

DA-cGAN: A Framework for Indoor Radio Design Using a Dimension-Aware Conditional Generative Adversarial Network

Chun-Hao Liu* Hun Chang* Taesuh Park†

Global AI Accelerator (GAIA), Ericsson Inc.

{chun-hao.liu, hun.chang, taesuh.park}@ericsson.com

Abstract

A novel “physics-free” approach of designing indoor radio dot layout for a floor plan is introduced by formulating it as an image-to-image translation problem and solved with customized dimension-aware conditional generative adversarial networks (DA-cGANs). The proposed model generates a desirable radio heatmap and its respective radio dot layout from a given floor plan with wall types, physical dimension, and macro-cell interference, by learning from the accumulated indoor radio designs by human experts. Considering the nature of radio propagation, two new loss functions and a two-stage training strategy are proposed for the generator to learn the right direction of signal propagation and precise dot locations, in addition to a sectional analysis for dealing with large floor plans. Experimental results show that the new model is effectively generating acceptable dot layout designs and that dimension-awareness is a key enabler for this type of prediction.

1. Introduction

Wireless connectivity becomes more important than ever as the era of the Internet of Things (IoT) has arrived. To provide better user experience, telecommunications companies are actively proposing solutions to increase cell coverage, enhance throughput, mitigate interference, and reduce latency. The current most popular mobile communication standard, i.e., the fourth generation (4G) Long Term Evolution (LTE) has been proven to be a good solution to provide high throughput/coverage for users. Recently, not only outdoor scenarios adopt the LTE solution, but also in-building scenarios start to use it, due to large amount of indoor usage taking 87% of usage time in the U.S. [12] and generating 70% of global mobile data traffic [16]. LTE has the capability to handle large amount of user equipment (UE) and provide good coverage, hence it is well-fitted to the large

in-building scenario. In addition, it can be seamlessly upgraded to the fifth generation (5G) services which can provide much higher throughput and ultra-low latency in the near future. Nowadays, the LTE radio dot system (RDS) has been proposed and implemented inside many buildings across the world [16], and we can foresee that they are upgraded synchronously with the incoming 5G services to avoid mismatched user experience between the indoor and outdoor systems.

However, there are two pain points to deploy the RDS inside a new building, i.e., both time and cost consuming. To figure out the optimal number and layout of radio dots for a floor, a time-consuming process called site survey is required, consisting of the following steps. First, we need to measure the wall type, e.g., concrete, glass, metal, etc. Then we need to measure macro-cell interference, i.e., the signal transmitted from nearby outdoor macro base stations, in order to calculate their interference on indoor locations. Once we have all surveys being done, an experienced radio designer determines the best radio dot locations by observing the signal propagation heatmap generated by a radio frequency (RF) planner software, in order to maximize the coverage efficiency for each floor [16]. Figure 1 shows the steps of a typical design process from a given raw floor plan to its RDS deployment. Figure 1(a) shows an example of a raw floor plan. In Figure 1(b), wall type information is added as the colored wall. The circle dots in Figure 1(c) are the radio dots being placed by a designer manually. Figure 1(d) represents the heatmap of the reference signal received power (RSRP) level, which is a type of received power measurement from the radio dots according to the given dot layout in Figure 1(c). Note that darker pixel indicates stronger RSRP. The whole design process is not only costly but tedious, and it can take up to tens of days to be completed for one large building. Therefore, this deployment methodology can not be scaled to large number of buildings.

Several works were done to support indoor radio dot planning. Among three components of the RDS (radio dots, radio heads, and baseband units), the authors in [2] pro-

*Equal contributions.

†Corresponding author.

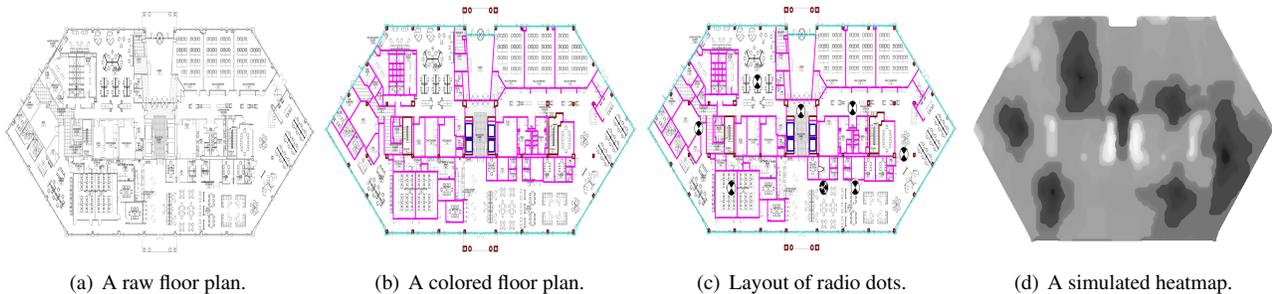


Figure 1. Typical design steps to generate a heatmap from a raw floor plan.

posed a clustering algorithm to assign baseband units and radio heads to the given radio dots. Another work [23] proposed to solve radio head placement problem by minimizing the total amount of wiring resources. However, none of the above works challenged to estimate radio dot locations, assuming that the hard part is given by human designers. For complementing their missing piece as well as tackling the two pain points for an indoor RDS, we propose to formulate the design process as an image-to-image translation problem and apply generative adversarial networks (GANs) [6]. Specifically, we propose a dimension-aware conditional GAN (DA-cGAN) to solve our heatmap prediction problem from a given floor plan. The contribution of our work is fourfold:

1. We propose a new cGAN architecture to learn the heatmap based on the physical dimension of a floor plan;
2. We propose two new loss functions with multi-stage training strategy to learn the signal propagation pattern and precise dot locations;
3. A sectional analysis scheme is proposed to deal with large floor plans;
4. We take macro-cell interference into consideration and it is served as another input branch to be merged with our DA-cGAN.

