

AIRMap — AI-Generated Radio Maps for Wireless Digital Twins

Ali Saeizadeh[†], Miead Tehrani-Moayyed[†], Davide Villa[†], J. Gordon Beattie, Jr.*,

Pedram Johari[†], Stefano Basagni[†], Tommaso Melodia[†]

[†]Institute for the Wireless Internet of Things, Northeastern University, Boston, MA, U.S.A.

*VIAVI Solutions, Inc.

E-mail: [†]{saeizadeh.a, tehranimoayyed.m, villa.d, p.johari, er.anderson, s.basagni, melodia}@northeastern.edu,

*gordon.beattiejr@viavisolutions.com

arXiv:2511.05522v1 [eess.SP] 28 Oct 2025

Abstract—Accurate, low-latency channel modeling is essential for real-time wireless network simulation and digital-twin applications. Traditional modeling methods like ray tracing are however computationally demanding and unsuited to model dynamic conditions. In this paper, we propose AIRMap, a deep-learning framework for ultra-fast radio-map estimation, along with an automated pipeline for creating the largest radio-map dataset to date. AIRMap uses a single-input U-Net autoencoder that processes only a 2D elevation map of terrain and building heights. Trained and evaluated on 60,000 Boston-area samples, spanning coverage areas from 500 m to 3 km per side, AIRMap predicts path gain with under 5 dB RMSE in 4 ms per inference on an NVIDIA L40S —over 7000× faster than GPU-accelerated ray tracing based radio maps. A lightweight transfer learning calibration using just 20% of field measurements reduces the median error to approximately 10%, significantly outperforming traditional simulators, which exceed 50% error. Integration into the Colosseum emulator and the Sionna SYS platform demonstrate near-zero error in spectral efficiency and block-error rate compared to measurement-based channels. These findings validate AIRMap’s potential for scalable, accurate, and real-time radio map estimation in wireless digital twins.

I. INTRODUCTION

Digital twins have emerged as transformative tools in wireless networks, creating virtual replicas—or “multi-verse”—of physical deployments that operate alongside the real world. Real-time digital twins let developers design, test, and evaluate “what-if” scenarios without risking live operations [1], [2]. Furthermore, by operating ahead of real time, they can forecast potential events, and in a fully closed-loop implementation, continuously ingest live data, update their models, and even trigger actions back in the physical network.

Real-time digital twins support a broad spectrum of wireless network use cases by leveraging continuously updated environmental models and fast inference engines. For instance, a Frequency Division Duplexing (FDD) massive MIMO base station can offload downlink channel estimation—or at least its dominant subspace—to the digital twin, dramatically reducing pilot and feedback overhead in the physical layer [3]. In

This work was supported in part by VIAVI Solutions, Inc.; by the Public Wireless Supply Chain Innovation Fund (PWSCIF) of the National Telecommunications and Information Administration (NTIA) under Award No. 25-60-IF011; and by the U.S. National Science Foundation under Grant CNS-1925601.

the radio interface, mobile users benefit from proactive Line-of-Sight (LOS) blockage prediction: by tracking scatterer motion in the twin, the system can initiate beam handovers or reconfiguration before link degradation occurs [4]. At the network layer, gathering site-specific datasets and sensing can be used to train ML models that achieve over 90% top-2 beam prediction accuracy with minimal real-world fine-tuning, enabling rapid beam management and resource allocation [5]. Finally, at the application layer, predicted link disruptions drive pre-caching strategies—for example, buffering video segments ahead of an anticipated outage—to maintain seamless user experiences under stringent latency constraints [6]. These examples demonstrate how real-time digital twins enable end-to-end optimization and adaptive control across all layers of wireless networks.

While traditional network simulators and digital twins both create virtual representations of networks, they fundamentally differ in their relationship with physical systems. Conventional simulators operate in isolation, running predefined scenarios with static inputs to predict theoretical outcomes [7]. In contrast, digital twin networks establish a bidirectional connection with their physical counterparts, continuously ingesting real-time data to create high-fidelity representations that evolve alongside the actual network [8]. This interactive mapping enables digital twins to not only reflect the current state of the network but also to provide closed-loop automation capabilities, where changes validated in the virtual environment can be safely applied to the physical network. This real-time synchronization capability makes digital twins particularly valuable for mission-critical applications in next-generation wireless networks, where they can facilitate network optimization, predictive maintenance, and innovative service development without risking operational disruptions [7].

In modern Radio Access Network (RAN) architectures—including the Open RAN (O-RAN) framework—the threshold for “real-time” performance varies by control-loop function and application. Non-real-time RAN Intelligent Controller (RIC) applications (*rApps*) operate on timescales exceeding one second for long-term optimization. Near-real-time functions (*xApps*) execute within ten to one thousand milliseconds for time-sensitive control [9]. The most stringent tier, distributed applications (*dApps*), enables sub-ten-millisecond

response times [10], [11]. Accordingly, digital-twin systems must meet these diverse latency targets, with the strictest real-time requirement—data acquisition and response—set below ten milliseconds for dApp-level operations. Thus, to achieve a digital twin framework that is effective across all layers of the protocol stack, accurate and agile channel modeling is essential to precisely and swiftly characterize the radio signal propagation through a dynamic environment. This constitutes the basis of a high-fidelity digital twinning system [7].

Channel modeling serves as the foundational layer upon which all digital twin capabilities are built. Without accurate propagation models, digital twins cannot reliably predict network behavior, optimize resource allocation, or trigger preemptive actions in the physical network. The challenge lies in achieving the dual requirements of high-accuracy and ultra-low-latency: channel models must be computed fast enough to support real-time decision making while maintaining sufficient fidelity to ensure reliable network operations. This creates a fundamental tension between computational complexity and model accuracy that existing approaches struggle to resolve.

A Primer on Current Approaches to Channel Modeling.

Channel modeling for RF scenarios traditionally falls into three main categories: measurement-based models, statistical models, and deterministic methods (e.g., ray-tracing). Measurement-based approaches capture site-specific phenomena with high accuracy but are costly, labor-intensive, and quickly outdated in dynamic environments [12]. Statistical channel models employ stochastic or deterministic mathematical equations to characterize wireless propagation [13]. However, these models often fail to represent all scenarios or capture environmental intricacies. Their reliance on simplified environmental assumptions leads to prediction inaccuracies, particularly in site-specific scenarios [14]. Ray tracing provides a deterministic means of modeling wireless channels by launching rays and simulating their interactions—reflection, diffraction, and transmission—with environmental geometries using material-specific parameters [15]. Although it achieves higher site-specific accuracy than empirical models—especially in architecturally complex settings—it remains impractical for real-time use. Even with GPU-accelerated implementations, the computational burden of dynamic mobility modeling and scenario adaptation is prohibitive, and the exhaustive pre-computation generally demands extensive storage. Furthermore, its fidelity depends on highly detailed 3D maps with accurate material assignments and accurate antenna patterns, since electromagnetic responses vary noticeably across surfaces. Therefore, routinely, in a trade-off against computational burden, ray tracing remains an approximation—constrained by finite ray sampling and often incomplete modeling of propagation phenomena like diffraction, scattering, and reflection for all possible points. These factors limit ray tracing’s suitability for high-fidelity channel modeling in digital twins.

Advanced propagation modeling attempts to find a balance between accuracy and computational efficiency. Deep Learning (DL) excels in this regard by training on extensive

propagation datasets to capture complex channel behaviors without explicit geometric simulation. Artificial Intelligence (AI)/Machine Learning (ML) frameworks can automatically learn nonlinear relationships among environmental features, measurements, and spatial dynamics. Consequently, they can deliver high-fidelity, real-time channel estimates, generalize across varied scenarios, and continuously refine their estimations as new data arrive—all while sidestepping the computational burdens inherent in conventional approaches [16], [17].

Radio (environment) maps provide a two-dimensional representation of averaged statistics of channel characteristics (e.g., received signal power, interference power, power spectral density, delay spread, and channel gain) over a geographic region [18]. Unlike individual point-wise channel estimates, radio maps capture spatial relationships and large-scale propagation patterns, reflecting how neighboring locations influence one another. This inherent spatial structure makes radio maps a natural and effective output format for DL models, which can leverage locality and spatial dependencies. For instance, Convolutional Neural Network (CNN) architectures can process environmental inputs such as terrain and building layouts to generate channel predictions for an entire area in a single inference step, rather than predicting each point independently. This structured approach not only improves scalability and inference speed but also provides the spatial context essential for real-time digital twin applications.

This paper presents **AIRMap**, a deep-learning-based framework for real-time, high-fidelity radio map estimation. AIRMap is trained on the largest site-specific dataset of its kind, automatically generated through a scalable pipeline using ray-tracing simulations and 2D elevation data. Unlike prior models requiring multiple inputs, AIRMap leverages a single-channel elevation map to produce accurate channel predictions with sub-5 dB RMSE and millisecond-level inference latency. A lightweight transfer-learning calibration procedure using sparse field measurements significantly improves accuracy, reducing median error to 10%. AIRMap is integrated into two state-of-the-art platforms—Sionna SYS and the Colosseum testbed—where it achieves near-zero error on spectral efficiency and block error rate metrics, validating its suitability for real-time digital twin applications across protocol layers.

