Cellular Network Optimization by Deep Reinforcement Learning and AI-Enhanced Ray Tracing

Zhangyu Wang*
zhangyuwang@ucsb.edu
University of California Santa Barbara
Santa Barbara, California, USA

Serkan Isci
sisci@research.att.com
AT&T Labs - Research
Bedminster, New Jersey, USA

Yaron Kanza
kanza@research.att.com
AT&T Chief Data Office
Bedminster, New Jersey, USA

Velin Kounev
velin@research.att.com
AT&T Labs - Research
Bedminster, New Jersey, USA

Yusef Shaqalle
ys974r@att.com
AT&T Labs - Research
Bedminster, New Jersey, USA

ABSTRACT

In this paper we study the use of deep reinforcement learning that is supported by a ray tracer, on top of a detailed 3D model of the geospatial environment, for optimization of antenna tilts in cellular networks. We propose two novel mechanisms—geospatial importance sampling and multi-path coefficient—to efficiently pass geospatial information to the reinforcement learning model. We show that this approach can be used for fast and scalable optimization of tilt levels of cellular antennas. We present an experimental evaluation that compares the use of reinforcement learning to greedy search, simulated annealing and Bayesian optimization. Our study shows that reinforcement learning is effective and can cope with optimization problems that are at a greater scale than the settings the other algorithms can cope with.

CCS CONCEPTS

• **Information systems** → **Spatial-temporal systems**; *Mobile information processing systems.*

KEYWORDS

Cellular networks, network optimization, AI, 3D modeling, reinforcement learning, 4G, 5G, radio propagation, ray tracing

ACM Reference Format:
Zhangyu Wang, Serkan Isci, Yaron Kanza, Velin Kounev, and Yusef Shaqalle. 2023. Cellular Network Optimization by Deep Reinforcement Learning and AI-Enhanced Ray Tracing. In *2nd ACM SIGSPATIAL International Workshop on Spatial Big Data and AI for Industrial Applications (GeoIndustry '23), November 13, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3615888.3627814

1 INTRODUCTION

In the last decades, the demand for bandwidth in cellular networks has been rising rapidly. The number of cellular devices that are connected to the network has been growing, and applications like streaming media and online social networks have led to a large increase in the traffic volume [17, 22, 23, 27]. To cope with the growing demand for network services, new generations of cellular networks, like 4G and 5G, utilize high frequencies and arrays of antennas (MIMO) [14]. This makes cellular transmissions susceptible to the effect of the geospatial environment. Obstacles like buildings, trees, and the terrain obstruct, reflect, and refract the electromagnetic transmissions of antennas, and the effect on the environment grows as the frequencies are increased. Hence, the design and optimization of new cellular networks become difficult but critical, especially in urban areas [15].

When planning a cellular network, the location of each antenna and its height are selected based on the availability of network towers in the area (and regulatory restrictions[1]). The goal is to provide effective cellular coverage. If a device is located too far from any antenna or when geospatial obstacles obstruct the transmission, the signal strength will be low. However, when a device receives radio-frequency signals from several antennas, this may cause an interference—the signals from some antennas interfere with the signals of other antennas. To optimize the network and increase the coverage while decreasing the interference, cellular operators change the tilt of antennas and their transmission power. Fig. 1 and Fig. 2 illustrate a cellular antenna and its tilt.

Optimization of the tilt and power of antennas should be done in real-time to address evolving situations, like changes in the demand for network coverage, and to cope with cell outage [3]. A cell outage compensation is a self-healing function often employed by Self-Organising Networks (SON) [24]. The goal is to mitigate the degradation of coverage, capacity, and service quality in the case of an outage of a cell or a site. The cell outage compensation is executed by tuning the electrical tilt of antennas and changing the downlink received-power level, in the cells surrounding the affected site. This requires real-time automation and fast reaction to the outage, while considering the features of the cellular network and the geospatial environment. Computation should also be fast when optimizing large-scale areas [6]. If planning at the scale of a large city or even a country is too slow, some areas would seldom be optimized, and this could affect the quality of service and the availability of the network.

*When this work was conducted, Zhangyu Wang was an intern at AT&T Labs.

[1] https://www.fcc.gov/wireless-telecommunications

GeoIndustry '23, November 13, 2023, Hamburg, Germany

ACM ISBN 979-8-4007-0350-8/23/11.
https://doi.org/10.1145/3615888.3627814

Figure 1: A cellular antenna with sectors that can be tilted.

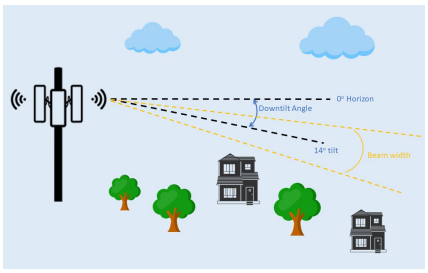


Figure 2: Downtilt of a cellular antenna.

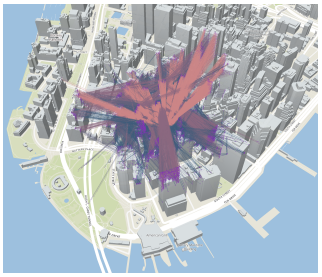


Figure 3: Ray tracing in Rhea: rays are emitted from the antenna for creating a heatmap of the signal strength.

For an effective computation of the coverage of cellular antennas, an accurate propagation model of the radio transmission should be combined with a detailed 3D model of the geospatial environment. Such a computation is expensive, so it could be ineffective in an optimization where a large number of states should be explored—if there are n sector antennas in an area and each antenna has t tilt levels, the search space has a size of t^n. For example, with 20 antennas and 10 tilt levels, the search space will have more than 10 trillion combinations. So, even if only a small portion of the search space is explored in the optimization, each step should be very fast, which is challenging over a detailed geospatial model.

In this paper, we introduce a novel approach in which reinforcement learning is combined with ray tracing to address this optimization problem. We use Rhea, a ray tracing tool developed at AT&T, to compute the propagation modeling of the transmission of each antenna [10]. The ray tracer produces training examples for different antenna parameters and tilt levels. It compresses the geospatial context of signal propagation paths by applying two novel mechanisms that we introduce—*geospatial significance sampling* and *multi-path coefficients*. A reinforcement learning model is trained over the propagation patterns and geospatial contexts created by Rhea. The model is used for efficiently optimizing antenna tilt levels for coverage and interference avoidance. Our experiments show that geospatial significance sampling and multi-path coefficients are effective for optimizing the performance of the reinforcement method. They support industry-level tilt optimization and the ability to cope with nation-wide scale.

This paper is organized as follows. In Section 2 we discuss related work. In Section 3 we formally define the research problem. We present our method and baseline algorithms in Section 4. Our experimental evaluation with a comparison between the algorithms is presented in Section 5. In Section 6 we discuss our conclusions.

