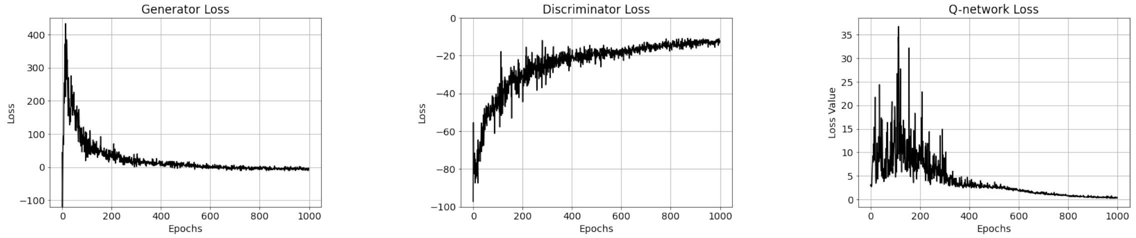
TABLE I: Simulation Datasets

Dataset	Freq.	Ant. Config.	Floor Plan	Ant. Dir.
$\mathbb{D}_{sim1}$	28 GHz	8×8 UPA	L-shaped room	Z-axis
$\mathbb{D}_{sim2}$	5 GHz	4×4 UPA	L-shaped room	Z-axis
$\mathbb{D}_{sim3}$	60 GHz	4×4 UPA	L-shaped room	Z-axis
$\mathbb{D}_{sim4}$	28 GHz	4×4 UPA	Office room	Z-axis
$\mathbb{D}_{sim5}$	28 GHz	4×4 UPA	Large office floor	Z-axis
$\mathbb{D}_{sim6}$	28 GHz	4×4 UPA	L-shaped room	X-axis

Training the recognition network was also a challenging task because the loss of the recognition network remained constant during the training phase. Despite the generation of desired images, we noticed that the loss of the Q-network remained constant and by progressing through the training phase, we did not see any change in the Q-network loss. Without converging the loss of the Q-network to the global optimum, the model cannot find any meaningful relationship between the generated images and latent code c. After analysis of gradients for Q-network layers, we observed the effect of *Dying Relu* [26] in our model. The dying Relu occurs when the network's neurons are inactive and generate zero outputs, as shown in [26]. In this situation, no improvement is obtained in the learning process. To address this issue, we employed the Glorot Normal method [27] to initialize the weights of the Q-network. This technique effectively regulates activation and backpropagation gradient variance and prevents the occurrence of dying neurons. The following section will present how the trained RecuGAN can generate high-quality RF maps for given attributes.

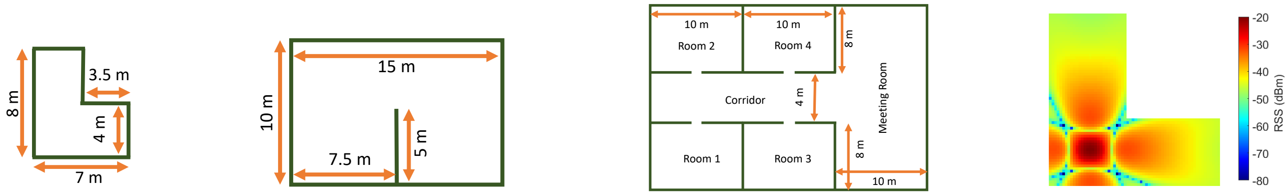
V. Evaluation

One of the important characteristics of RecuGAN is that it can learn the latent features of the RF maps (in our case RSS coverage distribution) in an unsupervised manner. Based on these latent features, it synthesizes new RF maps and categorizes them. In this section, we first evaluate the performance of RecuGAN in synthesizing diverse RF maps and categorizing them. We then evaluate the quality of the synthesized RF maps in terms of mean average percentage error (MAPE) and finally compare them with those generated using the UNet-based cGAN model presented in [19].



(a) (b) (c)

Fig. 4: Value of loss functions for: (a) generator, (b) discriminator, and (c) Q-network.



(a) (b) (c) (d)

Fig. 5: 2D view of (a) L-shaped room, (b) office room, and (c) large office. (h) Representative RF map of an L-shaped room using 4×4 UPA when BS is at position x=3m, y=3m.