The rest of the paper is organized as follows. The DA-cGAN architecture is introduced in Section 2. The loss functions being used to train the DA-cGAN are presented in Section 3. The proposed sectional analysis approach is presented in Section 4. Early prediction results with interference heatmap are discussed in Section 5. Experimental studies with quantitative and qualitative results are shown in Section 6, followed by Section 7 for conclusions.

2. Network Architecture

GANs are well-known for its capability of generating artificial images. GANs are composed of two deep neural

networks [6], i.e., a generator and a discriminator. Two of them compete and learn from each other, and in the end the generator can generate a fake image which can not be distinguished from the true image by the discriminator, hopefully also by the human eyes. Some interesting applications using GANs include: colorizing cartoon characters from their sketch images [15, 22], creating super-resolution images from the lower resolution [13], transforming from one domain to another domain [29], repairing images by filling their missing parts [20], and generating three-dimensional models given two-dimensional images of objects from multiple perspectives [4].

Among various applications, colorization is chosen as the starting point since it is most relevant to our problem. The colorization of a given sketch of floor plan not only needs to preserve its border shape, but also needs to learn from its internal structure to generate the correct signal heatmap since the structure of a floor plan highly correlates with its heatmap. For example, a concrete wall has much higher signal attenuation than a dry wall, and therefore, the radio signal strength decays quickly around a concrete wall. To solve our image translation problem, we adopt the conditional GAN (cGAN) [18], which learns the mapping from an input image and a random noise to the corresponding output image. However, we drop the random noise in our architecture following the convention in cGANs [7, 24, 29, 10], since we only focus on generating one heatmap with optimal radio dot placement.

In order to preserve most of the floor plan structure, we adopt the U-Net [21, 15] architecture as our generator. U-Net is a special encoder-decoder network such that it concatenates each layer in the encoder to the symmetric layer in the decoder. This bypass scheme will minimize the sketch structure loss through the entire feature extraction and reconstruction process. For the discriminator, we select an encoder only network as our architecture. Its goal is to classify between a real heatmap and a fake heatmap from the generator. The PatchGAN [7, 15] architecture is used to output a matrix of probabilities for the final layer in the discriminator.

A unique modification in the DA-cGAN is that it considers physical dimension as additional input. Physical dimension is usually not considered in this kind of problem, i.e., only pixel space is. However, pixel space is not sufficient for modeling radio signal propagation. In a traditional physics-based simulator, a heatmap is derived based on its path loss¹ which is dominated by two factors: 1) the distance between the radio dot and the current location, and 2) the environmental parameter known as path loss exponent [5]. The path loss increases logarithmically as the physical distance increases from a radio dot, and the path loss exponent is a fixed value measured based on different scenarios. To consider the physical distance, we need the resizing ratio from the original floor plan to the resized image which is served as the input to the cGAN, and a scaler (feet/pixel) s measured on the original floor plan. Denote $W, H, w,$ and h the width and height in pixels of the raw floor plan and the resized floor plan, respectively. To inform the cGAN regarding the physical dimensions and the path loss information, we propose a dimension-aware feature z defined as

$$z = \left[\frac{W}{w}, \frac{H}{h}, s, \log s \right]. \quad (1)$$

Another important feature of the DA-cGAN is the macro-cell interference. We propose to incorporate the measured RSRP heatmap I from outdoor macro cell into our model, so that it is guided to generate more radio dots to cover the highly interfered regions. By passing z to a fully connected layer and three deconvolutional layers [25], and passing I through similar encoder structure from U-Net, we can concatenate them with the deeper layers in our U-Net encoder. As shown in Figure 2, the proposed DA-cGAN adds z and I to U-Net, and rest of the architecture remains the same. Figure 3 shows two prediction results from the proposed DA-cGAN using two scales s within z , as an example of showing how physical dimension affects heatmaps when s is intentionally set to ten times larger than its real value. We can see larger s tends to generate small blobs of signal propagation and thus creates more radio dots (e.g. the emerged one inside upper room in Figure 3(b)), which is aligned with physics of radio propagation.

Two advanced layers are also added to the DA-cGAN. The first one is the spectral normalization layer [19], for stabilizing the training of the discriminator by limiting the weights to a certain range. The second one is the attention layer [27] to capture long range dependency in image generation tasks, through self-attention map construction. This can mitigate the drawback of traditional convolutional layer which is limited by using only spatially local points in lower

¹Path loss is defined as the transmitted power from a radio dot minus the received power on a UE.