The main contributions of this paper are as follows:

- **Ray-Tracing Efficiency Analysis:** We systematically analyze the computational complexity-fidelity trade-offs in ray-tracing simulations, identifying optimal configurations that enable large-scale dataset generation while maintaining high fidelity for deep learning applications.
- **Large-Scale Radio Map Dataset:** We develop an automated pipeline to generate the largest site-specific radio map dataset to date, comprising 60,000 Boston-area samples with diverse propagation scenarios for robust neural network training.
- **Variable-Scale Coverage Modeling:** Each sample covers a square region with side lengths ranging from 500 m to 3 km, allowing the model to generalize across both local and wide-area propagation conditions.
- **Single-Input U-Net Architecture:** We design a novel neural network architecture that requires only 2D eleva-

tion maps as input, achieving sub-5 dB RMSE with 4 ms inference time—over 7000× faster than GPU-accelerated ray tracing.

- **Fine-tuning:** We develop a novel calibration framework that leverages large-scale simulated ray-tracing data for pretraining, followed by fine-tuning with a small subset of real-world measurements. This approach effectively bridges the simulation-to-reality gap, enabling practical deployment with minimal measurement overhead while preserving model generalization and accuracy.
- **Real-Time Testbed Integration:** We demonstrate the first sub-10ms radio map generation in operational wireless testbeds (Colosseum and Sionna SYS), enabling true real-time digital twin applications with near-zero system-level performance error.

The remainder of this paper is organized as follows. Section II evaluates conventional channel modeling techniques, focusing on ray tracing and its limitations through a real-world measurement campaign. We identify key factors—such as material properties and antenna patterns—that contribute to the gap between simulated and measured propagation data. In Section III, we present our automated pipeline for generating a large-scale dataset using Sionna RT [19], detailing trade-offs between ray-tracing fidelity and computational efficiency. We also describe the dataset structure and its suitability for deep learning. Section IV introduces AIRMap, our U-Net-based AI framework for real-time radio map estimation. We outline the model architecture, training approach, and a lightweight calibration pipeline using sparse measurements to enhance accuracy. In Section V, we demonstrate the integration of AIRMap into the Sionna SYS simulator and the Colosseum testbed, validating its accuracy and real-time performance in system-level simulations and channel emulation. Finally, Section VI concludes our contributions.

II. PRACTICAL ANALYSIS OF RAY-TRACING-BASED CHANNEL MODELS

In this section, we analyze a measurement scenario to assess the impact of simulation parameters on the gap between ray-tracing predictions and real-world measurements. Since simulation data forms the foundation for training the DL model, we propose a calibration approach that uses a small set of field measurements to fine-tune the model. This enables the data-driven model to surpass ray tracing in accuracy by correcting for systematic simulation errors.

Measurement Campaign. We design a measurement scenario to capture channel characteristics at 933 MHz within a section of the Northeastern University campus. In this setup, the TX remains stationary on the rooftop of Dodge Hall, while the RX is mobile, following the route depicted in Fig. 1.

For the TX side, we used an Ettus USRP X410 as the Radio Unit (RU), followed by a Minicircuits ZHL-1000-3W+ amplifier, and a Pasternack PE510M1014 antenna. On the RX side, we employed the same antenna as used in the TX side. The RX RU was the Viavi Solutions T/Rx Software Defined Transceiver [20].

For channel measurements, we utilized the maximum available bandwidth of 2 MHz at a center frequency of 933 MHz.

Transmission was conducted in bursts of 500 consecutive GLFSR-14 codewords every 0.1 seconds. All losses and gains from cables, amplifiers, antennas, and other hardware components were removed by characterizing the entire setup in an anechoic chamber under identical conditions. A summary of the measurement setup is provided in Table I.

Path Gain Computation from CIR. We compute the path gain directly from the measured Channel Impulse Response (CIR).

$$h(t, \tau) = \sum_{i=1}^N \alpha_i(t) \delta(\tau - \tau_i(t)) \quad (1)$$

where $\alpha_i(t)$ is the complex amplitude and $\tau_i(t)$ is the delay of the i -th path at time t .

The total path gain at each instant can then be computed by integrating the squared magnitude of the CIR.

$$P_{\text{RX}}(t) = 10 \log_{10} \left(\sum_i |\alpha_i(t)|^2 \right) \quad (2)$$

In our mobile measurement scenario, the receiver's location changes over time; thus, we can associate each measurement with the receiver position $q_{\text{RX}}(t)$. Under this setting, the time-indexed path gain $P(t)$ can be equivalently expressed as a location-dependent function $P(q_{\text{RX}})$, allowing us to map the measured data directly into spatially indexed radio maps.

In Fig. 2, we compare ray-tracing (Sionna) results—generated using a maximum-detail configuration on the same locations as our measurements—to the empirical data. The 3D environment model was sourced from the Boston Planning Department [21], with all surfaces assigned concrete material properties and terrain modeled as dry ground. As shown in Fig. 2a, ray tracing and measured path gains exhibit substantially different levels. The scatter plot in Fig. 2b further illustrates that these differences persist across all sample points, deviating from the ideal linear relationship. This discrepancy highlights the limitations of ray tracing's

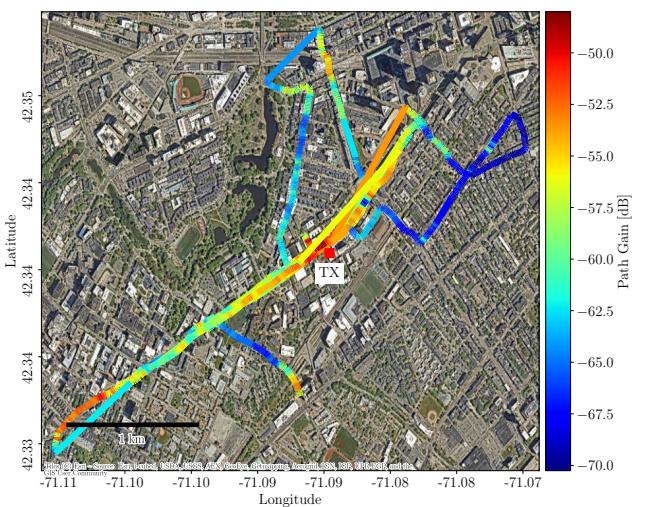
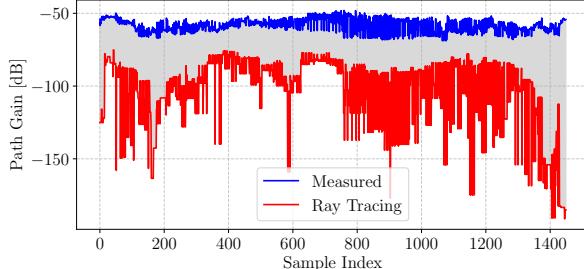
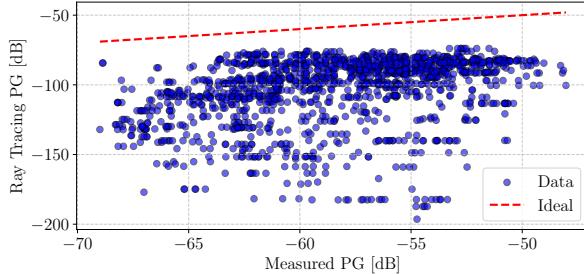


Fig. 1: Measurement campaign map showing the Transmitter (TX) location and Receiver (RX) route. The line color represents the path gain and coverage distribution.



(a) Comparison of measured and ray-traced path gain values across locations. Maximum normalized correlation is 0.3967 at zero lag.



(b) Scatter plot of the ray tracing results vs. measured values.

Fig. 2: Evaluation of ray tracing performance by comparing with actual measurement data.

approximations and its inability to fully capture real-world propagation.

Calibration techniques that adjust material properties can improve ray-tracing fidelity, but they are typically limited to small-scale indoor settings and do not extend well to large outdoor environments [22]–[24]. To quantify the effect of misconfigured materials and antenna patterns on path gain—especially when detailed environmental parameters are unknown—we performed a series of Sionna RT simulations. We conducted 50 different random scenarios with a maximum depth of 20 and diffraction enabled. In each run, all objects were initially assigned the ITU brick material model, and we then measured the resulting path-gain error assuming the true material was concrete. We repeated this procedure

for antenna patterns by comparing TR 38.901 specifications against an isotropic radiator to isolate the impact of antenna misconfiguration. The results, summarized in Fig. 3, illustrate how incorrect material assumptions or antenna-pattern settings degrade path-gain accuracy.

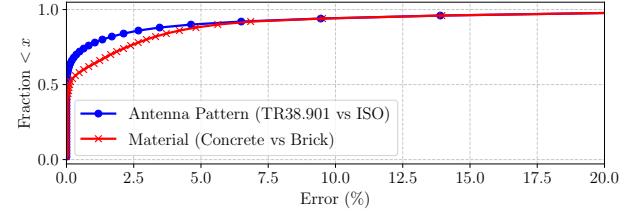


Fig. 3: Material and antenna misconfigurations Empirical Cumulative Distribution Function (eCDF) in ray tracing.

In Fig. 4, we plot the measured path gain values ordered by distance from the TX. The highly irregular urban geometry prevents a simple radial approximation, as signal propagation is neither symmetric nor uniform. Consequently, site-specific inputs—such as detailed environmental geometry—are essential. Traditional statistical models, which are not tailored to a particular site, fail to capture the complex, nonlinear propagation effects present in urban areas and therefore lack sufficient accuracy. It can be observed in Fig. 4 that it is not trivial to find a low-degree polynomial that fits the variations in the path gain at different distances from the transmitter.