A. Ray Tracing-based Simulated Dataset

In this paper, we rely on MATLAB-based ray tracing simulations to generate the simulation-based datasets. We use the Shoot and Bouncing Ray (SBR) method [28] to generate the propagation paths between the BS and Rx. We generate indoor RF maps at different frequencies and consider three indoor scenarios: an L-shaped room measuring 7×8 m^2, an office room measuring 15×10 m^2 with a partition in the middle, and a large office floor measuring 30×20 m^2 with multiple rooms and corridors. The large office floor provides a complex structure similar to the experimental area described in Section V-B. All the scenarios include the line of sight (LOS) and non-line of sight (NLOS) regions. The floor plans are illustrated in Fig. 5(d,e,f). In addition, we consider the surface material of the floor, walls, and ceiling as brick, with a relative permittivity of 3.75 Farad/meter and conductivity of 0.038 Siemens/meter. These parameters are easily configurable and updated as needed.

We consider an Rx with an isotropic antenna that produces an omnidirectional pattern. On the other hand, we consider uniform planer arrays (UPA) of different sizes with patch antenna elements for the BS. The considered UPA configurations are 4×4 and 8×8. Moreover, we study three carrier frequencies: 5 GHz, 28 GHz, and 60 GHz. The BS is placed at 3 meters from the ground with the broadside of the antenna facing downwards, i.e., from the ceiling towards the floor along the Z-axis (Fig. 5(g)) for all the datasets except one where the broadside of the antenna is along the X-axis. We place the Rx at 1.5 meters. Generally, the BS does not implement beam tracking indoors, so we fix the BS beam direction.

In order to generate an RF map, we divide the room area into a 64×64 grid, with each grid of size 0.014 m^2 for the L-shaped and 0.037 m^2 for the office room. In the case of the large office floor, we divide it into 128×128 grids, with each grid of size 0.036 m^2. We then place the BS in each grid tile and measure the RSS value. The RSS values are then used to generate an RF map, as depicted in Fig. 5(h). As a result, for the L-shaped and office room, the dataset consists of 4096 data points each; for the large office floor, it consists of 16384 data points. We list the configuration for each dataset in Table I.

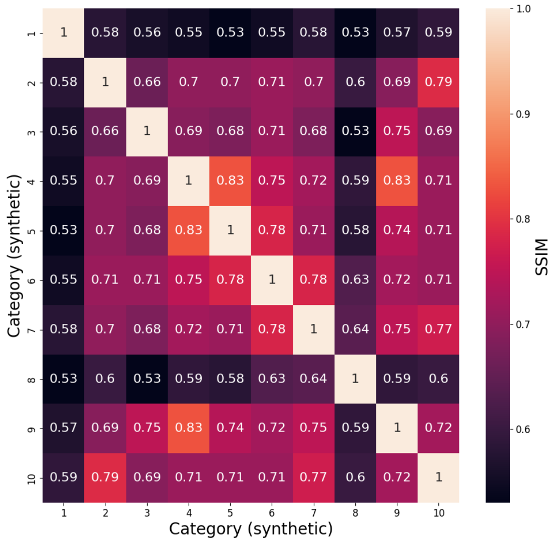


Fig. 6: SSIM matrix showing the categorization capability of RecuGAN.

1) Categorizing Synthetic RF Maps: To evaluate the effectiveness of RecuGAN in producing distinct categories of synthetic RF maps, we train the model with the datasets in Table I. For our experiments, we set $k = 128$. We also set $L = 30$ for $\mathbb{D}_{sim1}$ and $L = 10$ for the remaining datasets. For all but $\mathbb{D}_{sim6}$, the simulated RF maps were each enclosed within an image of dimensions 64×64 pixels, with certain pixels set to zero if no signal is present. At such resolution, we found that a single generator is adequate to synthesize representative RF maps. Once the training is complete, we generate 100 synthetic RF maps for each category. We calculate the average of the 100 RF maps per category and compute the SSIM among them. SSIM is a metric used to measure the similarity between two images by considering structural information, luminance, and contrast. It ranges from -1 to 1, where 1 indicates a perfect match and -1 indicates no match. Fig. 6, depicts the SSIM matrix for $\mathbb{D}_{sim_4}$ with 10 categories. The SSIM matrix for other datasets are similar but are not shown due to limitation of space. The data presented in the figure reveals that the SSIM value is consistently 1 when a category is compared with itself. However, when compared to other categories, the SSIM value decreases. This observation indicates that the RF maps produced by RecuGAN exhibit diversity within a given category compared to RF maps from different categories.