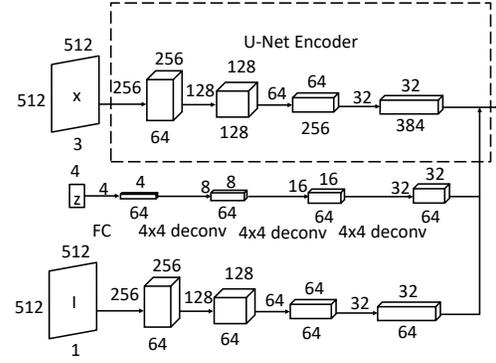
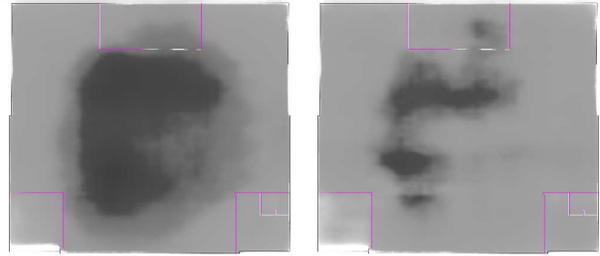


Figure 2. Proposed DA-cGAN with z feature and interference heatmap I . FC is a fully connected layer and deconv is a deconvolutional layer. Only the encoder part is modified in the U-Net.



(a) $s = 0.017$ (1X). (b) $s = 0.17$ (10X).

Figure 3. DA-cGAN prediction comparisons using different s values overlaid with a floor plan.

resolution feature maps. The attention layer is applied to the layer before the last convolutional layer in the discriminator.

3. Loss Functions

The DA-cGAN learns a mapping from a floor plan image x to its output heatmap image y . It is a one-to-one mapping problem, and therefore no random noise vector should be involved. The objective of our DA-cGAN is to optimize the value function V^2 according to the below optimization.

$$\min_G \max_D V(G, D) = \mathbb{E}_{x, y \sim p_{data}(x, y)} \{ \log D(x, y) \} + \mathbb{E}_{x \sim p_{data}(x)} \{ \log(1 - D(x, G(x))) \}, \quad (2)$$

where G is the fake image generated from the generator, D is the discriminator output probability to classify between real and fake input pair, $p_{data}(x, y)$ is the joint distribution for images x and y , $p_{data}(x)$ is the distribution of input image x , and $\mathbb{E}\{\cdot\}$ is the expectation operation.

To solve the min-max optimization problem, we can break it into two sub-problems and solve them iteratively.

²We assume the proposed feature vector z and interference heatmap I are embedded in the input image x to make the notation simple.

The loss function to optimize D is

$$L_D = -\mathbb{E}_{x,y \sim p_{data}(x,y)} \{\log D(x,y)\} - \mathbb{E}_{x \sim p_{data}(x)} \{\log(1 - D(x, G(x)))\}, \quad (3)$$

and the loss function to optimize G is

$$L_G = -\mathbb{E}_{x \sim p_{data}(x)} \{\log(D(x, G(x)))\}. \quad (4)$$

3.1. Baseline Loss Functions

Previous cGANs [15, 20] have shown that by mixing traditional loss functions with the cGAN objective, cGAN can be customized to learn the desired property. We apply three additional loss functions to our objective function for the generator, and merge them as the baseline loss functions. The first one is L_1 loss. It can maximize pixel level similarity between the generated image $G(x)$ and the ground truth image y [15]. It is represented as

$$L_1 = \mathbb{E}_{x,y \sim p_{data}(x,y)} \{\|y - G(x)\|_1\}, \quad (5)$$

where $\|\cdot\|_1$ is the ℓ_1 norm. The second one is the total variation loss L_{TV} [8] which encourages spatial smoothness for $G(x)$. In addition to pixel values, pixel distribution is also important in our problem. To enhance the visual quality of the generated image, we adopt the multi-scale structural similarity index (MS-SSIM) as the third baseline loss function [28]. MS-SSIM has been well-studied to compare two images by accounting the fact that human visual system is sensitive to changes in local structure, and it addresses the issue with variations of viewing conditions by multi-scaling. The loss function is represented as

$$L_{MS-SSIM} = 1 - \text{MS-SSIM}(x, G(x)), \quad (6)$$

where MS-SSIM index is defined in [26].

3.2. Gradient Similarity Loss

By inspecting our ground truth heatmap image, we notice a unique characteristic of it, i.e., the gradient pattern. The uniqueness of this feature is that it propagates the signal strength from the center of the radio dot, to its surrounding pixels smoothly. Since DA-cGAN should learn optimal design and their respective signal propagation pattern, it is important to reward a candidate by similarity of general propagation pattern rather than pixel-wise match while ignoring shape. To compare the gradient pattern between two images, we first propose a new metric, i.e., the gradient similarity (GSIM) index, and then extend it to its multi-scale version. GSIM can be derived in three steps. Consider a pixel location (i, j) for image y and $G(x)$. First we apply a two-dimensional Sobel filter [9] to obtain the gradient on (i, j) , and they are represented as $\nabla y_{i,j}$ and $\nabla G(x)_{i,j}$, respectively. Then we calculate their cosine similarity (CS),

denoted as $\text{CS}(\nabla y_{i,j}, \nabla G(x)_{i,j})$. Finally, we average out CS on all pixels to obtain GSIM, which is represented as

$$\text{GSIM} = \mathbb{E}_{x,y \sim p_{data}(x,y)} \{\text{CS}(\nabla y, \nabla G(x))\}, \quad (7)$$

where ∇y and $\nabla G(x)$ are two-dimensional matrices with $\nabla y_{i,j}$ and $\nabla G(x)_{i,j}$ as their elements, respectively.