2 RELATED WORK

The use of reinforcement learning for optimizing cellular networks has been studied extensively in the last two decades [20, 26, 32]. Vannella et al. [28, 29] showed how to use Safe Policy Improvement through Baseline Bootstrapping. The use of reinforcement learning for optimizing 5G networks has been considered in [31]. Tilt optimization for LTE networks has been investigated in [12] and [11]. These papers, however, have not used a ray tracer and they do not show how to apply reinforcement learning to a detailed model of the geospatial environment. Our work is the first to incorporate a nationwide, industry-level geospatial model in optimization of cellular networks based on reinforcement learning.

Several studies explored the creation and usage of a detailed geospatial model for network planning [5–7] and network optimization [2]. These studies, however, do not consider the use of reinforcement learning for tilt optimization, as in this paper.

The novelty of our approach is the carefully-designed combination of a ray tracer over a detailed model of the environment with reinforcement-learning-based tilt optimization. Geospatial information of the environment is aggregated and compressed by the ray tracer and provided to the reinforcement learning model. We show how to execute the optimization efficiently by utilizing this information. This combination enables an industry-level near real-time optimization of cellular networks.

3 FRAMEWORK

In this section we present our framework, describe the ray tracing tool Rhea, and define the research problem.

Geospatial model. The environment is represented by a geospatial model. The geospatial model is constructed based on a variety of data sources, including USGS[2], NLCD[3], OSM[4], and proprietary geospatial datasets containing objects with shape and height. The geospatial model consists of three types of entities—buildings, foliage, and ground (the terrain). Buildings and trees are modeled by 3D cuboids. Each object is a mesh that is represented by vectors of vertices, triangles that constitute the mesh, and the normal of each surface. Such a model can be generated from a LiDAR point cloud [8]. The terrain is represented by a mesh of triangles and it is produced based on the LiDAR point cloud.

The model consists for each cellular antenna all the parameters that affect the transmission, including power, bandwidth, frequency, antenna pattern, azimuth, and tilt. Azimuth is the angle in the horizontal plane between the center of the main lobe of the antenna transmission pattern (presented in Fig. 4) and the geographical north. Tilt is the angle in the vertical plane between the direction of the center of the main lobe of the antenna and the horizon.

[2]https://www.usgs.gov/the-national-map-data-delivery/gis-data-download
[3]https://www.usgs.gov/centers/eros/science/national-land-cover-database
[4]https://www.openstreetmap.org/

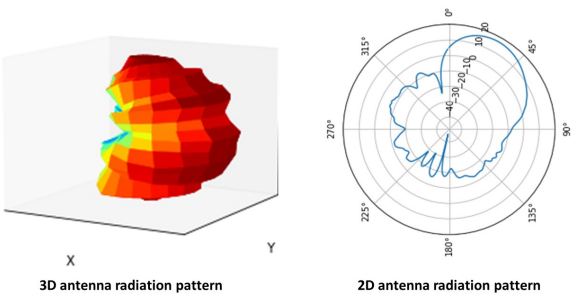


Figure 4: Antenna radiation pattern that maps transmission power to each direction. On the left, a 3D representation of the pattern. On the right, a 2D horizontal slice of the pattern.

The geospatial data is indexed by an octree [21]. The index supports the retrieval of geospatial objects per location with logarithmic time complexity in the number of objects. Penetrable objects (e.g., foliage) and reflective objects (e.g., buildings, ground) are stored in different indexes. It allows to treat them differently and compute the distance traveled through the penetrable objects.

Radio propagation model. Electromagnetic radiation travels in space in straight lines when not affected by objects in the environment. The signal is emitted from an antenna with frequency and power that are assigned to the antenna. The received power density decreases as a function of the distance from the transmitter. To compute the received power at a receiver antenna, we use Friis formula [25]. The formula computes the power loss as a function of the distance between the transmitter and a given point. By knowing the transmission power, in milliwatts or decibel-milliwatts, we can apply the path-loss formula and compute the power at the receiver.

Coverage and interference. The ground is partitioned into bins by a grid, where each bin has a size of 4m×4m. The computation of signal strength is conducted for each bin b by combining the transmissions that reach b. *Coverage* is the set of bins in which the received signal strength exceeds a predefined threshold. *Interference* within a bin whose carrier frequency is f is considered to be unacceptable if the received signal power from the strongest cell transmitting at f (the serving cell) is not sufficiently stronger than the net received signal power from all other cells operating at the same frequency f. That is, suppose the strongest signal at carrier frequency f reaching a given bin b is s_1, and there is another signal of strength s_2 reaching bin b from a different antenna operating at the same frequency f, then s_2 is said to cause critical interference if the ratio of s_1 to s_2 is below a threshold.

Computing Coverage and interference. Many applications rely on signal measurements to discover the coverage and the interference. However, for tilt optimization, measured signal strength does not fully consider the geospatial environment and does not always distinguish between different tilt levels. For example, Fig. 5 and Fig. 7 present the spatial distribution of the signal around the antenna in Area 1 and Area 2, respectively. Fig. 6 and Fig. 8 present the distribution of the measurements of the signal strength in Area 1 and Area 2, respectively. While the average and the overall distribution of measured signal strengths are similar, the spatial distributions of the signal are different in the two areas.

Fig. 9 and Fig. 10 present the spatial distribution of the signal for high and low tilt of a given antennas. In Fig. 11 we see in yellow the area in which the signal is stronger for a low tilt level and in purple the area in which the signal is stronger for a high tilt level. Note that a low tilt level mostly covers areas near the antenna and high tilt covers areas that are further away. Despite the large difference between the spatial distribution for the two tilt levels, the distribution of measured signal strengths in the area is similar for the two tilt levels. So, ordinary (non-spatial) distributions of measured signal strengths do not always distinguish between different tilt levels and are not able to measure the effect of the tilt. To address this, we use a ray traces to compute the signal strengths.

Ray Tracing. Our ray tracer, Rhea, was developed in-house. The main goal of Rhea is to accurately and efficiently compute coverage and interference per bin. The results can be presented as a heatmap. The ray tracer is applied to a model of the environment, see Fig.3. It executes a Monte-Carlo photon injection algorithm in the 3D model and the rays geometrically approximate electromagnetic signal propagation. In particular, key components such as antenna patterns, signal energy spreading with distance, reflections, diffractions and refractions on solid surfaces, and penetration through foliage are captured in a deterministic fashion, while accounting for the various components of attenuation and multipath fading.

When applying the ray tracer, various parameters are associated with each antenna, including position, transmit power, carrier frequency, spectrum bandwidth, waveguide loss, and antenna pattern. Antenna manufacturers provide 2D transmission patterns that are converted to 3D patterns by interpolation. The pattern is rotated to align with the antenna azimuth and tilt angles. Thus, any photon injected from the source at a relative azimuth θ and downtilt ϕ in reference to the antenna orientation is ascribed an antenna gain $G_{tx}(\theta, \phi)$. The hit points of photons are counted per each bin. The wavefronts of the photons are combined per rules of phasor addition to account for multipath fading, for determining the net received signal strength in each bin.