III. DATASET

A. Efficient Generation of High-Fidelity Radio Maps

Ray-tracing offers high-fidelity channel modeling but remains computationally expensive, especially when creating the large-scale datasets needed for training DL models. To enable scalable dataset generation, we explore the trade-off between simulation accuracy and runtime by varying ray-tracing configurations.

Among various ray-tracing tools, we selected Sionna RT [19] (version 0.19.2) for its ease of automation, native Python interface, and GPU acceleration—features that streamline large-scale dataset generation. Built on top of Mitsuba 3, a differentiable rendering system powered by the just-in-time

TABLE I: Measurement Setup and Equipment

<i>TX</i>	
RU	Ettus USRP X410
Amplifier	Minicircuits ZHL-1000-3W+ (38 dB)
Antenna	Pasternack PE51OM1014 (6 dBi)
Location	42°20'25"N 71°05'16"W
<i>RX</i>	
RU	T/RX provided by VIAVI Solutions
Antenna	Pasternack PE51OM1014 (6 dBi)
Location	Mobile (see Fig. 1)
<i>Meas. Details</i>	
Frequency	933 MHz
Bandwidth	2 MHz
Codeword	GLFSR-14
Synchronization	GPS clock for both TX and RX

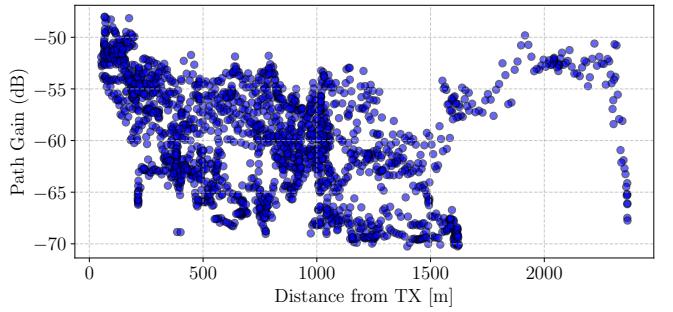


Fig. 4: Measurement results ordered by distance from the TX, highlighting the difficulty of fitting traditional empirical models to real-world data.

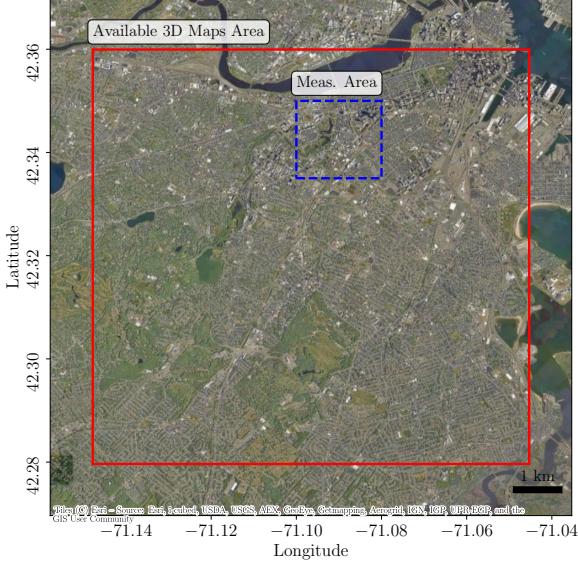


Fig. 5: Geographic study area in Boston. The red-outlined region marks the environment used for dataset generation, while the blue-highlighted region corresponds to the area used in the measurement campaign shown in Fig. 1.

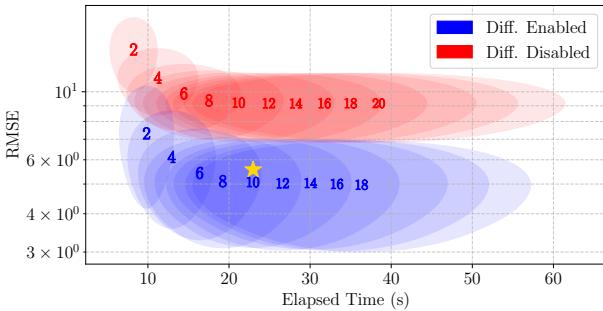


Fig. 6: Root Mean Squared Error (RMSE) versus runtime for varying ray-tracing configurations in Sionna. Marker labels denote maximum path depth (reflections). Shaded regions represent 95% confidence intervals.

compiler Dr.Jit, Sionna RT offers a flexible and efficient framework for radio propagation modeling, making it particularly well-suited for AI-driven wireless system design.

We use the Boston area shown in Fig. 5 as a representative urban environment and randomly select 100 TX locations placed outside buildings. For each ray-tracing configuration, we generate path gain radio maps and evaluate accuracy using the RMSE metric. The configuration with diffraction enabled and a maximum depth of 20 is used as the ground truth (maximum configuration). All other configurations are compared against this baseline to identify the most efficient setup that balances accuracy and simulation time. All simulations are conducted using Sionna RT on an NVIDIA L40S GPU. Reported runtimes exclude scene initialization and geometry loading to focus solely on ray-tracing execution.

Fig. 6 shows the simulation runtime for 100 scenarios (each repeated ten times, with shaded 95% confidence intervals)

across different ray-tracing configurations. As expected, enabling diffraction and increasing the maximum path depth both lead to longer runtimes. However, when diffraction is enabled, the RMSE levels off beyond a path depth of 10, indicating diminishing returns in accuracy. Fig. 7 illustrates how varying these parameters affects the resulting path-gain radio maps in a representative scenario. Based on this analysis, we identify a “sweet spot” configuration—diffraction enabled with a path depth of 10—that balances fidelity and efficiency, requiring approximately 30 seconds to generate each radio map.

B. Automated Generation of Large-Scale Datasets

For all TX locations within the Boston area (see Fig. 5), we load the scene geometry from the BostonTwin model [21] into Sionna RT with a carrier frequency of 1 GHz. Both transmitter and receiver use vertically polarized isotropic antennas, and the transmit power is fixed at 44 dBm to match typical urban macrocell deployments [25]. We then generate path-gain radio maps for each TX position.

For training data, we randomly select 12,000 valid TX locations within the rectangle defined by latitudes 42.2796° – 42.3599° N and longitudes 71.1478° – 71.0453° W, excluding points inside building footprints. Each sample comprises a pair of uniformly cropped, rasterized maps: (1) the path-gain radio map and (2) the corresponding 2D building elevation map, with the TX centered in both. To introduce variability in coverage area, the map extent for each sample is drawn uniformly between 500 m and 3 km. This pipeline can ingest any 3D urban dataset for which detailed geometry is available, enabling seamless generalization to new locations.

To mitigate overfitting and enhance model robustness, we apply geometric data augmentation: each sample is rotated by 90° , 180° , and 270° , and flipped horizontally and vertically. This expands the dataset to 60,000 samples, offering ample diversity for training deep-learning models. As a result, to our knowledge, this represents the largest radio-map dataset, with effectively unlimited extensibility in both geographic scope and sample count.

IV. DL-BASED ESTIMATION OF RADIO MAPS

A. Related Work on AI/ML-based Channel Models

In AI/ML-based channel modeling, a mapping function is established between the wireless environment and its corresponding channel properties, enabling accurate and realistic channel parameter representation. Over the past few years, numerous studies have explored and developed AI/ML-based channel models [33], [48], [49].

Early studies [40]–[42], [50] primarily employed conventional machine learning techniques—such as Random Forest, KNN, and Support Vector Machine (SVM)—for channel path loss prediction, with Random Forest generally achieving the best performance. Seretis et al [27] later compared Random Forest with XGBoost, integrating extensive feature engineering to reduce model complexity. Their results indicated that XGBoost yielded higher prediction accuracy. A detailed summary of these works is presented in Table III.

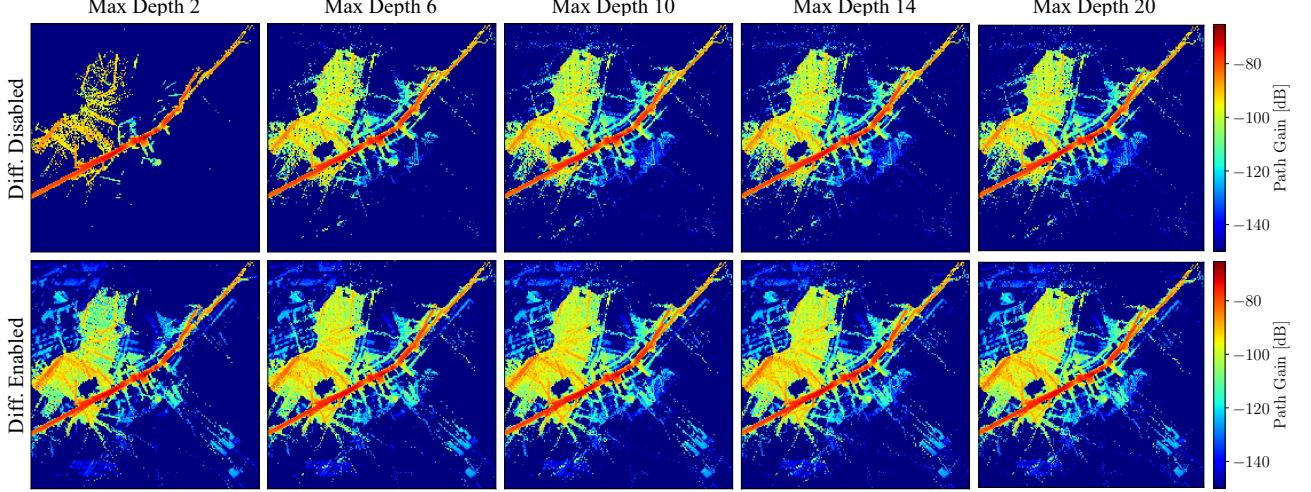


Fig. 7: Evolution of radio maps with increasing ray-tracing depth. Each column corresponds to a different maximum path depth. Rows compare configurations with diffraction disabled (top) and enabled (bottom).