Through experiments, we found GSIM is sensitive to image scale. On some large-scale images, although visually we can see a good gradient pattern from a predicted image $G(x)$, GSIM provides low score. To address this scaling issue, we propose a multi-scale version of GSIM, i.e., MS-GSIM index. The key idea is that we apply GSIM on different image scales, and then merge them with a meaningful average function. First, we define an average pooling function $avg_pool(y, k, \xi, \rho)$ for scaling purpose, given its input image y , kernel size k , stride ξ , and padding method ρ . The output of different scales of images can be represented recursively as $a_{i+1} = avg_pool(a_i, k, \xi, \rho)$ and $b_{i+1} = avg_pool(b_i, k, \xi, \rho)$ with $a_1 = y$ and $b_1 = G(x)$, $\forall 1 \leq i \leq N$, where N is number of scale. Then we define an average function f to combine the GSIM score from all scales. The proposed MS-GSIM index is calculated with $f(\text{GSIM}(a_1, b_1), \dots, \text{GSIM}(a_N, b_N))$. Note we adopt $k = 2$, $\xi = 2$, ρ as zero padding, and $N = 5$ through our experiments.

The most intuitive function for f is the arithmetic mean. But after some experiments, we found using harmonic mean provides better result as it shifts the mean toward the lower GSIM loss. In this case we will have smaller loss value if some of the scales result in good GSIM loss. The proposed MS-GSIM loss function is represented as

$$L_{MS-GSIM} = f((1 - \text{GSIM}(a_1, b_1)), \dots, (1 - \text{GSIM}(a_N, b_N))), \quad (8)$$

where $f(x_1, \dots, x_N) = \frac{N}{\sum_{i=1}^N \frac{1}{x_i}}$.

3.3. Focal L_1 Loss

Since our final goal is to place radio dots onto the generated heatmap, the pixels with stronger signal are more important, although all the above loss functions treat all pixels equally. Inspired by the focal loss originally being used to lower the miss-classification probability for object detection [14], we propose a new focal shifted L_1 loss by adjusting the weight of a pixel proportional to its signal strength.

In its implementation, weight change should be considered in different signal strength distributions among heatmaps by shifting the signal range. The proposed focal shifted weight $w_{i,j}$ for each pixel (i, j) is represented as

$$w_{i,j} = \begin{cases} -\log \epsilon, & \text{if } q \geq y_{i,j}, \\ -\left(\frac{1-y_{i,j}}{1-q}\right)^\gamma \times \log\left(\frac{y_{i,j}-q}{1-q} + \epsilon\right), & \text{if } q < y_{i,j}, \end{cases} \quad (9)$$

where ϵ is a very small number used to stabilize numerical computation for log, q is a shifting value that is determined by 1.5% quantile of heatmap, and γ is a hyperparameter used to control the slope for weight decay. We assume $y_{i,j} \in [-1, 1]$ where -1 is the strongest signal and 1 is the weakest signal on the heatmap. The focal shifted L_1 loss is represented as

$$L_{focal} = \mathbb{E}_{x,y \sim p_{data}(x,y)} \{ \mathbf{W} \circ \|y - G(x)\|_1 \}, \quad (10)$$

where \mathbf{W} is a two-dimensional matrix with the elements $w_{i,j}$ defined in equation (9) and \circ is the element-wise multiplication operator.

3.4. Multi-Stage Training Strategy

Since MS-GSIM is good at preserving the global propagation pattern while MS-SSIM is good at preserving local pixel layout, we need both to benefit our prediction. A straightforward method is to add them together. However, through experiments we found that MS-SSIM and MS-GSIM loss conflict with each other, during the early stage of training. Therefore, we propose a multi-stage training strategy to exploit the advantages for both losses. The idea for this strategy is that: we execute the same training procedure as any other GANs, except the loss function for the generator is replaced. The loss function contains two stages. We use L_1 and $L_{MS-GSIM}$ for the first stage because finding propagation pattern and then pixel-wise matching is easier than its vice versa. Once the propagation pattern is trained to be good enough, i.e., validation performance ζ reaches a certain level θ , we switch to the second stage and use L_{focal} and $L_{MS-SSIM}$ to focus more on generating pixel-wise correct heatmap and thereby estimating more precise dot locations. L_G and L_{TV} losses are general baseline loss functions which are applied on both stages. The multi-stage loss function L_{Total} for the generator is represented as

$$L_{Total} = \begin{cases} L_G + \lambda_t L_{TV} + \lambda_l L_1 + \lambda_g L_{MS-GSIM}, & \text{if } \zeta \leq \theta, \\ L_G + \lambda_t L_{TV} + \lambda_f L_{focal} + \lambda_s L_{MS-SSIM}, & \text{if } \zeta > \theta, \end{cases} \quad (11)$$

where λ_t , λ_l , λ_g , λ_f , and λ_s are weights to control the effect of each corresponding loss function. Figure 4 shows the two contrastive two-stage training strategies, demonstrating the importance of the sequence. The solid line is from our proposed order while the dashed line is from its inverse order. It shows that the inverse order starting from MS-SSIM causes a significant performance drop when switching to the second stage, since MS-SSIM is a more detailed loss function in terms of image structure, and it can be stuck at a local optimum where MS-GSIM is hard to find a better fit nearby.