The computation has two phases: *(i)* photon traversal, and *(ii)* bin-level consolidation. In the photon traversal phase, photons are emitted from the source in pseudo-random directions, uniformly. The total number of photons in the simulation is based on the processing budget (up to 40 million) and desired bin granularity. The path of each photon is traced in the 3D model using standard geometrical procedures. The system records for each photon the traveled distance, net attenuation, phase shifts, reflections, and travel through foliage, to compute the path loss. While light ray reflections are glossy (combination of specular and diffusive) [30], the radio signal reflections are predominantly specular, given that their wavelengths are considerably longer than surface imperfections.

In the second phase, the net received signal strength is computed for every bin on the ground. For this purpose, the set of distinct wavefronts (marked by different paths) is identified per each bin based on the photons hitting that bin. These wavefronts are combined per rules of phasor addition to account for multipath fading and compute the net received signal strength for each bin.

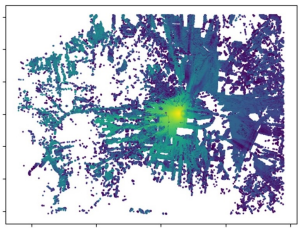


Figure 5: The spatial distribution of the signal in Area 1.

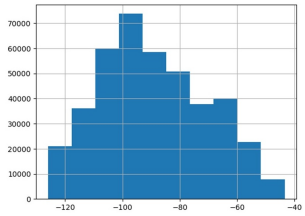


Figure 6: Distribution of the measured signal strength in the area depicted in Fig. 5

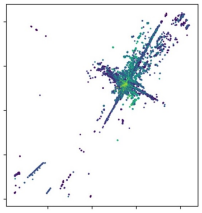


Figure 7: The spatial distribution of the signal in Area 2.

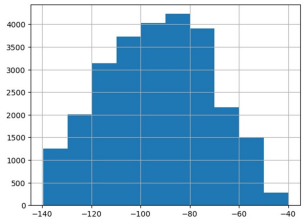


Figure 8: Distribution of the measured signal strength in the area depicted in Fig. 7.

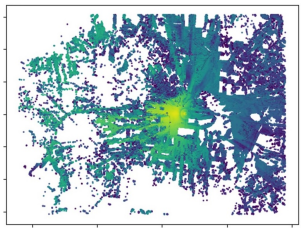


Figure 9: The spatial distribution of the signal for a high tilt.

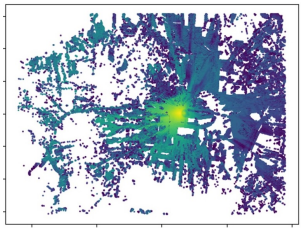


Figure 10: The spatial distribution of the signal for a low tilt.

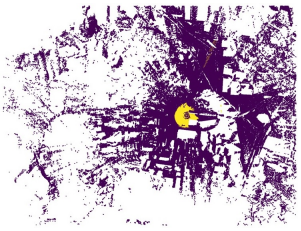


Figure 11: Where the signal is stronger for low tilt (yellow) and for high tilt (purple).

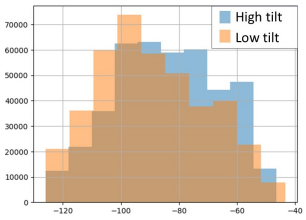


Figure 12: The distribution of the signal strength for the high and low tilt levels.

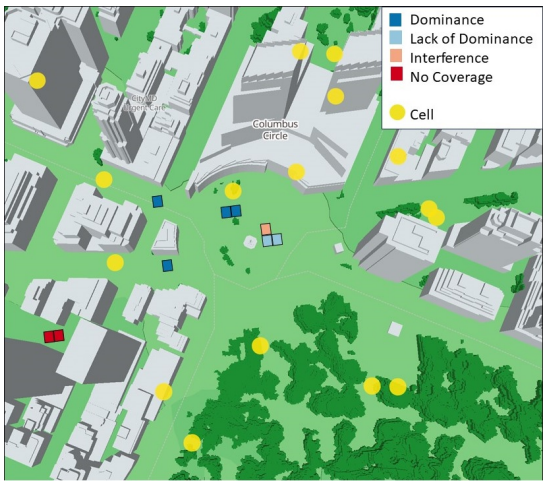


Figure 13: Different states of bins in an area. (The cell locations were selected randomly.)

Tilt Optimization. The goal of network optimization is to increase coverage and decrease interference, in the considered area. Recall that the area is partitioned into bins. Each bin is in one of the following four states, defined based on practical industry standards, for two arbitrary thresholds α and β.

- **Dominance.** The strongest signal in the bin is less than α dBm below the environment noise floor (the sum of all the other signals), and the second and third strongest signals in this bin are more than β dBm below the strongest signal.
- **Lack of Dominance.** The strongest signal is less than α dBm below the environment noise floor, and the second and third strongest signals are less than β dBm below the strongest signal. (Two signals are too close to the strongest.)
- **Interference.** The strongest signal is more than α dBm below the environment noise floor, that is, it is not strong enough in comparison to the sum of all other signals.
- **No Coverage.** There is no signal in the bin (all the signals are too low).

Fig. 13 presents examples of different bin states. The antenna locations are marked by yellow circles. (These locations were selected randomly just for the example.) Bins that are close to a single antenna typically are in a dominance state. Bins between antennas are in a lack of dominance or interference state. Bins that do not have any antenna directed to them or are hidden by obstacles are often in a no-coverage state.

The optimization aims to maximizing the total number of bins with a dominance state and minimizing the total number of bins without dominance of a single serving cell. In the optimization, we cannot move or shut down existing antennas. It is only possible to electrically change the antenna tilts and the transmission strength, to change the coverage. For example, up-tilting an antenna usually makes the covering sector larger and the energy more sparsely distributed, while down-tilting usually makes a smaller, more concentrated covering sector. But because of geospatial obstacles, the actual effect is complicated and we use Rhea to compute it.

Typically, it is impossible to achieve dominance in all the bins. Thus, both dominance and lack-of-dominance are considered acceptable states, while trying to prevent interference and no-coverage states. To evaluate this, the system is maximizing a weighted sum

$S = w_1 s_{dominance} + w_2 s_{no\text{-}dominance} - w_3 s_{interference} - w_4 s_{no\text{-}coverage}$, where w_1, w_2, w_3 and w_4 are positive weights whose sum is equal to 1, and $s_{dominance}$, $s_{no\text{-}dominance}$, $s_{interference}$, $s_{no\text{-}coverage}$ are the number of bins in each state. The weights are selected by network engineers as part of the network quality policy of the cellular provider. The sum S is called *coverage quality*. The optimization objective is to find tilt values with the maximum coverage quality.

This optimization objective is intrinsically geospatial. Changes in sector coverage and energy distribution depend on the distance from the antenna, the antenna height, the terrain, geospatial entities in the area (like buildings), and other environmental features (like the reflection factors of the geospatial objects). When the geospatial environment is complex and the number of antennas increases, the effect of changes in tilt on the coverage quality is non-monotonic and there is no simple function to tell how a change in tilt will affect the coverage quality. We need to simulate each case by Rhea and evaluate each setting independently.