Subsequent efforts shifted towards shallow neural network approaches. For instance, [44]–[46] investigated techniques such as RBF neural networks to predict time-varying path loss, shadow fading, and small-scale channel characteristics, including the number of propagation paths and angular statistics, particularly within Geometry-Based Stochastic Model (GBSM). Huang et al. [46] compared the performance of RBF networks with MLP neural networks, reporting that MLP slightly outperformed RBF by a fraction of a decibel.

Further, Zhang et al. [47] utilized an MLP neural network with additional environmental features, including terrain type and building occlusion, to predict the signal strength coverage within an urban scenario. Their model demonstrated superior accuracy compared to traditional propagation models.

The introduction of CNNs and their ability to learn spatial correlations in images led to a new generation of models that utilize images as input to neural networks. These approaches involve constructing images from propagation-related features, which CNNs analyze to extract spatial patterns, allowing the model to perform automated feature engineering and improve prediction accuracy.

Examples of image-based environmental representations include distance maps and building maps, as proposed by Imai et al. [26], as well as low-resolution building height information used in [43]. These CNN models are typically followed by FC layers for regression tasks, predicting channel parameters. Additionally, some studies have incorporated system parameters—such as frequency and antenna tilt—into the FC layers as extra features, as these parameters may not be easily represented within an image format.

Beyond structured maps, satellite and aerial images have also been integrated as additional input channels to CNN models [28], [29]. The impact of various environmental map images was explored in [28], while [32] examined how different image sizes and spatial map construction methods between TX and RX affect prediction accuracy. Their findings indicate that a building occupancy rate map is more effective than

aerial imagery and that including both RX and TX image data achieves similar performance to adding system parameters.

Hybrid approaches have also been explored, where a physics-based model is used to generate rough parameter estimates, which are then refined using a neural network. This correction-based method, proposed in [29], [31], improves prediction performance by leveraging domain knowledge in combination with deep learning. In these studies, rough estimates were integrated into the FC regression layer, whereas in [32], a heat map of rough estimates—constructed using a free-space model—was incorporated as an additional input channel for the CNN model.

Recent studies have focused on leveraging richer environmental information as input images to predict channel parameter heat maps in a single step. Works such as [33]–[39] have explored CNN-based autoencoders and U-Net architectures for fast and accurate predictions. While U-Net enhances prediction accuracy, it comes at the cost of increased computational complexity. A summary of DL-based methods for channel parameter prediction is presented in Table II.

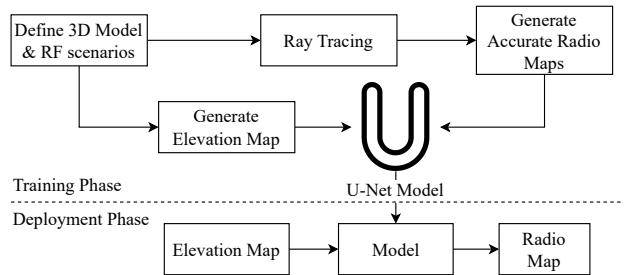


Fig. 8: DL-based model for radio map generation.

B. U-Net Architecture for Radio Map Estimation

We propose a DL framework for estimating radio maps from environmental context, comprising separate training and deployment phases (Fig. 8). During training, the model learns

TABLE II: Applications of Deep Learning in Channel Parameter Prediction

Ref.	AI/ML Methods	Inputs	Predicted Parameters	Frequency	Environment	Data Source	Best Performance
[26]	CNN + MLP	Distance map, building height	Path loss	2 GHz	Urban macrocell	Ray-tracing	RMSE = 9.94 dB
[27]	CNN, MLP	Building height map	Path loss	900 MHz	146 urban environments	Ray-tracing	RMSE = 4.42 dB
[28]	CNN + MLP	Aerial image, building info, distance, angle, system parameters	RSS	2 GHz	Urban (Tokyo)	Measurements	RMSE = 3.96 dB
[29]	Correctional CNN + MLP	Distance, relative geodetic distance, Rx coordinates, satellite image	RSRP	0.811, 2.63 GHz	Urban	Measurement, Ray-tracing	RMSE = 4.3 dB @ 0.811 GHz, RMSE = 4.24 dB @ 2.63 GHz
[30]	Correctional CNN + MLP	Delta lat/long, image of local surroundings, link budget rough estimate	RSS (RSSP)	0.811, 2.63 GHz	Urban, suburban, highway	Measurement	RMSE = 4.7 dB @ 0.811 GHz, RMSE = 4.1 dB @ 2.63 GHz, RMSE = 6.3 dB across environments
[31]	Correctional CNN + MLP	Distance, relative geodetic distance, Rx locations, satellite image	Path loss	3.5 GHz	Urban	Simulation + Measurement	RMSE = 4.46 dB
[32]	CNN + MLP	Spatial map, Tx-Rx spatial map, FSPL rough estimate map, system parameters	RSS	LTE 2.1 GHz	Urban (Tokyo)	Measurement	MSE = 8.07
[33]	CNN Autoencoder (AE)	Spatial map (LiDAR + CADMAPPER)	Path loss	28 GHz	Urban canyon	Measurements	RMSE = 4.8 dB
[34]	CNN AE + U-Net	Spatial map, distance	RSS map	-	-	-	MAE = 2.2 dB (avg.)
[35]	U-Net	Terrain height, building height, foliage height, LOS, BS height	Path loss	28 GHz	-	Ray-tracing	RMSE = 5.8 dB
[36]	U-Net	City map, Tx location, some path loss measurements	Path loss	5.9 GHz	Urban	Ray-tracing	-
[37]	U-Net	Permittivity, conductivity, distance, FSPL, spatial map	Path loss heat map	0.433, 2, 3.7 GHz	Indoor	Ray-tracing	RMSE = 4.21 dB
[38]	U-Net	Spatial map, Tx location	RSS map	2.5 GHz	Campus	Ray-tracing	NMSE = 0.0006752
[39]	U-Net	3-channel image (building height, Tx location Gaussian kernel, distance)	Path loss	30 GHz	Urban	Ray-tracing	RMSE = 5.78 dB

TABLE III: Applications of Conventional Machine Learning in Channel Parameter Prediction

Ref.	AI/ML Methods	Inputs	Predicted Parameters	Frequency	Environment	Data Source	Best Performance
[40]	Random Forest, AdaBoost, k-Nearest Neighbors (KNN), NN	Distance, path angle, vegetation, terrain complexity	RSSI	2.4 GHz	Suburban (ground-to-ground)	Measurement	MAE = 3.72 (Random For-est)
[41]	Random Forest, KNN	Path visibility, distance, elevation angle, Tx/Rx altitude	Path loss	2.4 GHz	Urban (air-to-air)	Ray-tracing	MAE = 2.27 (Random For-est)
[42]	Random Forest, KNN	Rx coordinates, building stats, elevation angle	Path loss, delay spread	2.4, 5.8, 28, 37 GHz	Urban (air-to-ground)	Ray-tracing	MAE = 1.64 (Random For-est)
[43]	XGBoost, Random Forest	23 features (LOS path, Rx area, Tx/Rx positions)	Path loss	900, 1800 MHz	Outdoor	Ray-tracing	MAE = 3.17 (XGBoost)
[44]	Radial Basis Function (RBF)	Snapshot index, distance	Path loss, shadow fading, DoA	26 GHz	Outdoor microcell	Measurement	RMSE = 4.52
[45]	RBF	Snapshot index, distance	Path loss, shadow fading, small-scale parameters	26, 28 GHz	Outdoor microcell, indoor	Measurement	RMSE = 0.44
[46]	RBF, Fully Connected (FC)	Tx/Rx coords, distance, carrier frequency	Received power, delay spread, angle spreads	11–60 GHz	Indoor	Measurement, GBSM model	RMSE = 2.63 (FC)
[47]	Multilayer Perceptron (MLP)	Distance, relative height, terrain, building occlusion	RSRP	2.5 GHz	Urban, dense urban	Measurement	MAE < 5

TABLE IV: Network Architecture of the Modified PMNet

Encoder		
#	Layer Type	Output Size
	Input: Image ($1 \times 200 \times 200$)	
1 (\downarrow)	Conv2d (64 ch), MaxPool2d	$64 \times 100 \times 100$
2	ResLayer (3 blocks)	$256 \times 100 \times 100$
3 (\downarrow)	ResLayer (3 blocks)	$512 \times 50 \times 50$
4 (\downarrow)	ResLayer (27 blocks)	$512 \times 25 \times 25$
5	ResLayer (3 blocks)	$1024 \times 25 \times 25$
6	ASPP + Conv2d (fc1)	$1024 \times 25 \times 25$

Decoder		
#	Layer Type	Output Size
	Output: Image ($1 \times 200 \times 200$)	
1 (\uparrow)	Conv2d	$512 \times 25 \times 25$
2 (\uparrow)	ConvTranspose2d	$512 \times 50 \times 50$
3 (\uparrow)	ConvTranspose2d	$256 \times 100 \times 100$
4	Conv2d	$256 \times 100 \times 100$
5	Conv2d	$128 \times 100 \times 100$
6	Conv2d + final head	$1 \times 200 \times 200$

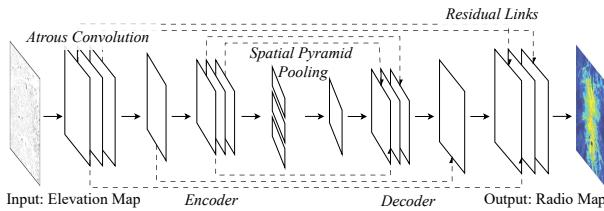


Fig. 9: Proposed U-Net architecture for channel modeling.

to generate a spatial map of a target channel parameter—such as path gain or RMS delay spread—for any specified TX location. Unlike our previous approach [51], which required two inputs (an initial rough radio-map estimate plus an elevation map) and incurred significant overhead, the new model uses only a single-channel 2D elevation image encoding terrain and building heights. The architecture extends an autoencoder inspired by PMNet [52], with adjusted encoder and decoder convolutional layers to process this one-channel input in place of PMNet’s original dual-input design. The network structure is shown in Fig. 9, with architectural details provided in Table IV.