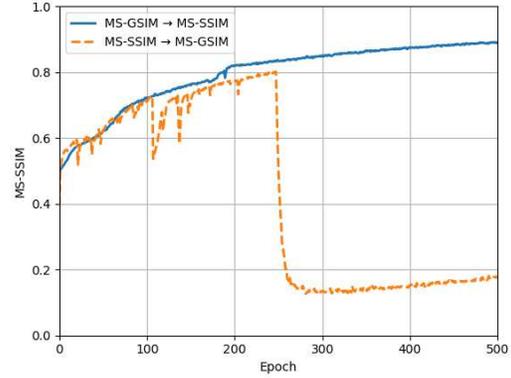


Figure 4. Two training strategies evaluated with MS-SSIM score using $\theta = 0.8$.

4. Sectional Analysis

Since each floor plan has different physical dimension (height, width in feet and area in square feet) as well as pixel dimension, we need a method to resize all of them to a single pixel dimension for training the DA-cGAN with limited VRAM of graphics processing units (GPU) that we use. One possible approach is to resize all images to as high pixel dimension as possible, but this requires very large memory size for training. Authors in [24, 3] tackle image translation tasks on high-definition images by enlarging their generator and image size, which is not practical in our case as our floor plans can be as large as over 10000×10000 pixels. If we resize such floor plans to an acceptable dimension such as 512×512 , the resizing process will lose thin lines and therefore we suffer from the broken wall structure resulting in wrong prediction. Therefore, we propose to divide the floor plan that has large physical dimension into small sections, predict heatmap for each section, and merge them to a single large heatmap.

We define a wide floor plan and a regular floor plan. A wide floor plan is the one with its height greater than 300 feet or width greater than 400 feet, and the other floor plans are defined as the regular floor plan. We only include regular floor plans in our training set, and we place wide floor plans in the validation set which are handled by the sectional analysis scheme. The sectional analysis scheme is described as follows. First we define a golden width w_g and height h_g by calculating the average width and height from all floor plans in the training set, in physical dimension. Second, we pad the floor plan to width W_{pad} and height H_{pad} such that $\frac{W_{pad}}{w_g}$ and $\frac{H_{pad}}{h_g}$ become integers. Third, we divide the whole padded floor plan into $\frac{W_{pad}}{w_g} \times \frac{H_{pad}}{h_g}$ sliced images. Then we resize all the sliced images to a fixed input dimension $w \times h$ for the DA-cGAN and predict the heatmap for each sliced image. Finally we reconstruct the

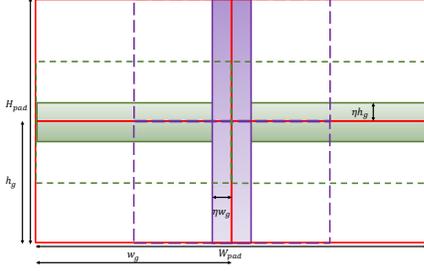


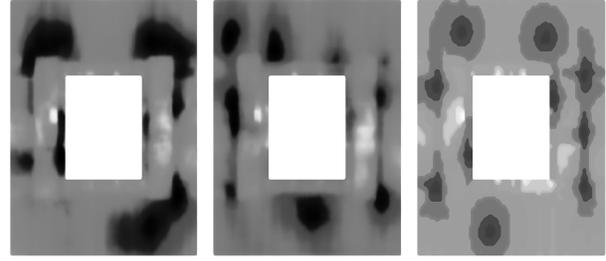
Figure 5. An example of floor plan with sectional analysis. Main sliced floor plans are composed of red lines, sliced floor plans with green lines are for horizontal border recovery, and sliced floor plans with purple lines are for vertical border recovery. The green and purple shaded areas are the actual overlapping region.

whole heatmap by stitching the sliced heatmaps together.

However, there is one drawback for this approach. On the borders for each sliced floor plan, it is possible that there exists a wall structure being cut off. Our DA-cGAN may not be able to capture this edge scenario and hence it may miss radio dots on the borders. To solve this issue, we propose to apply additional two sets of sliced floor plans for prediction. The first set is to place the sliced floor plans on all the horizontal borders, but with a $\frac{h_g}{2}$ shift from the original sliced floor plans. Similarly, the second set is to place the sliced floor plans on all the vertical borders, but with a $\frac{w_g}{2}$ shift from the original sliced floor plans. We ignore the outmost 4 borders. After predicting on both sets, we only stitch a ratio η of h_g from horizontal borders with the prediction from the first set, and η of w_g from vertical borders with the prediction from the second set. Since the prediction results from two sets observe similar floor plan structure as the original sliced floor plan, we expect them to have very similar prediction results, except the region closed to the border. Figure 5 illustrates the concept of main sliced floor plans and the horizontal/vertical sliced floor plans.

5. Macro-Cell Interference

We present one preliminary result using I feature in our DA-cGAN, and it is shown in Figure 6. This is a floor plan that is heavily interfered by its surrounding macro-cell on the border of inner white square area in the same RF channel as the radio dots. The reason for the inner area to be affected may be due to the height for macro base station is taller than this floor, and its angle of antenna is tilted toward the inner area. In a heavily interfered area, the RF planner has the option to choose to either avoid placing radio dots to increase the spectrum efficiency, or place more radio dots if it must have radio coverage. We can see a clear difference in the prediction pattern from the figure, since our DA-cGAN can learn from the behavior of RF designer if I feature is adopted. However, only 15% of our floor plans are affected



(a) Prediction with interference heatmap. (b) Prediction without interference heatmap. (c) Ground truth heatmap.