Problem Definition. Computing the maximum coverage quality is hard in large-scale settings. So, our goal is to search for the best solution that can be discovered fast. In an industrial setting, we may be given a limited amount of time τ for the computation and the goal is to compute the best solution within this time frame. The problem that we are studying is as follows. Given an area, the antennas in it, and a computation-time threshold τ, compute, within time τ, the tilt levels with the highest coverage quality.

4 ALGORITHMS

In this section we present our method for tilt optimization by using reinforcement learning. We also present three baseline methods to which we compare our method.

4.1 Reinforcement Learning

Our method is based on the classic multi-agent deep reinforcement learning (MADRL) framework [13]. Intuitively, reinforcement learning (RL) is learning by trial and error. In each step, multiple agents take actions independently and apply changes to the environment. The agents observe the new states and receive rewards accordingly. The history of the changes and the rewards are recorded and used to train a model for maximizing the rewards.

In our setting, for every antenna there are 15 tilt levels, where 0 is an up tilt towards the horizon and 14 is a down tilt towards the ground. An agent is assigned to each antenna. An action of an agent is an increase (decrease) of the tilt level by 1. So, for n antennas, in each step there are roughly 3^n combinations of possible changes, where each antenna can be tilted up, tilted down or have its tilt level kept unchanged. For n antennas with 15 tilt levels, the entire search space has a size of 15^n.

The environment consists of the geospatial model and the antennas (with their features). For every combination of tilt levels, we compute the coverage of each antenna and the state of each bin (dominance, interference, ...), by Rhea [10]. Note that the selection of the bin size is affected by the tradeoff between the accuracy of the model and the computation time. In practice, a partition into $16m \times 16m$ bins is sufficiently accurate for tilt optimization.

The algorithm gathers and manages the following information for each antenna A and bin b,

(1) the strength of the signal from A at b,
(2) the coverage quality of A at b,
(3) the strongest signal at b,
(4) the level of interference at b, and
(5) the *multi-path coefficient*, which is the difference between *(i)* the signal strength when considering all the paths from A to b, including indirect ones where reflection is involved, and *(ii)* the signal strength based on the direct path (line-of-sight) if such a path exists.

To increase the efficiency, the algorithm processes a subset of the bins in each step. This subset is selected uniformly and its size B is a hyperparameter of the algorithms.

Training. During training, the initial tilts are selected randomly (uniformly). This yields the initial environment state $S^{(init)}$. The overall coverage quality Q under this tilt setting is computed by Rhea. The goal is to find a tilt setting whose overall coverage quality is higher than $Q^{(init)}$ by an *improvement threshold* T.

At each step, every agent executes an *off-policy action* or an *on-policy action*. When an off-policy action is performed, a random action vector a is drawn uniformly from the action space. When an on-policy action is executed, a mini-batch of B bins is selected uniformly. Coverage quality, signal strength and interference are computed for the B bins, by Rhea, and the actions that increase coverage quality for these bins are computed and applied.

We use a three-layer feed-forward neural network [4] with ReLU activation [1] as the policy net. It uses a hyperparameter ϵ which is the probability of executing the off-policy action (while $1 - \epsilon$ is the probability of an on-policy action). The parameter ϵ decays as the number of training iterations grows. Thus, the effect of the decisions of the policy net grows over time.

In each step, we have the updated state and the previous state of the environment, denoted $S^{(u)}$ and $S^{(p)}$, with coverage qualities $Q^{(u)}$ and $Q^{(p)}$, respectively. For an agent A, $Q_A^{(u)}$ and $Q_A^{(p)}$ are the updated and previous coverage qualities in the bins where the signal from antenna A is the strongest (that is, bins in which A is the serving cell). We reward the agents for the way they improve $Q^{(u)}$ in comparison to $Q^{(p)}$. We design the reward function based on the following three principles.

Principle 1: the reward function encourages the agents to cooperate to improve the overall coverage quality. Our experiments show that if we only reward the agents for improving their individual coverage quality, they will execute greedy steps instead of improving the overall quality.

Principle 2: the reward function encourages the agents to have an appropriate individual coverage quality, not too high or too low. When an antenna serves too many bins it may exceed its communication capacity. Serving too few bins is a waste of energy.

Principle 3: the reward function discourages agents from competing for dominance in bin b if it is unlikely that they will be the serving cell of b. Our experiments show that if an antenna has little chance of serving a bin (for example, the line of sight is blocked by a building), encouraging it to compete for dominance will only increase interference.

To enforce these principles, we split the reward into five components: *step-wise penalty*, *individual-improvement reward*, *global-improvement reward*, *winning reward*, and *lose penalty*. Step-wise

Table 1: Components of the reward per agent

Component	Notation	Definition	Formulation
Step-wise penalty	S	A constant penalty per action. Encourages agents to change tilts.	Negative hyperparameter if $Q^{(u)} - Q^{(p)} \leq 0$, 0 otherwise
Individual-improvement reward	II	Increasing coverage quality for agent A.	$II = (Q_A^{(u)} - Q_A^{(p)})/T$
Global-improvement reward	GI	Increasing overall coverage quality. Encourages agents to cooperate.	$GI = (Q^{(u)} - Q^{(p)})/T$
Winning reward	WR	Improving the initial coverage quality by a margin of the winning threshold T. Encourages the agents to cooperate.	Positive hyperparameter if $Q^{(u)} - Q^{(init)} > T$, 0 otherwise.
Lose penalty	LP	Penalty for under-serving or over-serving. Agent A has its under-serving threshold $T_A^{(under)}$ and over-serving threshold $T_A^{(over)}$.	Negative hyperparameter if $Q_A^{(u)} < T_A^{(under)}$ or $Q_A^{(u)} > T_A^{(over)}$

penalty encourages the agents to take active actions instead of remaining still, but can also discourage agents to compete for dominance in bins that they cannot serve, because taking actions in that way will only result in receiving step-wise penalties. The individual-improvement reward makes agents explore cases of under-serving and avoid them. Global-improvement reward compensates agents which improve the overall coverage quality by helping other agents, such as reducing interference or lack-of-dominance. This encourages agents to cooperate. Winning reward is large, compared to the other rewards, and its goal is to make cooperation more profitable. Lose penalty penalizes under-serving and over-serving agents, to enhance individual-improvement reward. Table 1 is a summary of the notations, definitions and formulations of these components.

We noticed that it boosts training to scale up the global improvement reward when the overall coverage quality is low, because this encourages the policy net to explore larger parts of the search space, by exploring new areas, without penalty. To do so, ψ is a step function of the improvement ratio $P = (Q^{(u)} - Q^{(init)})/T$. It measures how close the current state is to the goal. Similarly, it helps to scale up the global improvement reward when the overall coverage quality is close to winning, when it is difficult to improve the coverage quality. At that stage, the agent needs a large reward to make progress. The values of ψ are as follows.