Choosing an effective input space is critical for enabling the model to accurately learn channel characteristics. Prior work has explored a variety of input encoding strategies. For example, Lee et al. [52] used two input images: a building height heat map and a one-hot encoded image indicating the TX location. In contrast, Bakirtzis et al. [53] incorporated richer environmental descriptors such as conductivity, permittivity, relative distance, and free-space path loss maps, primarily for indoor scenarios.

In our approach, we define the input space as $\mathbf{x} = \mathbf{I}_{el}$, where \mathbf{I}_{el} represents a 2D elevation map derived from the scenario’s 3D model. This map encodes terrain elevation, building heights, and other physical obstructions. Although the model input has a fixed spatial dimension (e.g., 200×200 pixels), we vary the spatial resolution of each sample—ranging from 2.5 m/pixel to 15 m/pixel—so that the corresponding physical area spans from 500 m to 3 km per side. This resolution-

adaptive approach enables the model to learn propagation characteristics over different deployment scales

The elevation map for our scenario is depicted in Fig. 10. Through experimental analysis, we found that applying min-max normalization, which scales all values between 0 and 1, enhances model convergence. Additionally, we invert the elevation values so that taller buildings—being the primary obstructions in the coverage map—are assigned values closer to zero. This transformation ensures that the model learns the impact of major blockages more effectively.

Ensuring that the model accurately interprets the TX location is crucial. Our literature review identified two effective methods for encoding this information: (1) representing the TX location as a one-hot-encoded heat map, where a single pixel in the image corresponds to the TX position on the building map, and (2) centering the TX within the input image, ensuring that the building map is always aligned around the TX location. For this work, we adopt the latter, as described in Section III.

C. Experimental Results and Analysis

For model evaluation, the entire dataset is divided using 5-fold cross-validation. Within each fold, we follow a 70-15-15 split for training, validation, and testing, respectively. To prevent data leakage, we ensure that augmented versions of the same scenario do not appear in the other sets.

As shown in Fig. 11, the median error percentage across all folds and coverage maps remains around 4.43%, demonstrating the model’s high accuracy. The path gain values in the test dataset range from -150 dB to -50 dB, encompassing a wide variety of propagation conditions. Despite this large dynamic range, the model maintains consistent performance with minimal deviations across different test cases, highlighting its robustness in estimating radio maps with exceptionally low inference error in real-time. Additionally, Fig. 10 presents qualitative results showcasing the model’s performance across various scenarios, which were randomly selected from the Boston area.

The proposed model consists of 37,601,537 parameters, reflecting its capacity to learn complex spatial patterns in radio maps. To evaluate computational efficiency, we measured the inference time over 1,000 runs on an NVIDIA L40S GPU, achieving an average of 4.2 milliseconds per run. This low-latency performance makes the model well-suited for real-time or large-scale radio map estimation in wireless systems and digital twins.

D. Model Calibration

It is important to note that the ray-tracing simulations used for training were conducted at a frequency of 1 GHz, employing an isotropic antenna pattern. Additionally, the environmental geometry, while detailed, lacks full accuracy compared to the real-world scenario. Furthermore, the material properties assigned in the simulations differ from actual environmental materials, introducing discrepancies between simulated and measured propagation data. These factors contribute to the simulation-to-reality gap that our model aims to bridge.

To address these discrepancies and improve the real-world accuracy of our AI-driven model, we introduce a calibration pipeline for fine-tuning. The model is first pretrained on simulated radio maps to learn general channel characteristics. It is then fine-tuned using a small set of measurements to adapt to real-world conditions while preserving its generalization capabilities.

Given the cost and complexity of large-scale data collection, we intentionally use only a 20% subset of measurements for model calibration. This enables effective fine-tuning while preserving an unbiased portion of the data for testing. We adopt a geographically-aware train/test split that clusters points by spatial region. This approach is more appropriate for geographic data, as it enforces spatial separation between training and testing areas—thereby providing a more realistic evaluation of the model’s ability to generalize across space. To incorporate sparse measurements into training, we further modify the loss function using a weighted formulation that emphasizes measured points without overriding the knowledge learned from simulation.

This strategy enables efficient fine-tuning with minimal data, significantly improving predictive performance. The full calibration process is illustrated in Fig. 12.

We perform the calibration process 100 times to account for randomness in the train-test splits, ensuring the results are not biased toward any specific subset of the data. Fig. 13 shows the performance comparison between Sionna RT, the calibrated U-Net model, and the uncalibrated model. As shown, the calibrated model consistently outperforms the others, reducing

the error to around 10% and producing predictions that more closely align with real-world measurements.

This calibration pipeline allows the U-Net model to continuously incorporate real-world measurements, ensuring that the digital twin remains closely aligned with the physical network. By iteratively collecting field data, updating the model, and regenerating the radio maps, the system maintains high fidelity and accuracy for subsequent “what-if” analyses and control actions.

Despite Sionna RT’s support for gradient-based calibration of material and antenna parameters [23], we did not apply this process to our comparisons for two main reasons. First, the calibration methods in Sionna RT rely on full CIR and differentiable parametrizations of scattering and antenna patterns, whereas our measurements supply only scalar path-gain values. Without access to absolute delays, phases, or multi-tap CIR data, Sionna’s calibration pipeline cannot be directly applied. Second, the end-to-end differentiable calibration in Sionna RT incurs substantial computational overhead—requiring repeated ray tracing and backpropagation through complex computational graphs—which contradicts our real-time objectives. In contrast, our lightweight U-Net calibration uses only sparse path-gain samples and executes in milliseconds, ensuring both agility and fidelity for digital-twin applications.

V. RADIO MAPS FOR REAL-TIME DIGITAL TWINS

As discussed in Section I, the ultimate objective of these radio maps is to enable real-time estimation of channel parameters across a geographic area. The next step is to integrate

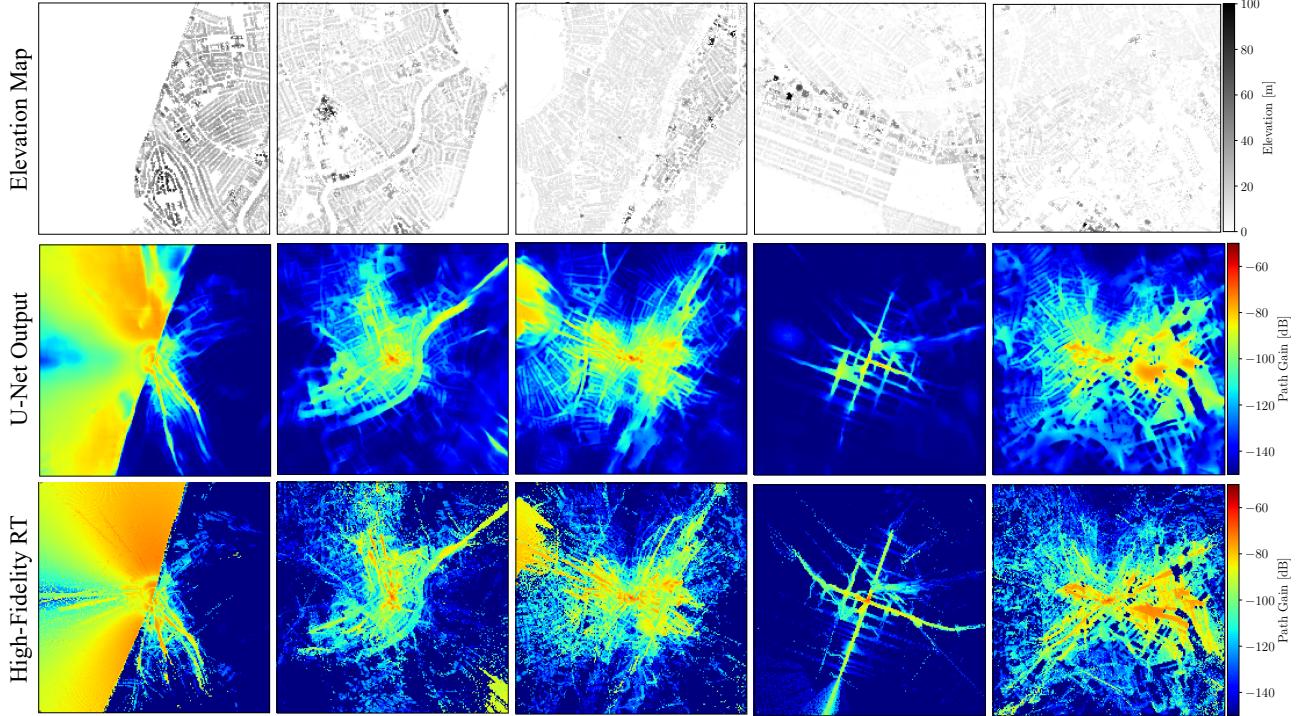


Fig. 10: Visual comparison of radio maps across scenarios, ranging from open to dense urban areas. The top row displays the 2D elevation map inputs, the middle row presents the U-Net model’s predictions, and the bottom row shows the corresponding high-fidelity ray-tracing.

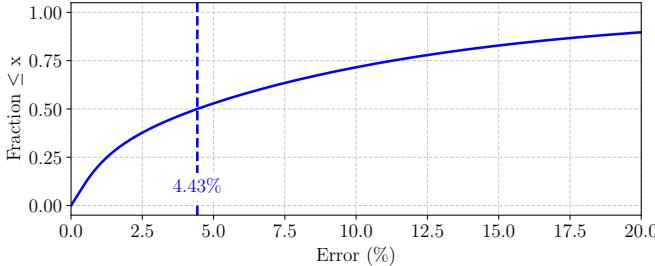


Fig. 11: eCDF of model performance on the test data, showing a median error of only 4.43 % across all coverage maps. Radio map values range from -150 dB to -50 dB.

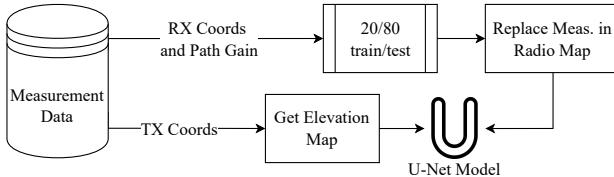


Fig. 12: Calibration pipeline using transfer learning with a weighted loss to align simulation with measurement data.

our U-Net model into higher-layer workflows for system-level evaluation. In this section, we demonstrate and evaluate how the U-Net serves as a channel-parameter generator for both Sionna SYS system simulations [19], and Colosseum—the world’s largest wireless network emulator [54].

A. Evaluation on Sionna SYS

Sionna SYS provides a modular framework to simulate full-stack system-level behavior. In each time slot, the system performs user scheduling across the resource grid, allocates transmit power to each user, computes the Signal to Interference plus Noise Ratio (SINR), selects the Modulation and Coding Scheme (MCS) via link adaptation, and generates decoded bits along with Hybrid Automatic Repeat reQuest (HARQ) feedback using physical layer abstraction.

In this framework, we assume perfect channel state information for precoder and equalizer design, achievable-rate estimation (for scheduling decisions), and channel-quality feedback (for link adaptation). We evaluate performance using

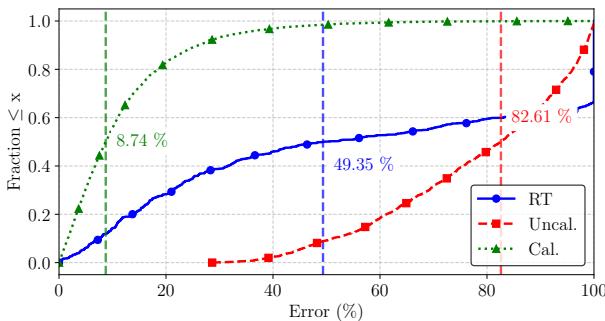


Fig. 13: eCDF of error with and without calibration, using only 20% of the measurement dataset for training.

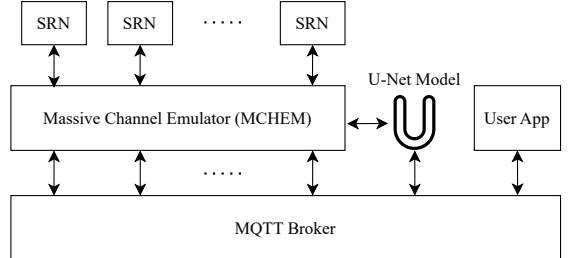
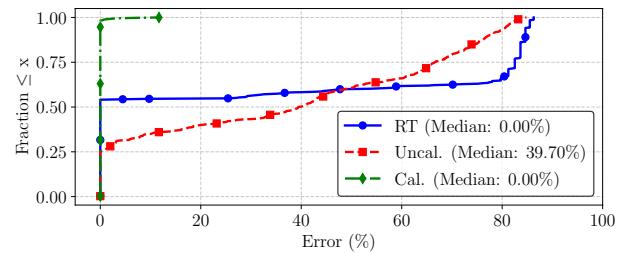
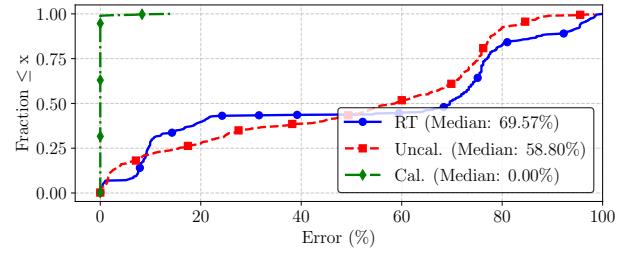


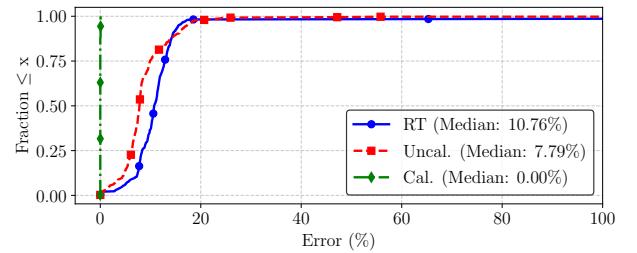
Fig. 14: Real-time channel emulation pipeline on Colosseum. User-selected locations trigger U-Net inference, with predicted taps sent via MQTT to Massive Channel Emulator (MCHEM) for emulation.



(a) ECDF of Shannon capacity error



(b) ECDF of link-adaptation efficiency error



(c) ECDF of BLER error

Fig. 15: Error distribution ECDFs comparing measurement data, Sionna RT, and both uncalibrated and calibrated U-Net models across (a) Shannon capacity, (b) link-adaptation efficiency, and (c) block-error rate.

both Shannon capacity and Outer Loop Link Adaptation (OLLA)-adjusted spectral efficiency, where OLLA dynamically selects the optimal MCS level.

We evaluate how path gain prediction errors impact system-level performance, focusing on spectral efficiency and Block Error Rate (BLER), as well as the feasibility of real-time simulation. For this analysis, we use the Sionna SYS simulator [19], which enables realistic wireless network simulations through

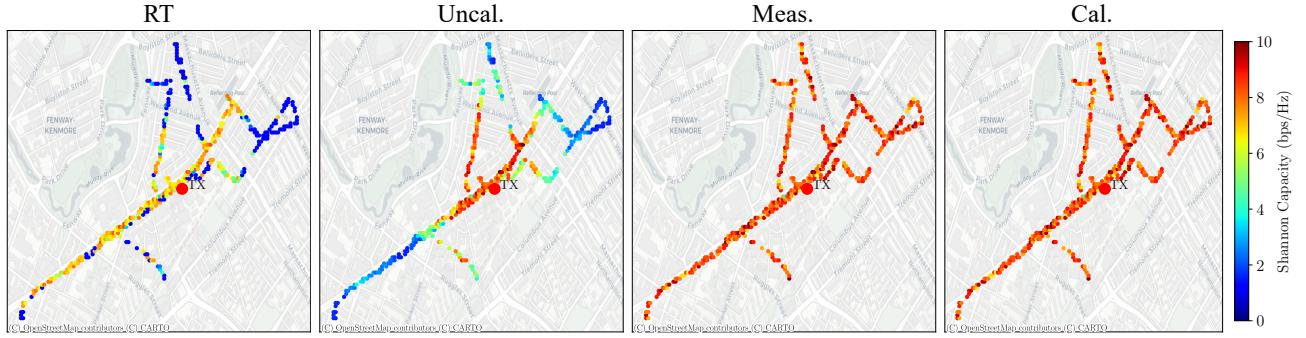


Fig. 16: Shannon capacity heatmaps for ray tracing, uncalibrated U-Net, measurement data, and calibrated U-Net model estimations on test measurement points.

a flexible and Python-based interface.

Fig. 15 compares errors for three channel-modeling approaches—Sionna RT 0.19.2, the uncalibrated U-Net, and the calibrated U-Net—against measurement data. Each model employs a single-tap channel with fixed propagation delay and randomized phase to focus exclusively on path-gain effects in system-level evaluations.

The results show that, despite a 10% error in path-gain estimates, the calibrated U-Net achieves virtually near-zero error across system-level metrics when compared to measurement-based channels. The uncalibrated U-Net also outperforms the ray-tracing baseline. These outcomes confirm that the U-Net’s residual path-gain deviations do not impact higher-layer simulations, validating its applicability for real-time system-level wireless emulation.

We implemented the full pipeline—from elevation map to system-level metrics—using Sionna SYS on an NVIDIA L40S GPU. After warm-up, each step is processed in 0.75 ms, and full radio map generation (200×200 points) takes 280 ms (284 ms including radio map processing), enabling real-time operation. The pipeline leverages TensorFlow XLA with mixed precision and batch parallelism, achieving 14,005 steps/sec/point for rapid adaptation in dynamic wireless environments.

B. Deployment on Colosseum

To enable real-time channel emulation as well as simulation using our U-Net model, we integrated it into the Colosseum platform. We followed the approach introduced in ColosSUMO [55], which extends Colosseum with real-time scenario generation capabilities. As shown in Fig. 14, when a user selects or updates a location, the corresponding building elevation map is loaded, transformed into the appropriate input shape and format within 20 ms, and passed to our U-Net model. The model performs inference in approximately 4 ms, producing a path gain prediction. This result is then sent to an MQTT broker, which forwards it to Colosseum’s MCHEM, the FPGA-based Massive Channel Emulator. MCHEM then uses these taps to emulate the corresponding wireless channel in real time.

In Fig. 17, we show results from running OpenAirInterface (OAI) on the Colosseum testbed using single-tap time-

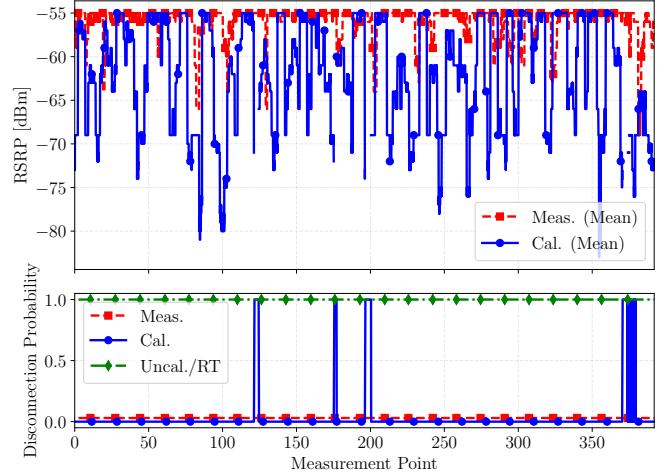


Fig. 17: Comparison of RSRP reports from OAI on Colosseum (10 runs per experiment). The Calibrated U-Net closely matches measurements, while Ray Tracing and Uncalibrated U-Net failed to establish a connection in this setup.

domain channels with simple propagation delays to simulate mobile scenarios across four channel models: Ray Tracing, Calibrated U-Net, Uncalibrated U-Net, and actual measurements. Each experiment was repeated 10 times. Both the Ray Tracing and Uncalibrated U-Net models failed to establish a connection under the same configuration, resulting in no Reference Signal Received Power (RSRP) data. In contrast, the Calibrated U-Net model successfully produced RSRP patterns closely matching empirical measurements, with an RMSE of 8.86 dBm.

VI. CONCLUSIONS

We have presented AIRMap, an AI-driven framework for real-time radio-map estimation that bridges the gap between high-fidelity ray tracing and fast, scalable inference. By relying solely on a novel single-channel elevation map—which is simple, cost-effective, and widely accessible—as input, and a U-Net-based autoencoder, AIRMap produces accurate path-gain maps in under 4 ms. Its data-driven design allows efficient transfer-learning calibration using minimal field measurements, enabling site-specific adaptation and correction of

residual errors. Extensive evaluation on a large Boston-area dataset shows AIRMap matches measurement-based performance on key system-level metrics, such as spectral efficiency and BLER. Integration into the Colosseum emulator and the Sionna SYS platform confirms AIRMap’s practical feasibility for real-time, end-to-end wireless network emulation.

A key feature of AIRMap is its resolution-adaptive input representation: while the model accepts fixed-size elevation map inputs, it supports variable spatial resolutions, enabling radio-map estimation across physical areas ranging from 500 m to 3 km per side without altering the architecture. This design supports flexible deployment scenarios—from dense urban zones to wider suburban or campus-scale regions—while preserving inference speed and model accuracy.

AIRMap’s contributions lie in drastically reducing computational cost and latency for radio environment modeling while using only easily obtainable elevation maps. Its data-driven nature allows for seamless incorporation of realistic wireless channel effects into real-time network management, adaptive resource allocation, and rapid scenario testing. These capabilities empower researchers and network operators to dynamically optimize system performance and accelerate the development of advanced digital twin applications requiring fast, accurate propagation predictions from minimal input data.

REFERENCES

- [1] A. Alkhateeb, S. Jiang, and G. Charan, “Real-Time Digital Twins: Vision and Research Directions for 6G and Beyond,” *IEEE Communications Magazine*, vol. 61, no. 11, pp. 128–134, Nov. 2023. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/10198573>
- [2] “Digital twins: what are they and how are they enabling future networks?” July 2023. [Online]. Available: <https://www.ericsson.com/en/blog/2022/3/what-are-digital-twins-three-real-world-examples>
- [3] J. Guo, T. Chen, S. Jin, G. Y. Li, X. Wang, and X. Hou, “Deep learning for joint channel estimation and feedback in massive MIMO systems,” *Digital Communications and Networks*, vol. 10, no. 1, pp. 83–93, Feb. 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S235286482300024X>
- [4] M. Alrabeiah and A. Alkhateeb, “Deep Learning for mmWave Beam and Blockage Prediction Using Sub-6 GHz Channels,” *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5504–5518, Sep. 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/9121328/>
- [5] M. Elsayed and M. Erol-Kantarci, “Radio Resource and Beam Management in 5G mmWave Using Clustering and Deep Reinforcement Learning,” in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, Dec. 2020, pp. 1–6, iSSN: 2576-6813. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9322401>
- [6] X. Li, X. Wang, K. Li, and V. C. M. Leung, “CaaS: Caching as a Service for 5G Networks,” *IEEE Access*, vol. 5, pp. 5982–5993, 2017. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/7890455>
- [7] C. Bell, “Digital twin ran: Key enablers.”
- [8] C. P. Robinson, A. Lacava, P. Johari, F. Cuomo, and T. Melodia, “TwiNet: Connecting Real World Networks to their Digital Twins Through a Live Bidirectional Link,” Nov. 2024, arXiv:2411.03503 [cs]. [Online]. Available: <http://arxiv.org/abs/2411.03503>
- [9] L. Bonati, S. D’Oro, M. Polese, S. Basagni, and T. Melodia, “Intelligence and learning in O-RAN for data-driven NextG cellular networks,” *arXiv:2012.01263 [cs.NI]*, pp. 1–7, December 2020.
- [10] S. D’Oro, M. Polese, L. Bonati, H. Cheng, and T. Melodia, “dApps: Distributed Applications for Real-Time Inference and Control in O-RAN,” *IEEE Communications Magazine*, vol. 60, no. 11, pp. 52–58, Nov. 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/9846950>
- [11] A. Lacava, L. Bonati, N. Mohamadi, R. Gangula, F. Kaltenberger, P. Johari, S. D’Oro, F. Cuomo, M. Polese, and T. Melodia, “dApps: Enabling Real-Time AI-Based Open RAN Control,” Jan. 2025, arXiv:2501.16502 [cs]. [Online]. Available: <http://arxiv.org/abs/2501.16502>
- [12] X. Zhao, F. Du, S. Geng, Z. Fu, Z. Wang, Y. Zhang, Z. Zhou, L. Zhang, and L. Yang, “Playback of 5g and beyond measured mimo channels by an ann-based modeling and simulation framework,” *IEEE journal on selected areas in communications*, vol. 38, no. 9, pp. 1945–1954, 2020.
- [13] 3GPP, “Study on channel model for frequencies from 0.5 to 100 GHz,” 3rd Generation Partnership Project (3GPP), Technical Report 38.901, 2020, version 16.1.0.
- [14] T. Zemen, J. Gomez-Ponce, A. Chandra, M. Walter, E. Aksoy, R. He, D. Matolak, M. Kim, J.-I. Takada, S. Salous, R. Valenzuela, and A. F. Molisch, “Site-Specific Radio Channel Representation for 5G and 6G,” *IEEE Communications Magazine*, vol. 63, no. 6, pp. 106–113, Jun. 2025. [Online]. Available: <https://ieeexplore.ieee.org/document/10742569/>
- [15] F. Fuschini, E. M. Vitucci, M. Barbirolli, G. Falciaiucca, and V. Degli-Esposti, “Ray tracing propagation modeling for future small-cell and indoor applications: A review of current techniques: RAY TRACING RADIO PROPAGATION MODELING,” *Radio Science*, vol. 50, no. 6, pp. 469–485, jun 2015. [Online]. Available: <http://doi.wiley.com/10.1002/2015RS00569>
- [16] S. M. Aldossari and K.-C. Chen, “Machine Learning for Wireless Communication Channel Modeling: An Overview,” *Wireless Personal Communications*, vol. 106, no. 1, pp. 41–70, may 2019. [Online]. Available: <https://doi.org/10.1007/s11277-019-06275-4>
- [17] J. Huang, C.-X. Wang, L. Bai, J. Sun, Y. Yang, J. Li, O. Tirkkonen, and M.-T. Zhou, “A Big Data Enabled Channel Model for 5G Wireless Communication Systems,” *IEEE Transactions on Big Data*, vol. 6, no. 2, pp. 211–222, jun 2020, conference Name: IEEE Transactions on Big Data. [Online]. Available: <https://ieeexplore.ieee.org/document/8561241/?arnumber=8561241>
- [18] Z. El-friakh, A. M. Voicu, S. Shabani, L. Simić, and P. Mähönen, “Crowdsourced Indoor Wi-Fi REMs: Does the Spatial Interpolation Method Matter?” in *2018 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, Oct. 2018, pp. 1–10, iSSN: 2334-3125. [Online]. Available: <https://ieeexplore.ieee.org/document/8610487>
- [19] J. Hoydis, S. Cammerer, F. Ait Aoudia, M. Nimier-David, L. Maggi, G. Marcus, A. Vem, and A. Keller, “Sionna,” 2022, <https://nvlabs.github.io/sionna/>.
- [20] VIAVI Solutions, “VIAVI Solutions - Enabling the Network of the Future,” <https://www.viavisolutions.com/en-us>.
- [21] P. Testolina, M. Polese, P. Johari, and T. Melodia, “BostonTwin: the Boston Digital Twin for Ray-Tracing in 6G Networks,” in *Proceedings of the ACM Multimedia Systems Conference 2024, ser. MMSys ’24, Bari, Italy*. Available online at: <https://arxiv.org/abs/2403.12289>, Apr. 2024.
- [22] O. Kanhere and T. S. Rappaport, “Calibration of nyuray, a 3d mmwave and sub-thz ray tracer using indoor, outdoor, and factory channel measurements,” *arXiv preprint arXiv:2302.12380*, 2023.
- [23] J. Hoydis, F. A. Aoudia, S. Cammerer, F. Euchner, M. Nimier-David, S. t. Brink, and A. Keller, “Learning radio environments by differentiable ray tracing,” *arXiv preprint arXiv:2311.18558*, 2023.
- [24] J. Jemai, P. C. Eggers, G. F. Pedersen, and T. Kurner, “Calibration of a uwb sub-band channel model using simulated annealing,” *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 10, pp. 3439–3443, 2009.
- [25] M. Deruyck, W. Joseph, and L. Martens, “Power consumption model for macrocell and microcell base stations,” *Transactions on Emerging Telecommunications Technologies*, vol. 25, no. 3, pp. 320–333, 2014, _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/ett.2565>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.2565>
- [26] T. Imai, K. Kitao, and M. Inomata, “Radio propagation prediction model using convolutional neural networks by deep learning,” in *2019 13th European Conference on Antennas and Propagation (EuCAP)*. IEEE, 2019, pp. 1–5.
- [27] S. P. Sotiroudis, S. K. Goudos, and K. Siakavara, “Feature importances: A tool to explain radio propagation and reduce model complexity,” in *Telecom*, vol. 1, no. 2. MDPI, 2020, p. 9.
- [28] T. Hayashi, T. Nagao, and S. Ito, “A study on the variety and size of input data for radio propagation prediction using a deep neural network,” in *2020 14th European Conference on Antennas and Propagation (EuCAP)*. IEEE, 2020, pp. 1–5.
- [29] J. Thrane, D. Zibar, and H. L. Christiansen, “Model-aided deep learning method for path loss prediction in mobile communication systems at 2.6 ghz,” *IEEE Access*, vol. 8, pp. 7925–7936, 2020.
- [30] J. Thrane, B. Sliwa, C. Wietfeld, and H. L. Christiansen, “Deep learning-based signal strength prediction using geographical images and expert knowledge,” in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.

- [31] T. T. Nguyen, R. Caromi, K. Kallas, and M. R. Souryal, "Deep learning for path loss prediction in the 3.5 ghz cbrs spectrum band," in *2022 IEEE wireless communications and networking conference (WCNC)*. IEEE, 2022, pp. 1665–1670.
- [32] K. Inoue, K. Ichige, T. Nagao, and T. Hayashi, "Learning-based prediction method for radio wave propagation using images of building maps," *IEEE Antennas and Wireless Propagation Letters*, vol. 21, no. 1, pp. 124–128, 2021.
- [33] A. Gupta, J. Du, D. Chizhik, R. A. Valenzuela, and M. Sellathurai, "Machine learning-based urban canyon path loss prediction using 28 ghz manhattan measurements," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 6, pp. 4096–4111, 2022.
- [34] O. Ozegen, S. Mohammadjafari, M. Cevik, K. El Mokhtari, J. Ethier, and A. Basar, "An empirical study on using cnns for fast radio signal prediction," *SN Computer Science*, vol. 3, no. 2, p. 131, 2022.
- [35] V. V. Ratnam, H. Chen, S. Pawar, B. Zhang, C. J. Zhang, Y.-J. Kim, S. Lee, M. Cho, and S.-R. Yoon, "Fadenet: Deep learning-based mm-wave large-scale channel fading prediction and its applications," *IEEE Access*, vol. 9, pp. 3278–3290, 2020.
- [36] R. Levie, Ç. Yasar, G. Kutyniok, and G. Caire, "Radiounet: Fast radio map estimation with convolutional neural networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 4001–4015, 2021.
- [37] S. Bakirtzis, J. Chen, K. Qiu, J. Zhang, and I. Wassell, "Em deepray: an expedient, generalizable, and realistic data-driven indoor propagation model," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 6, pp. 4140–4154, 2022.
- [38] J.-H. Lee, O. G. Serbetci, D. P. Selvam, and A. F. Molisch, "Pmnet: Robust pathloss map prediction via supervised learning," *arXiv preprint arXiv:2211.10527*, 2022.
- [39] K. Qiu, S. Bakirtzis, H. Song, J. Zhang, and I. Wassell, "Pseudo ray-tracing: Deep leaning assisted outdoor mm-wave path loss prediction," *IEEE Wireless Communications Letters*, vol. 11, no. 8, pp. 1699–1702, 2022.
- [40] C. A. Oroza, Z. Zhang, T. Watteyne, and S. D. Glaser, "A machine-learning-based connectivity model for complex terrain large-scale low-power wireless deployments," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 576–584, 2017.
- [41] Y. Zhang, J. Wen, G. Yang, Z. He, and X. Luo, "Air-to-air path loss prediction based on machine learning methods in urban environments," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.
- [42] G. Yang, Y. Zhang, Z. He, J. Wen, Z. Ji, and Y. Li, "Machine-learning-based prediction methods for path loss and delay spread in air-to-ground millimetre-wave channels," *IET Microwaves, Antennas & Propagation*, vol. 13, no. 8, pp. 1113–1121, 2019.
- [43] S. P. Sotirovdis, S. K. Goudos, and K. Siakavara, "Deep learning for radio propagation: Using image-driven regression to estimate path loss in urban areas," *ICT Express*, vol. 6, no. 3, pp. 160–165, 2020.
- [44] N. Sun, S. Geng, S. Li, X. Zhao, M. Wang, and S. Sun, "Channel modeling by rbf neural networks for 5g mm-wave communication," in *2018 IEEE/CIC International Conference on Communications in China (ICCC)*. IEEE, 2018, pp. 768–772.
- [45] X. Zhao, F. Du, S. Geng, N. Sun, Y. Zhang, Z. Fu, and G. Wang, "Neural network and gbsm based time-varying and stochastic channel modeling for 5g millimeter wave communications," *China Communications*, vol. 16, no. 6, pp. 80–90, 2019.
- [46] J. Huang, C.-X. Wang, L. Bai, J. Sun, Y. Yang, J. Li, O. Tirkkonen, and M.-T. Zhou, "A big data enabled channel model for 5g wireless communication systems," *IEEE Transactions on Big Data*, vol. 6, no. 2, pp. 211–222, 2018.
- [47] H. Zhang, J. Dong, X. Liu, J. Liu, and X. Zhang, "An artificial intelligence radio propagation model based on geographical information," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 12, pp. 12 049–12 060, 2022.
- [48] A. Seretis and C. D. Sarris, "An overview of machine learning techniques for radiowave propagation modeling," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 6, pp. 3970–3985, 2021.
- [49] C. Huang, R. He, B. Ai, A. F. Molisch, B. K. Lau, K. Haneda, B. Liu, C.-X. Wang, M. Yang, C. Oestges *et al.*, "Artificial intelligence enabled radio propagation for communications—part ii: Scenario identification and channel modeling," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 6, pp. 3955–3969, 2022.
- [50] Y. Wen, W. Hu, S. Geng, and X. Zhao, "Machine learning based mm-wave 60 ghz channel modeling for 5g wireless communication systems," in *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*. IEEE, 2019, pp. 1005–1010.
- [51] A. Saeizadeh, M. Tehrani-Moayyed, D. Villa, J. G. Beattie, I. C. Wong, P. Johari, E. W. Anderson, S. Basagni, and T. Melodia, "AI-Assisted Agile Propagation Modeling for Real-Time Digital Twin Wireless Networks," in *2024 IEEE 29th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2024, pp. 1–6.
- [52] J.-H. Lee and A. F. Molisch, "A scalable and generalizable pathloss map prediction," *IEEE Transactions on Wireless Communications*, vol. 23, no. 11, pp. 17 793–17 806, 2024.
- [53] S. Bakirtzis, J. Chen, K. Qiu, J. Zhang, and I. Wassell, "Em deepray: an expedient, generalizable, and realistic data-driven indoor propagation model," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 6, pp. 4140–4154, 2022.
- [54] M. Polese, L. Bonati, S. D'Oro, P. Johari, D. Villa, S. Velumani, R. Gangula, M. Tsampazi, C. P. Robinson, G. Gemmi, A. Lacava, S. Maxenti, H. Cheng, and T. Melodia, "Colosseum: The Open RAN Digital Twin," *IEEE Open Journal of the Communications Society*, pp. 1–13, August 2024.
- [55] G. Gemmi, P. Johari, P. Casari, M. Polese, T. Melodia, and M. Segata, "ColosSUMO: Evaluating Cooperative Driving Applications with Colosseum," in *2024 IEEE Vehicular Networking Conference (VNC)*, May 2024, pp. 97–100, iSSN: 2157-9865. [Online]. Available: <https://ieeexplore.ieee.org/document/10576000/>