Figure 6. DA-cGAN prediction comparisons with and without interference on a single floor plan.

by macro-cell interference. This kind of imbalanced data is hard for our DA-cGAN to capture well regarding the interference pattern. Thus this part of research is still under development.

6. Experiments

6.1. Dataset

In order to train DA-cGAN, we need raw floor plans and radio signal heatmap corresponding to the floor plans, and the heatmap is designed by experienced radio engineers. These images are obtained through our internal customers from hundreds of buildings in the U.S., locating largely in California and Texas. In total we get 1256 raw floor plans and heatmaps, and we divide them into 1057 pairs for training set, 154 pairs for model validation set, and 45 pairs for overall validation set. All images in our dataset are regular floor plans, except 16 out of 45 images in the overall validation set are wide floor plans.

Before training DA-cGAN, several image preprocessing schemes are applied to our image pairs. First we transform the heatmap from an RGB image to a grayscale image. Since different RSRP values map to different colors, we build a lookup table to achieve this conversion. Then we crop the raw floor plans by removing unnecessary sections to match up to the region of interest shown in its heatmap. Finally we normalize pixel range to $[-1, 1]$ and resize all images to 512×512 to fit our DA-cGAN.

Despite cropping the floor plans, DA-cGAN is still hard to distinguish between desired and undesired area. The outside of the floor plan (undesired area) and the inside of the floor plan (desired area) are with the same white color. DA-cGAN tends to generate white color in the desired area rather than produce signal heatmap, since it can not interpret the meaning of the white color. To solve this issue, we propose to fill the undesired area with a dummy color.

6.2. Method

For training and validation set, the harmonic mean of MS-SSIM and MS-GSIM is used as a similarity score ζ of heatmaps to choose the best model hyperparameters. Testing the final model requires another metric of measuring dot placement similarity with human design, because of possible discrepancy between the similarity of heatmaps and dot placements. For measuring the dot placement similarity, heatmaps should be converted to blobs with their centers, to which radio dots should be placed. For the blob analysis, we run a blob counting algorithm, i.e., applying image thresholding and hierarchical density-based clustering (HDBSCAN) [17] on $G(x)$ and y to get the predicted and true dot locations, respectively. From the given pair of blobs, two proposed metrics, i.e., dot ratio and dot deviation (dev.), are calculated to evaluate our framework. Dot ratio is simply the ratio between the predicted number of dots from $G(x)$ and the true number of dots from ground truth y . Ideal dot ratio is 1.0 and dot deviation is 0. Our internal customers set up the performance criteria for the practical deployment of this framework, i.e., median dot ratio should be 1.0, and median dot deviation should be smaller than 10 feet to be a feasible design, less than 5 feet for a desirable design.

For training our DA-cGAN, Adam optimizer [11] is used with $\beta_1 = 0.5$, $\beta_2 = 0.999$, and learning rate 0.0002, while spectral normalization and batch normalization with batch size 20 are applied to all convolutional layers. Also, for reducing generalization gap, stochastic image augmentation is applied to the training set by horizontally flipping with 50% chance and rotating by one of 0, 90, 180, or 270 degree with 12.5% probability. We select the best hyperparameters through our validation sets. In our hyperparameter optimization, the best configuration for loss functions selected from model validation set is $\lambda_t = 0.001$, $\lambda_1 = 100$, $\lambda_g = 40$, $\lambda_f = 100$, $\lambda_s = 84$, $\epsilon = 10^{-14}$, $\gamma = 2$, and $\theta = 0.8$. The best hyperparameters for sectional analysis are $w_g = 271$ (ft), $h_g = 189$ (ft), and $\eta = 0.1$ from overall validation set. All our implementations are based on TensorFlow version 1.13.1 [1] with multiple NVIDIA RTX 2080 Ti GPUs.

6.3. Results

We find the similarity scores shown in our model validation set are mostly similar across all models but their propagation patterns are visually quite different. Here we only report location estimation performance on various models in regular floor plans in the following based on the interests of our customers. The location estimation performance is derived by taking the median of all dot ratios and dot deviations from the overall validation set. Table 1 shows the numerical results by comparing different models with and without using z , i.e., physical dimensions. We only com-

pare dot deviations in this table since all dot ratios are 1.0. Note that we include L_D , L_G , and L_{TV} in every model. We can see that using z can reduce the dot deviation significantly across different models. $L_1 + L_{MS-SSIM}$ results in the lowest deviation among single-stage approaches, and this is due to its structural optimization ability, which is superior than the pixel-wise (L_1) or the gradient-wise ($L_{MS-GSIM}$) models. Without z , $L_1 + L_{MS-GSIM}$ significantly contributes to reduce the dot offset compared with the typical L_1 . We also observe another interesting point showing that $L_1 + L_{MS-GSIM}$ and $L_1 + L_{MS-MSIM}$ perform similarly with z , but pretty differently without z . This is due to $L_{MS-GSIM}$ and z are somehow sharing the global propagation pattern. However, with the two-stage approach, we can further reduce the dot deviation from best single-stage model thanks to the power of selecting precise dot locations from L_{focal} . The distribution of true dot location is a function of gradient pattern, but L_{focal} has no such direction by its definition. Optimizing L_{focal} itself simply brings the prediction toward the non-conditional dot location distribution, and that is why we need two-stage approach to make L_{focal} search in the conditional space. Lastly, we study the two-stage model by removing the discriminator. Clearly without D , we can see the dot deviation increases by 9.76% (with z) or 22.12% (without z). Interestingly, the median absolute deviation of dot deviation increases by around 50% if it is not trained with D , regardless of z . The result implies that cGAN is a better choice for training a robust model, which is less affected by inter-sample variance, than the G only model at least in the context of pixel-to-pixel translation.