P	≤ 0.05	(0.05, 0.1]	(0.1, 0.2]	(0.2, 0.5]	(0.5, 0.8]	> 0.8
ψ	20	10	5	2	1	$(1+p)^2$

The reward r_A for agent A is computed by

$$r_A = II + GI \cdot \psi + WR + LP + S$$

After computing the reward, all the information (actions, updated tilts, updated environment state, and rewards) is saved as a record of history, and the next step is executed. The process terminates if the overall goal of coverage-quality improvement is achieved, or if the number of steps exceeds a threshold s_{max}. After termination, the tilts are initialized randomly, and the process is repeated, until the number of records exceeds a limit M on the history length.

The history records are fed into the policy net, and the model learns from past trials what actions may lead to higher rewards. Specifically, the loss and gradients are computed by comparing the reward at each step with the expected reward (i.e., the expectation of rewards across the entire mini-batch) using smooth L1 loss.[5] A reward discount factor γ is applied to weigh short-term rewards and long-term rewards—smaller γ means that the significance of the rewards decreases faster over time. Eventually, the weights of the policy net, denoted W_p, are copied and then updated by gradient descent with learning rate l. The result weights are called the *target net weights* and denoted W_t. Then, the policy net weights are updated by setting them as $W_p = W_p + \tau * W_t$, where τ is a hyperparameter called the *update rate*. Finally, the history is cleaned and the learning process repeats. Each training iteration is called an *epoch*. The process is repeated until convergence (no change anymore) or when reaching a limit E on the number of epochs.

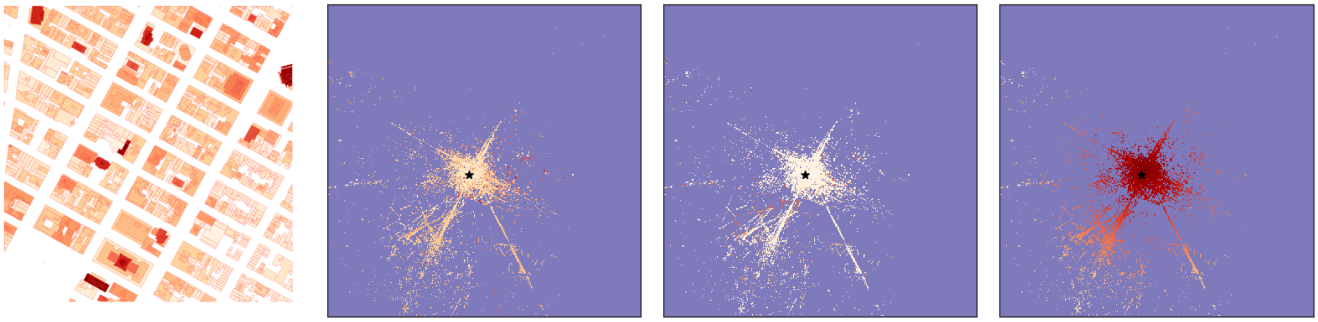
Testing. Testing is similar to training, with two differences. (1) The policy net is fixed. (2) All the actions are on-policy. Note that in both training and testing, there is no specific information about the region of interest involved, such as locations of buildings and trees, however, the effect of the environment is expressed by the Rhea multi-path coefficients. Therefore, a policy net trained on data obtained from a region R_1 can be tested on data obtained from another region R_2, as long as Rhea has the geospatial information of R_2. Hence, our method is generally applicable—models can be applied to new areas that were not included in the training.

4.2 Geospatial Significance Sampling (GSS)

To reduce the computational complexity, we only observe a subset of the bins in each step. But selecting the bins randomly often yields a non-representative set, because many bins have a minor effect on the training. For example, bins that are near a single antenna or bins that are in an area with no signal at all have a small effect on the optimization, while bins that are covered by several antennas are likely to be affected by tilt changes. To effectively select bins for the optimization, we use a selection that is similar to importance sampling [18] and learning from hard samples [9].

To select the bins for B, we compute the average variances as an indicator of informativeness. For every antenna A and bin b, we use the ray tracer to compute the variabce $v_{A,b}$ of the signal strengths across all tilt levels. If the average variance $\bar{v}_b = \frac{1}{n}\Sigma_{A=1}^{n} v_{A,b}$ over all the n antennas is below a threshold t_{GSS}, then bin b is considered non-informative. Hence, during training, the subsets of bins are

[5] https://pytorch.org/docs/stable/generated/torch.nn.SmoothL1Loss.html



(a) Geospatial Map (b) Signal strength variance, by Rhea (c) Rhea multi-path coefficient (d) Euclidean distance

Figure 14: Distribution of different forms of geospatial information and their alignment with the geospatial distribution of the variance of the signal strength, based on ray tracing. The black star is the location of the antenna.

drawn from the set $\{b \mid \bar{v}_b > t_{GSS}\}$ of bins b whose variance exceeds the threshold. In our experiments, we used $t_{GSS} = 3$ dBm.

We refer to this selection strategy as *geospatial significance sampling* (GSS), because it implicitly encodes the global geospatial information into the selection of training data. Ablation studies show that GSS significantly improves the performance of our optimization method. That is, policy nets trained on all the bins perform consistently worse than those trained only on informative bins.

4.3 Multi-Path Coefficient

Naive models of the signal strength only rely on the Euclidean distance between bins and antenna. Such models do not fully exploite the geospatial information and they neglect the effect of obstructions by geospatial objects on the signal strength in different bins. To examine the effect of the geospatial objects, we compute the *multi-path coefficient* of each bin. This coefficient is defined as follows. For antenna A and bin b, let $s_{A,b}$ be the strength of the signal from A at b while considering all the paths between A and b, including indirect paths due to reelection and refraction. Let $s'_{A,b}$ be the strength of the signal from A at b when considering only the direct path (when there is a line of sight between A and b). We define $r_{A,b} = 1 - s_{A,b}/s'_{A,b}$ as the multi-path coefficient. When $r_{A,b}$ is far from 0, there is a significant effect of the geospatial objects on the signal strength in the bin. When $r_{A,b}$ is positive it is typically because of blockage of the line of sight. When it is negative, it is because of reflections that add strength to the indirect signal. When $r_{A,b}$ is close to 0, the effect of the geospatial objects is small.

Fig. 14 illustrates the importance of the multi-path coefficient. Fig. 14a depicts the buildings in the region, where darker red represents higher buildings. The antenna location is marked by a black star. Fig. 14b presents the spatial distribution of the signal-strength variance $v_{A,b}$ of the antenna, as defined in Section 4.2. Darker red represents higher variance, that is, bins where the coverage quality is more likely to change when altering the antenna tilt. Fig. 14c shows the spatial distribution of the multi-path coefficients. Darker red represents higher values. In Fig. 14d, the colors are a function of the Euclidean distances between the bins and the antenna. Darker points are closer to the antenna. We can see that the spatial distribution of the multi-path coefficient aligns better with the ray-tracing results than with the Euclidean distances. Therefore, the policy net can use the multi-path coefficients to estimate which bins are more likely to be affected by antenna tilts and are significant for improving the overall coverage quality.