2) Diversity of Synthetic RF Maps: Fig. 7 depicts the diversity of RF maps generated by RecuGAN for each category for datasets $\mathbb{D}_{sim_1}$ to $\mathbb{D}_{sim_4}$ using a single generator. Moreover, Fig. 8 demonstrates the generation of synthetic RF maps for datasets $\mathbb{D}_{sim5}$ and $\mathbb{D}_{sim6}$ using multiple generator networks. As highlighted in Section V-A, the RF maps in $\mathbb{D}_{sim6}$ are of dimension 128×128 pixels. Consequently, we employ four generators, each producing parts of the large floor plan (Fig. 5(f)) simultaneously, where the output RF map dimension of each generator is 64×64. The outputs from these generators are combined to form the complete RF map represented in Fig. 8(a). Conversely, in Fig. 8(b), we use the dataset $\mathbb{D}_{sim5}$ containing RF maps for L-shaped room of dimension 64×64 pixels. Here, we employ two generators, one synthesizing an RF map of 32×64 and the other generating an RF map of 32×32. This demonstrates RecuGAN's capability to incorporate multiple generators to generate RF maps for complex and larger floor plans efficiently.

3) Comparison of RecuGAN and UNet-cGAN: We compare RecuGAN with the GAN model described in [19]. Basically, [19] uses a cGAN architecture which is a derivative of the pix2pix GAN [29]. The generator network is constructed using a UNet, which is based on convolutional neural networks. The discriminator network consists of stacks of convolutional blocks. The authors also utilize regularization techniques to improve the performance of their cGAN model. A detailed description of their model is presented in [19]. Since the model uses a UNet to construct their GAN, we call the model UNet-cGAN for future reference. It is important to note that UNet-cGAN takes the RF environment (in our case, the floor plan) and the location of the BS as conditional inputs.

We train the UNet-cGAN and RecuGAN with 3096 samples of $\mathbb{D}_{sim4}$ and keep aside 1000 RF maps for testing. For each RF map in the test dataset, we calculate the MAPE of the RF maps generated by the UNet-cGAN. MAPE measures the percentage difference between synthesized RF maps and test RF maps, giving an indication of how well the GAN model performs on average. The MAPE calculation formula is expressed as follows:

$$\text{MAPE} = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{x(i) - \tilde{x}(i)}{x(i)} \right| \tag{10}$$

where $x(i)$ and $\tilde{x}(i)$ represents the i^{th} pixel of the test and synthesized RF map, respectively. n is the total number of pixels in the RF maps being evaluated.

The average MAPE obtained by UNet-cGAN is 2.5%. For RecuGAN, we generate 200 RF maps for each of the 10 categories, totaling 2000 RF maps. Next, we use KL divergence to find the closest RF map within the RecuGAN-synthesized RF maps for each RF map in the test dataset. Finally, we compute the average MAPE values. RecuGAN achieves a MAPE value of 1.18%, which is 1.32% better than the UNet-cGAN. Though the performance improvement is not that high, RecuGAN can generate new RF maps that were not exposed during training with very high quality without the need for any labeled data.

Fig. 9 depicts samples of the test RF maps and synthetic RF maps generated by both RecuGAN and UNet-cGAN. As seen from the figure, UNet-cGAN fails to capture the RF propagation due to the directional antenna at mmW frequency. In contrast, RecuGAN captures this effect by utilizing the gradient-loss function described in Section III-C.

4) Time Complexity Comparison: In this paper, we investigate the time complexities of RecuGAN and MATLAB-based ray tracing simulation. RecuGAN's time complexity hinges on its network architecture and input size, while ray tracing simulation's complexity is predominantly determined by factors such as scene geometry, material intricacies, and ray count. To assess their time complexities, we conducted experiments averaging the time required to generate 1000 RF maps for both RecuGAN and ray tracing simulation. Once trained, RecuGAN exhibits swift generation times, averaging 0.00285 seconds, markedly outpacing ray tracing simulation, which averages 14.15 seconds per simulation.

B. Experimental Dataset

The experimental dataset ($\mathbb{D}_{exp}$) [30] comprises RSS values for Wi-Fi devices measured on the 4th floor of the DJ building of Huazhong University of Science and Technology. The measurement area comprises 6 classrooms and a corridor as shown in Fig. 5(a), with odd-labeled classrooms measuring $10.5 \times 9.56\ \text{m}^2$ and even-labeled classrooms measuring $10.5 \times 7.76\ \text{m}^2$. The size of the corridor is $32.6 \times 3.6\ \text{m}^2$. The measurement area is divided into 1890 RPs, where the distance between two RPs is set to 0.6 meters. $\mathbb{D}_{exp}$ includes RSS values from 695 APs. At each RP, 10 RSS values were collected for each AP. In cases where RSS data was not received from an AP at a given RP, its value was set to 0.

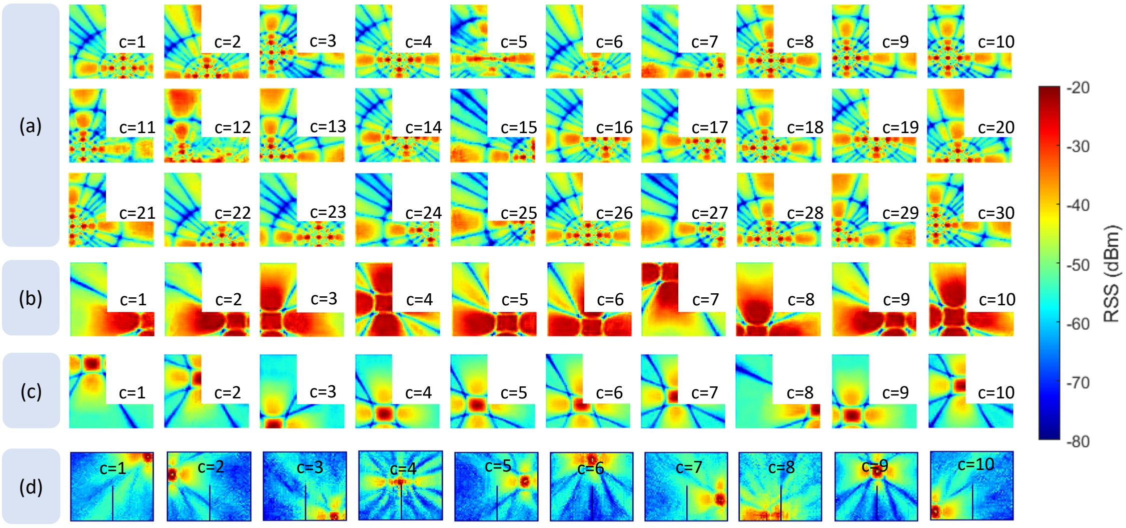


Fig. 7: RecuGAN synthesized RF maps based on simulated training data with one generator: (a) 28 GHz and 8×8 UPA; (b) 5 GHz and 4×4 UPA; (c) 60 GHz and 4×4 UPA for L-shaped floor plan; and (d) 28 GHz and 4×4 UPA for small office.

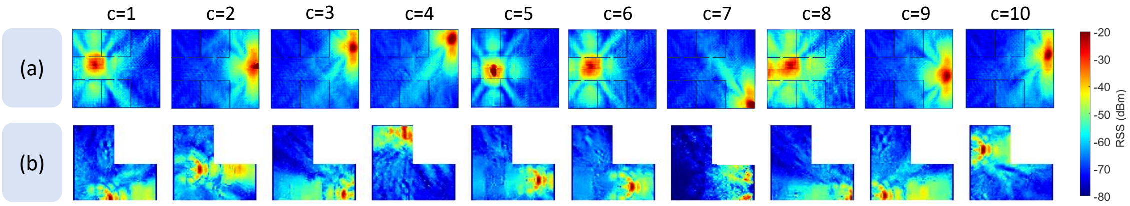


Fig. 8: RecuGAN synthesized RF maps at 28 GHz and using 4×4 UPA by incorporating multiple generators: (a) four generators were used, each synthesizing a 64×64 part of the 128×128 RF map of the large floor plan, and (b) two generators were used, one generator synthesizes 32×32, and the other synthesizes 32×64, generating the complete L-shape room.