Figure 7 shows three qualitative results with the overall validation set. The first two rows are regular floor plans. The last row is an example for a wide floor plan. The first column is the input floor plan image, the second column is the ground truth heatmap from human designers, and the last column is the predicted heatmap from DA-cGAN. We observe from the floor plan structure, if we have a simple and clear wall type and more open spaces, DA-cGAN tends to predict better. However, if we have more complicated, blurred, or broken floor plan with unseen or unusual tags, the predicted heatmap will diverge from the ground truth. For a very large floor plan, although we segment them into a reasonable floor plan size and make prediction, we can see it tends to predict more dots in the end. This is due to a wide floor plan does not necessarily have large number of dots. We believe the discrepancy of layouts between a human designer and our model does not necessarily mean that the model predicted performance is inferior to the ground truth, because our model has the potential of exploring the search space where human designers have not found.

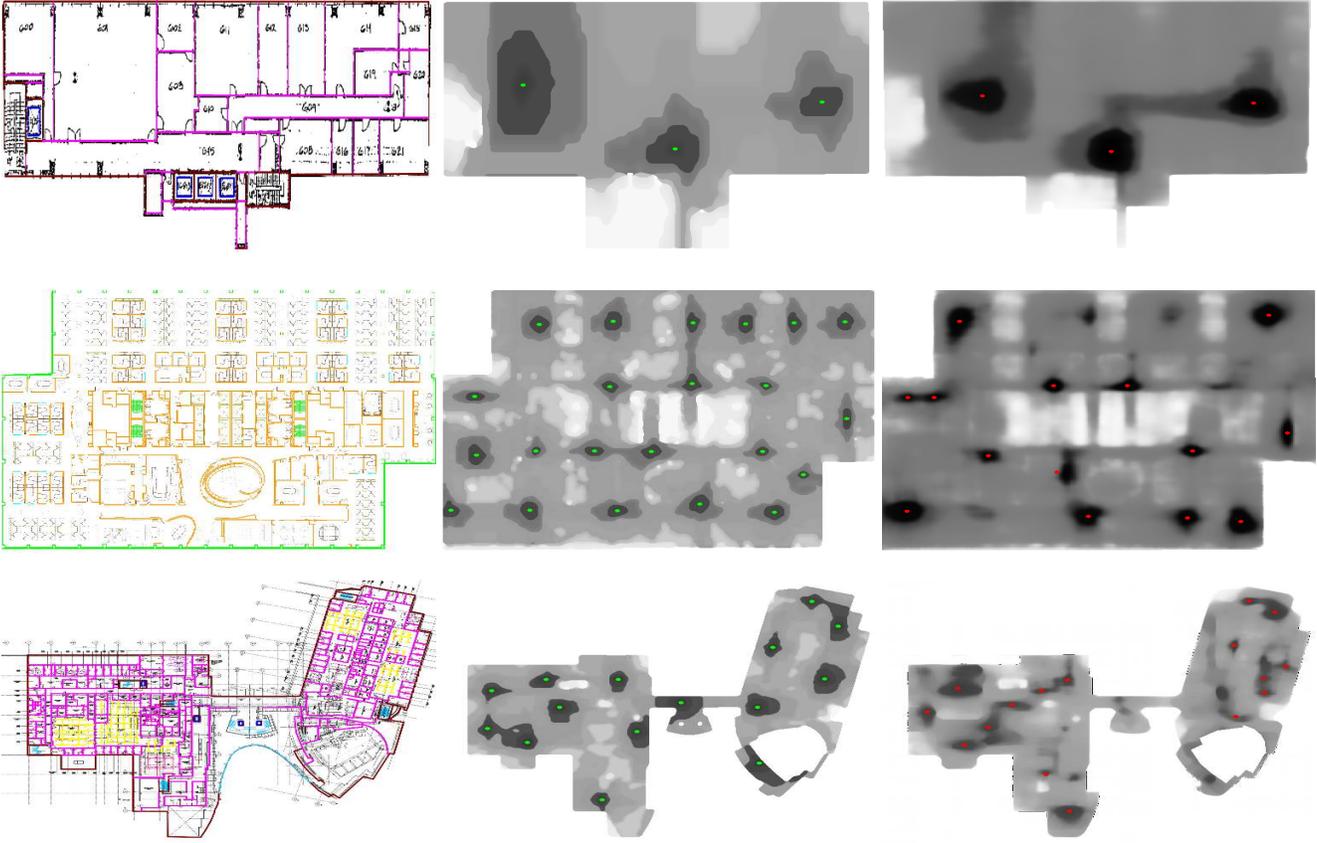


Figure 7. Examples of predictions. The first column is for floor plans, the second one is for their ground truth dot layout and heatmaps, and the third one is for the predicted dot layout and heatmaps. Radio dot locations are marked as green circles in the ground truth heatmap and red circles in the predicted heatmap. The first and second rows are the examples of good and neutral prediction on a regular floor plan, respectively. The third row is an example of wide floor plan prediction with sectional analysis. The area size for each row is {8440, 63056, 85998} square feet. The dot deviation and dot ratio for each row are {(4.68, 1.00), (13.01, 0.64), (9.60, 1.07)}.