4.4 Baseline Algorithms

We compare reinforcement learning to three different baseline algorithms: greedy search, simulated annealing and Bayesian optimization. All of them were applied over the signal strengths computed by Rhea, using the same geospatial setting. We do not include here a comparison with complicated deep learning models such as convolutional neural network (CNN) and attention models because they are too large and too slow to meet our requirement of near real-time optimization. For example, generating a CNN-based embedding of the environment state alone takes more than 1 minute, and it does not scale well. Next, we briefly introduce the baseline methods and refer readers to external sources for more details.

4.4.1 Greedy Search. Greedy search is a naive optimization algorithm. An initial tilt setting is uniformly selected. The algorithm is given a limit s_{max} on the number of search steps. At each step, it computes the overall coverage quality after changing the tilt of a single antenna by at most one level. The selected change is the one that improves the coverage quality the most. For n antennas and 2 possible changes per antenna, at most $2n$ changes are examined in each step. If the highest quality score is gained for several different actions, one of them is selected arbitrarily. The search ends when no action improves the coverage quality (i.e., reaching a local maximum) or when the number of steps reaches s_{max}.

Greedy search has two disadvantages. 1) To complete one step of the greedy search, we need to evaluate the overall coverage quality approximately $2n$ times. 2) Often a greedy search gets stuck in a local maximum and misses the global maximum.

4.4.2 Simulated Annealing. Simulated annealing is a stochastic metaheuristic to approximate global optimization in a large search space [19]. Like greedy, it starts with a randomly-selected initial tilt setting and the number of search steps is limited by s_{max}. At step s, given a hyperparameter search range k, it randomly chooses an action a that changes at most k antennas, by at most one level each.

It computes a *temperature parameter* $t = 1 - s/s_{max}$, and accepts action a if $Q^{(u)} - Q^{(p)} > 0$, or otherwise, accepts a with probability $e^{-(Q^{(u)}-Q^{(p)})/t}$. If a is rejected, no action is executed in this step. The value k is gradually decreased with each iteration.

In each step, simulated annealing applies bigger changes than the greedy search and it typically explores a larger subspace of the search space. So, while it may also get stuck in a local maximum, it typically finds better solutions than the greedy search and finds these solutions faster.

4.4.3 Bayesian Optimization. Bayesian optimization is a statistical method that searches for the optima of a black-box parametric function, i.e., a multi-variate function without a closed-form. In our case, the overall coverage quality is such a black-box parametric function of tilts. The strategy is to think of the black-box function as a sample drawn from a stochastic process, which is called the *prior*. In this paper, we use a Gaussian process for the prior.

We start with a batch of n_{init} random tilt settings and compute the coverage quality of bins by Rhea. Then, we compute the posterior probability of observing these coverage quality scores, given a sample (i.e., a specification of the black-box function) from the prior. We choose the specification that has the highest probability as the estimation of the black-box function, and draw new random tilt settings based on the estimation. The process is repeated until it converges or reaches the step limit s_{max}. We use the `gp_minimize` function from the `scikit-optimize` package for Bayesian optimization. Other hyperparameters of the method are the acquisition function *acq* and the noise level l_n.

5 EXPERIMENTS

In this section we present our experimental evaluation and examine the ability of the algorithms to find an effective tilt setting.

5.1 Experimental Setting

Dataset. Our geospatial model is based on real open-source geospatial datasets, as described in Section 3. We selected an 8km × 8km region of New York, and placed antennas in this region (the antenna locations were selected randomly due to business confidentiality). We executed experiments with 16 antennas and 32 antennas. Each antenna has 15 tilt levels, labeled from 0 (horizontal) to 14 (14 degrees away from the horizontal line).

Machine. All the experiments were conducted on a Linux server with a Tesla V100 32GB GPU.

Code. The algorithms were implemented in Python where the scikit-learn package[6] was used for Bayesian optimization and the OpenAI Gymnasium interface[7] was used for implementing the reinforcement learning (RL) algorithm.

5.2 Training Schedule of RL

During training, for each epoch we selected the initial tilt setting and the winning threshold. There are 15^{16} (15^{32}) possible initial tilt settings, for 16 (32) antennas, so it is impractical to examine all the initial tilt settings during training. We found that compared to a complete random selection of the initial tilts, it is more efficient to train with a schedule, because the policy net needs to see the same initial tilt setting multiple times in a row to consolidate what it learns (like *learn-and-recap*). The learning goal changes from easy to hard, to avoid starting the training with a task that is too hard or provide a goal that is too easy at later stages.

Our training schedule is as follows. The total number of epochs is 2000. Every 20 epochs, a new initial tilt setting is selected uniformly. The initial winning threshold is set to be 100, and it is increased by 20 every 500 epochs. Other hyperparameters used during training are presented in Table 2.

Table 2: Training hyperparameters

Hyperparameter	Value
step-wise penalty S	-1
winning reward WR	100
lose penalty LP	-20
under-serving threshold $T^{(under)}$	20
over-serving threshold $T^{(over)}$	200
hidden dimension h	128
batch size B	500
maximum history length M	256
reward discount factor γ	0.99
initial off-policy probability ϵ	0.9
decay rate of ϵ	9×10^{-5}
update rate τ of the policy network	0.005
learning rate l	0.001

For more details on the hyperparameters, see [16].

5.3 Hyperparameters of Baseline Methods

Table 3 lists the choices of hyperparameters for the baseline models.

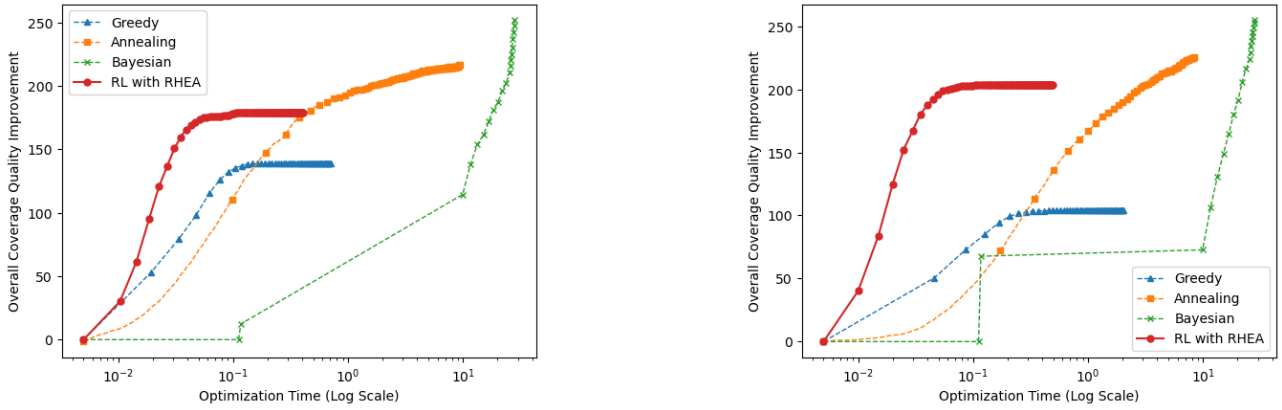
Table 3: Hyperparameters of the baseline methods