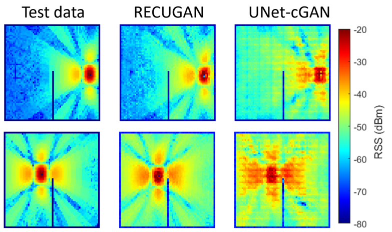
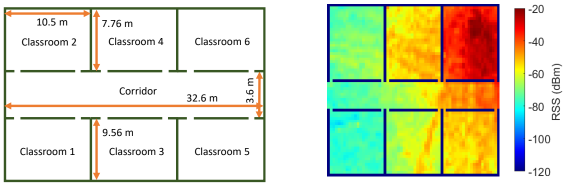


Fig. 9: Comparison between synthetic RF maps generated by RecuGAN and UNet-cGAN given a test RF map.



(a) (b)

Fig. 10: (a) Floor plan used to collect experimental Wi-Fi measurements and (b) a representative RF map generated using the Wi-Fi measurements.

Before training our model with the experimental dataset, we needed to perform some data preprocessing. First, we averaged the RSS values for each AP at every RP. Secondly, we adjusted the 0 values (indicating no AP detected) to -120 dBm based on the observed range of the highest and lowest RSS values (-20 dBm to -120 dBm). Finally, during the visual analysis of the RF maps for each AP, we noticed some RF maps were mostly empty (Fig. 5(b)), with RSS values only at a few RPs. We discarded these RF maps, resulting in approximately 403 RF maps. A representative RF map containing valid RSS values for all RPs is shown in Fig. 5(c).

We train RecuGAN with $\mathbb{D}_{exp}$ for 7 categories. To evaluate RecuGAN generated RF maps, we calculate the MAPE between the synthetic and the RF maps. Fig. 11 depicts a synthetic RF map and the closest experimental test RF map in terms of RSS distribution obtained using KL divergence for specific values of c_i. The average MAPE obtained by RecuGAN for the experimental dataset is 3.18%. This indicates that the synthetic RF maps are a good representation of the training dataset.

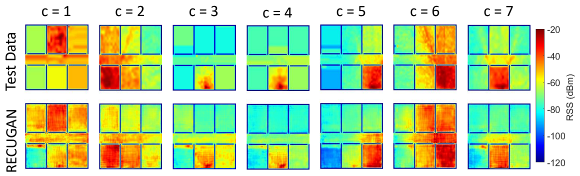


Fig. 11: Comparison between synthetic RF maps generated by RecuGAN and experimental RF maps.

VI. Conclusion

In this paper, we presented RecuGAN, a novel solution that addresses the challenges associated with RF map generation. We applied the principles of information maximizing GAN with and incorporated a modified gradient penalty-based WGAN objective function. We also implemented a gradient-based loss function to capture the effect of RF propagation for a given environment when generating synthesized RF maps. Our experiments with both real and simulated datasets indicated that RecuGAN can synthesize diverse and high-quality RF maps in an unsupervised manner. Specifically, it outperformed UNet-cGAN models, as evidenced by better MAPE values. By providing accurate RF maps efficiently and cost-effectively, RecuGAN contributes to improved network planning and optimization in wireless communication systems.

Currently, RecuGAN can only capture the RSS coverage distribution for a given Tx location as the latent feature and categorize RF maps accordingly. As future work, we intend to evaluate the performance of our model for other RF map attributes. Specifically, we will explore how the latent code can capture the transmission frequency and antenna configuration. Moreover, we plan to formulate a process to determine the optimal number of categories for a given dataset.

References

[1] D. Romero, S.-J. Kim, G. B. Giannakis, and R. López-Valcarce, "Learning power spectrum maps from quantized power measurements," *IEEE Transactions on Signal Processing*, vol. 65, no. 10, pp. 2547–2560, 2017.

[2] C. Wu, Z. Yang, Y. Liu, and W. Xi, "Will: Wireless indoor localization without site survey," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, pp. 839–848, 2013.

[3] B. Huang, Z. Xu, B. Jia, and G. Mao, "An online radio map update scheme for wifi fingerprint-based localization," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6909–6918, 2019.

[4] Q. Niu, Y. Nie, S. He, N. Liu, and X. Luo, "RecNet: A convolutional network for efficient radiomap reconstruction," in *Proc. IEEE Int. Conf. on Communications (ICC)*, 2018, pp. 1–7.

[5] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-wave cellular wireless networks: Potentials and challenges," *Proc. of the IEEE*, vol. 102, no. 3, pp. 366–385, 2014.

[6] V. Erceg, S. Fortune, J. Ling, A. Rustako, and R. Valenzuela, "Comparisons of a computer-based propagation prediction tool with experimental data collected in urban microcellular environments," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 4, pp. 677–684, 1997.

[7] M. Lecci, P. Testolina, M. Polese, M. Giordani, and M. Zorzi, "Accuracy versus complexity for mmwave ray-tracing: A full stack perspective," *IEEE Transactions on Wireless Communications*, vol. 20, no. 12, pp. 7826–7841, 2021.

[8] D. Romero and S.-J. Kim, "Radio map estimation: A data-driven approach to spectrum cartography," *IEEE Signal Processing Magazine*, vol. 39, no. 6, pp. 53–72, 2022.

[9] C. Wu, Z. Yang, and C. Xiao, "Automatic radio map adaptation for indoor localization using smartphones," *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 517–528, 2018.

[10] Z. Yin, C. Wu, Z. Yang, and Y. Liu, "Peer-to-peer indoor navigation using smartphones," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1141–1153, 2017.

[11] S. Mohammadjafari, S. Roginsky, E. Kavurmacioglu, M. Cevik, J. Ethier, and A. B. Bener, "Machine learning-based radio coverage prediction in urban environments," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2117–2130, 2020.

[12] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. of the 27th Int. Conf. on Neural Information Processing Systems (NIPS)*. MIT Press, December 2014, pp. 2672–2680.

[13] H. Zou, C.-L. Chen, M. Li, J. Yang, Y. Zhou, L. Xie, and C. J. Spanos, "Adversarial learning-enabled automatic wifi indoor radio map construction and adaptation with mobile robot," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6946–6954, 2020.

[14] W. Njima, M. Chafii, A. Chorti, R. M. Shubair, and H. V. Poor, "Indoor localization using data augmentation via selective generative adversarial networks," *IEEE Access*, vol. 9, pp. 98 337–98 347, 2021.

[15] W. Jiang, Q. Niu, S. He, and N. Liu, "Adaptive radio map reconstruction via adversarial wireless fingerprint learning," *Neural Computing and Applications*, pp. 1–18, 2023.

[16] W.-Y. Kim, S.-H. Tae, and D.-H. Seo, "Access-point centered window-based radio-map generation network," *Sensors*, vol. 21, no. 18, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/18/6107

[17] D. Quezada-Gaibor, J. Torres-Sospedra, J. Nurmi, Y. Koucheryavy, and J. Huerta, "Surimi: Supervised radio map augmentation with deep learning and a generative adversarial network for fingerprint-based indoor positioning," in *Proc. IEEE 12th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2022, pp. 1–8.

[18] J. H. Seong and D. H. Seo, "Selective unsupervised learning-based Wi-Fi fingerprint system using autoencoder and GAN," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1898–1909, 2020.

[19] S. K. Vankayala, S. Kumar, I. Roy, D. Thirumulanathan, S. Yoon, and I. S. Kanakaraj, "Radio map estimation using a generative adversarial network and related business aspects," in *Proc. of 24th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, 2021, pp. 1–6.

[20] R. Levie, Ç. Yapar, G. Kutyniok, and G. Caire, "Radiounet: Fast radio map estimation with convolutional neural networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 4001–4015, 2021.

[21] D. Brunet, E. R. Vrscay, and Z. Wang, "On the mathematical properties of the structural similarity index," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1488–1499, 2012.

[22] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[23] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," *Advances in Neural Information Processing Systems*, pp. 2180–2188, 2016.

[24] J. Adler and S. Lunz, "Banach wasserstein GAN," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[25] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein GANs," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[26] L. Lu, Y. Shin, Y. Su, and G. E. Karniadakis, "Dying relu and initialization: Theory and numerical examples," *arXiv preprint arXiv:1903.06733*, 2019.

[27] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. of the thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[28] H. Ling, R.-C. Chou, and S.-W. Lee, "Shooting and bouncing rays: Calculating the rcs of an arbitrarily shaped cavity," *IEEE Transactions on Antennas and propagation*, vol. 37, no. 2, pp. 194–205, 1989.

[29] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[30] A. M. Rajab and B. Wang, "Wifi fingerprinting radio map database for indoor localization," *IEEE Dataport*, 2023. [Online]. Available: https://dx.doi.org/10.21227/sdpr-fh91