Table 1. z performance comparisons for DA-cGANs on regular floor plan in overall validation set without I feature. Median value is taken from all samples. \pm stands for one median absolute deviation.

Model	Dev. w/ z	Dev. w/o z
L_1	10.37 ± 5.02	13.27 ± 6.28
$L_1 + L_{MS-GSIM}$	10.27 ± 4.73	10.80 ± 5.53
$L_1 + L_{MS-SSIM}$	10.07 ± 5.41	11.75 ± 6.20
Two-stage L_{focal} w/o D	10.45 ± 7.24	11.98 ± 7.40
Two-stage L_{focal} w/ D	9.52 ± 4.74	9.81 ± 4.92

7. Conclusions

A novel approach of generating optimal indoor radio design from a given floor plan is proposed by formulating it to an image-to-image translation problem and applying a customized cGAN to be aware of dimension for the purpose of simulating physical radio propagation. The proposed DA-cGAN can generate the desirable heatmap for a floor plan

and place radio dots accordingly by learning patterns from accumulated human design while considering its macro-cell interference. Several novel loss functions and a two-stage training strategy are proposed to guide the DA-cGAN to learn the radio propagation pattern as well as the precise radio dot locations, in addition to a sectional analysis, i.e., a divide-and-conquer method for handling large floor plans with limited VRAM of GPUs. The trained model achieved the goal of less than 10 feet in median dot deviation and 1.0 in median dot ratio. The proposed solution is expected to reduce lead time and total cost of indoor radio design process significantly.

References

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga,

- Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. [7](#)
- [2] Rocío Acedo-Hernández, Matías Toril, Salvador Luna-Ramírez, Carlos Úbeda, and María Josefa Vera. Automatic clustering algorithms for indoor site selection in lte. *EURASIP Journal on Wireless Communications and Networking*, 2016(1):87, Mar 2016. [1](#)
- [3] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. [5](#)
- [4] M. Gadelha, S. Maji, and R. Wang. 3d shape induction from 2d views of multiple objects. In *2017 International Conference on 3D Vision (3DV)*, pages 402–411, Oct 2017. [2](#)
- [5] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, 2005. [3](#)
- [6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. [2](#)
- [7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016. [2](#)
- [8] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016. [4](#)
- [9] Nick Kanopoulos, Nagesh Vasanthavada, and Robert L. Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, 23(2):358–367, 1988. [4](#)
- [10] Taeksoo Kim, Moonsoo Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML’17, page 1857–1865. JMLR.org, 2017. [2](#)
- [11] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. [7](#)
- [12] Neil E. Klepeis, William C. Nelson, Wayne R. Ott, John P. Robinson, Andy M. Tsang, Paul. Switzer, Joseph V. Behar, Stephen C. Hern, and William H. Engelmann. The national human activity pattern survey (nhaps): A resource for assessing exposure to environmental pollutants. *Journal of Exposure Analysis and Environmental Epidemiology*, 11(3):231–252, Mar 2001. [1](#)
- [13] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 105–114, July 2017. [2](#)
- [14] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018. [4](#)
- [15] Yifan Liu, Zengchang Qin, Zhenbo Luo, and Hua Wang. Auto-painter: Cartoon image generation from sketch by using conditional generative adversarial networks. *ArXiv*, abs/1705.01908, 2017. [2, 4](#)
- [16] Chenguang Lu, Miguel Berg, Elmar Trojer, Per-Erik Eriksson, Kim Laraki, Olle V. Tridblad, and Henrik Almeida. Connecting the dots: small cells shape up for high-performance indoor radio. In *Ericsson Review*. Ulf Ewaldsson, 2014. [1](#)
- [17] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11):205, 2017. [7](#)
- [18] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. [2](#)
- [19] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018. [3](#)
- [20] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2544, June 2016. [2, 4](#)
- [21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. [2](#)
- [22] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. *Computer Vision and Pattern Recognition, CVPR*, 2017. [2](#)
- [23] F. Tonini, M. Fiorani, M. Furdek, C. Raffaelli, L. Wosinska, and P. Monti. Radio and transport planning of centralized radio architectures in 5g indoor scenarios. *IEEE Journal on Selected Areas in Communications*, 35(8):1837–1848, Aug 2017. [2](#)
- [24] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. [2, 5](#)
- [25] Xiaolong Wang and Abhinav Gupta. Generative image modeling using style and structure adversarial networks. In *ECCV*, 2016. [3](#)
- [26] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thirty-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, volume 2, pages 1398–1402 Vol.2, Nov 2003. [4](#)
- [27] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97. PMLR, 09–15 Jun 2019. [3](#)
- [28] H. Zhao, O. Gallo, I. Frosio, and J. Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, 3(1):47–57, March 2017. [4](#)

- [29] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017. 2