Method	Parameter	Value
Greedy Search	s_{max}	30
Simulated Annealing	s_{max}	300
Simulated Annealing	k	5 for steps 1-50 3 for steps 51-100 1 for steps 101-300
Bayesian Optimization	s_{max}	30
Bayesian Optimization	n_{init}	2
Bayesian Optimization	acq	EI
Bayesian Optimization	l_n	0.01

5.4 Results

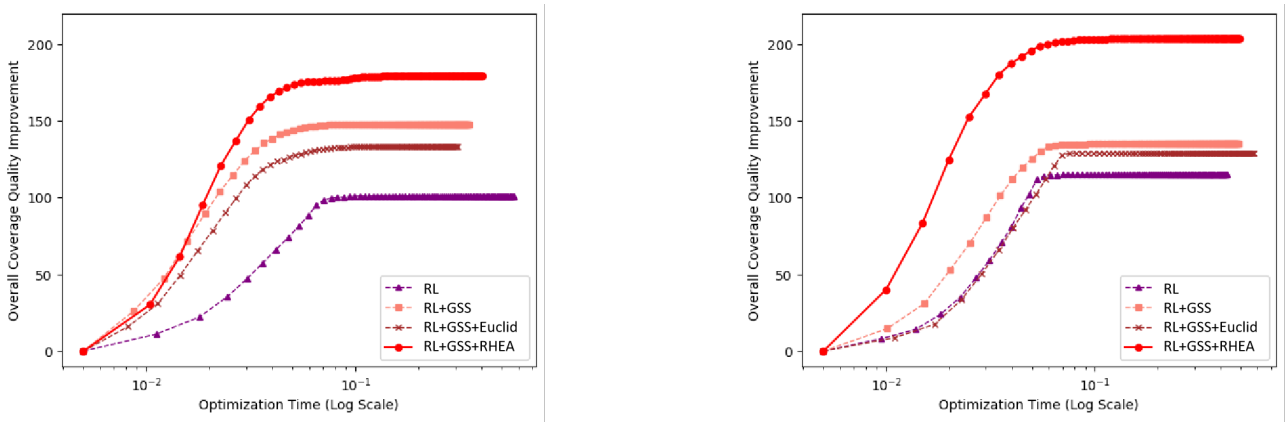
We evaluated the trained policy net against the baseline methods. All the results are the average over 1000 random initial tilt settings. Each initial tilt setting was tested in 5 independent runs, while considering two metrics: (1) overall coverage quality improvement; and (2) running time. There is a tradeoff between these metrics, so the goal is to have a good balance between them. The experiments show that our reinforcement model can optimize the antenna tilts effectively—much faster than the baseline methods.

[6] https://scikit-optimize.github.io/stable/modules/generated/skopt.gp_minimize.html
[7] https://www.gymlibrary.dev/index.html



(a) 16 Antennas **(b) 32 Antennas**

Figure 15: Performance and optimization time of reinforcement learning (RL) with RHEA vs baselines.



(a) 16 Antennas **(b) 32 Antennas**

Figure 16: Ablation study on the effect of geospatial significance sampling (GSS) and multi-path coefficient (RHEA)

Fig. 15a and Fig. 15b show the relation between optimization time and the improvement in coverage quality. The units of the x-axis are seconds, in log-scale. Each marker represents a single optimization step. The abrupt surge of the green curve at step 2 is due to the log-scale—the first two steps of the Bayesian optimization collect random samples without estimating Gaussian processes, and this takes around 0.1s while other steps take around 3s.

We can see the following in Fig. 15a and Fig. 15b. First, greedy search consistently achieves the worst improvement. This shows that our optimization problem is non-convex. Second, the reinforcement learning (RL) method improves the overall coverage quality by a large margin (50% to 100% more improvement than greedy search) very fast (in less than 0.05s). Third, in the 16-antenna case, simulated annealing takes 10× more time to achieve similar improvement as RL, and Bayesian optimization takes nearly 200× more time than RL for similar improvement. This gap increases to 40× and 400×, respectively, in the 32-antenna case. This is because simulated annealing takes significantly more steps to converge when the size of the search space increases, and the estimation process is longer for Bayesian optimization as the dimension of the search space increases. Fourth, though random annealing and Bayesian optimization can achieve higher improvement than RL if given sufficient time, we can see when comparing the 16-antenna case and the 32-antenna case that the optimization becomes more difficult as the number of antennas increases, because the size of the search space increases exponentially. For comparison, the policy net scales up well because it learns to trim the search space based on observations and to select the action based on features of the geospatial environment.

As an ablation study, we show in Fig. 16a and Fig. 16b the effect of geospatial significance sampling (GSS) and multi-path coefficient (denoted RHEA in the graphs) on the performances. In the graphs, **RL** is standard reinforcement learning with random selection of mini-batching and without multi-path coefficients as input. **RL+GSS** is RL with geospatial significance sampling but without multi-path coefficients as input. **RL+GSS+Euclid** is RL

with geospatial significance sampling where the multi-path coefficient is replaced by the Euclidean distances between the bin and the antennas. We refer to these variations as *partial methods*. **RL+GSS+RHEA** denotes the RL method we describe in Section 4, and we refer to it as the *complete method*.

Fig. 16a and Fig. 16b illustrate the importance of GSS and multi-path coefficients. First, the complete method outperforms the partial methods by a large margin. Second, RL with GSS outperforms standard RL by a large margin. Third, adding local geospatial information is not always helpful. When comparing RL+GSS and RL+GSS+Euclid, we see that the Euclidean distances do not provide useful information that the policy net can utilize to improve its decisions. They just introduce noise and degrade the training efficiency. Fourth, as the number of antennas increases, the gap between the complete method and the partial methods increases, while the gaps among the partial methods decreases. So, as the optimization problem becomes harder, neither global geospatial information nor local geospatial information alone is sufficient for the policy net to make correct decisions.

6 CONCLUSION

In this paper, we presented a novel method for optimizing the tilt of cellular antennas by using reinforcement learning trained over propagation models computed by a ray tracing module. The use of ray tracing provides the ability to train the model over a large variety of settings, for different parameters and different tilt levels. This makes the model accurate and reliable in comparison to models that are trained on a smaller set, without considering a large variety of parameters. We show that the use of reinforcement learning provides a favorable tradeoff between effectiveness (the level of optimization) and running time. This supports cases in which systems optimize large-scale areas or react quickly to changes in the network, like in the case of outage compensation and self-organizing networks. We show that the proposed approach is scalable and efficient and that it outperforms optimization based on simulated annealing and Bayesian optimization. Future work includes taking user movement and user density into account when optimizing the network.

REFERENCES

[1] Abien Fred Agarap. 2019. Deep Learning using Rectified Linear Units (ReLU). arXiv:1803.08375 [cs.NE]

[2] Vahab Akbarzadeh, Christian Gagne, Marc Parizeau, Meysam Argany, and Mir Abolfazl Mostafavi. 2012. Probabilistic sensing model for sensor placement optimization based on line-of-sight coverage. *IEEE transactions on instrumentation and measurement* 62, 2 (2012), 293–303.

[3] Mehdi Amirijoo, Ljupco Jorguseski, Remco Litjens, and Lars-Christoph Schmelz. 2011. Cell outage compensation in LTE networks: algorithms and performance assessment. In *Proceedings of the 73rd Vehicular Technology Conference*. IEEE.

[4] George Bebis and Michael Georgiopoulos. 1994. Feed-forward neural networks. *IEEE Potentials* 13, 4 (1994), 27–31.

[5] Philip E Brown, Krystian Czapiga, Arun Jotshi, Yaron Kanza, and Velin Kounev. 2020. Interactive testing of line-of-sight and Fresnel zone clearance for planning microwave backhaul links and 5G networks. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*.

[6] Philip E. Brown, Krystian Czapiga, Arun Jotshi, Yaron Kanza, Velin Kounev, and Poornima Suresh. 2020. Large-Scale Geospatial Planning of Wireless Backhaul Links. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '20)*. ACM.

[7] Philip E Brown, Krystian Czapiga, Arun Jotshi, Yaron Kanza, Velin Kounev, and Poornima Suresh. 2023. Planning Wireless Backhaul Links by Testing Line of Sight and Fresnel Zone Clearance. *ACM Transactions on Spatial Algorithms and Systems* 9, 1 (2023), 1–30.

[8] Philip E. Brown, Yaron Kanza, and Velin Kounev. 2019. Height and Facet Extraction from LiDAR Point Cloud for Automatic Creation of 3D Building Models. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '19)*. ACM.

[9] Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. 2017. Active Bias: Training More Accurate Neural Networks by Emphasizing High Variance Samples. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA.

[10] Krystian Czapiga, Serkan Isci, Yaron Kanza, James T. Klosowski, Velin Kounev, and Gopal Meempat. 2022. Playable Ray Tracing for Real-Time Exploration of Radio Propagation in Wireless Networks. In *Proc. of the 30th International Conference on Advances in Geographic Information Systems*. ACM.

[11] Ryan M Dreifuerst, Samuel Daulton, Yuchen Qian, Paul Varkey, Maximilian Balandat, Sanjay Kasturia, Anoop Tomar, Ali Yazdan, Vish Ponnampalam, and Robert W Heath. 2021. Optimizing coverage and capacity in cellular networks using machine learning. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 8138–8142.

[12] Harald Eckhardt, Siegfried Klein, and Markus Gruber. 2011. Vertical antenna tilt optimization for LTE base stations. In *Proc. of the 73rd Vehicular Technology Conference (VTC Spring)*. IEEE.

[13] Sven Gronauer and Klaus Diepold. 2022. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review* (2022), 1–49.

[14] Akhil Gupta and Rakesh Kumar Jha. 2015. A survey of 5G network: Architecture and emerging technologies. *IEEE access* 3 (2015), 1206–1232.

[15] Teyu Hsiung and Yaron Kanza. 2019. SimCT: Spatial simulation of urban evolution to test resilience of 5G cellular networks. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on GeoSpatial Simulation*. 1–8.

[16] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. 1996. Reinforcement learning: A survey. *Journal of artificial intelligence research* 4 (1996).

[17] Yaron Kanza, David Gibbon, Divesh Srivastava, Valerie Yip, and Eric Zavesky. 2020. Smartmedia: Locally & Contextually-Adapted Streaming Media. In *Proc. of the 28th ACM International Conf. on Advances in Geographic Information Systems*.

[18] Angelos Katharopoulos and François Fleuret. 2019. Not All Samples Are Created Equal: Deep Learning with Importance Sampling. arXiv:1803.00942 [cs.LG]

[19] Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. 1983. Optimization by simulated annealing. *science* 220, 4598 (1983), 671–680.

[20] Paulo Valente Klaine, Muhammad Ali Imran, Oluwakayode Onireti, and Richard Demo Souza. 2017. A survey of machine learning techniques applied to self-organizing cellular networks. *IEEE Communications Surveys & Tutorials* 19, 4 (2017).

[21] Donald Meagher. 1982. Geometric modeling using octree encoding. *Computer graphics and image processing* 19, 2 (1982), 129–147.

[22] Alan Mislove, Massimiliano Marcon, Krishna P Gummadi, Peter Druschel, and Bobby Bhattacharjee. 2007. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*.

[23] Fabian Schneider, Anja Feldmann, Balachander Krishnamurthy, and Walter Willinger. 2009. Understanding online social network usage from a network perspective. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*.

[24] Cigdem Sengul, Aline Carneiro Viana, and Artur Ziviani. 2012. A survey of adaptive services to cope with dynamics in wireless self-organizing networks. *ACM Computing Surveys (CSUR)* 44, 4 (2012), 1–35.

[25] Joseph A Shaw. 2013. Radiometry and the Friis transmission equation. *American journal of physics* 81, 1 (2013), 33–37.

[26] Muhammad Naseer ul Islam and Andreas Mitschele-Thiel. 2012. Reinforcement learning strategies for self-organized coverage and capacity optimization. In *2012 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE.

[27] Jacobus Van der Merwe, Subhabrata Sen, and Charles Kalmanek. 2002. Streaming video traffic: Characterization and network impact. In *Proceedings of the Seventh International Web Content Caching and Distribution Workshop*.

[28] Filippo Vannella, Grigorios Iakovidis, Ezeddin Al Hakim, Erik Aumayr, and Saman Feghhi. 2021. Remote electrical tilt optimization via safe reinforcement learning. In *2021 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE.

[29] Filippo Vannella, Jaeseong Jeong, and Alexandre Proutiere. 2020. Off-policy learning for remote electrical tilt optimization. In *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*. IEEE.

[30] Yunchou Xing, Ojas Kanhere, Shihao Ju, and Theodore S Rappaport. 2019. Indoor wireless channel properties at millimeter wave and sub-terahertz frequencies. In *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 1–6.

[31] Zehui Xiong, Yang Zhang, Dusit Niyato, Ruilong Deng, Ping Wang, and Li-Chun Wang. 2019. Deep reinforcement learning for mobile 5G and beyond: Fundamentals, applications, and challenges. *IEEE Vehicular Technology Magazine* 14, 2 (2019).

[32] Kun Yang, Cong Shen, and Tie Liu. 2020. Deep reinforcement learning based wireless network optimization: A comparative study. In *INFOCOM Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE.