A Neural Approach to Ray Tracing for Realistic Wireless Channel Simulation in Indoor and Urban Scenarios

Francisco Javier Somolinos-Simón, Adina Murg, Hanli Liu,
Carlos J. Hellín, Josefa Gómez and Abdelhamid Tayebi
Department of Computer Science
University of Alcalá
Alcalá de Henares, Spain
e-mail: {francisco.somolinos | adina.murg | hanli.liu
| carlos.hellin | josefa.gomezp | hamid.tayebi}@uah.es

***Abstract*—Accurate modeling of wireless channels is essential for the design and optimization of next generation communication networks such as 6G. Traditional ray tracing techniques provide physically consistent simulations but suffer from high computational complexity, limiting their scalability and real-time applicability. This work proposes a neural network based surrogate model for ray tracing in complex 3D environments. This approach leverages multilayer perceptrons to predict the interaction of electromagnetic rays with surfaces, estimating critical channel parameters such as gain, time-of-flight, and propagation angles. The model is trained and validated using datasets generated by the Sionna ray tracing engine in both indoor and large urban scenarios. Results demonstrate that the neural surrogate achieves low prediction errors in key metrics and generalizes well across different environments. This neural ray tracing framework offers a scalable, flexible, and efficient alternative to conventional physics based simulators.**

***Keywords*-neural networks; ray tracing; MIMO systems.**

I. Introduction

In recent years, many 6G network research topics have required the simulation of specific radio environments using ray tracing. This requirement arises from the need for a spatially consistent correspondence between a physical location in a scene and the impulsive channel response, a feature not readily provided by widely used stochastic channel models. The main challenges for the design, deployment and optimization of wireless communication networks are based on understanding and accurately modeling the characteristics of the real-time propagation channel, allowing fast and accurate simulations of complex scenarios such as massive Multiple-Input Multiple-Output (MIMO) systems and the deployment of digital twins, among others [1].

The physics of such ElectroMagnetic (EM) wave propagation between a transmit and receive point are analytically given by the Maxwell equations: the transmitted wave undergoes different interactions with the environment (e.g., reflection), and the receiver gets the wave through multiple paths with different times-of-flight and powers, and from different directions. However, solving the Maxwell equations with boundary conditions requires in-depth knowledge of the propagation environment, therefore, classically modeling EM propagation is intractable for most engineering applications [2].

Ray tracing-based simulators are commonly employed for modeling wireless channel properties [3]–[6]. In the ray tracing process, electromagnetic rays are uniformly launched from the transmitter antenna, undergoing reflections, transmissions, and diffractions with various buildings and floors, ultimately reaching the receiver locations. These ray paths and interactions yield valuable wireless channel information, such as channel gain, channel transfer function, and channel impulse response [7].

While ray tracing has been a popular tool in wireless channel modeling, its computational complexity escalates with the number of ray-object interactions. To address these needs, neural network based forward surrogate models emerge as an attractive solution.

A neural network is a mechanism that takes inputs and learns associations to predict some outputs [8]. Artificial Neural Network (ANN) models are gaining importance in the field of predictive modeling because of their capability to model nonlinear relationships in a high-dimensional dataset. ANN models can predict a complex relationship between variables, which is not otherwise possible with other models such as logistic regression models [9].

ANN models work on the principles of biological neural networks containing nodes (analogous to cell bodies) that communicate with other nodes through connections (analogous to axons and dendrites) [10]. An ANN consists of an input layer, hidden layers and an output layer while the information is fed into the model through the input layer, processed through the hidden layers and put out from the output layer [9] (Figure 1).

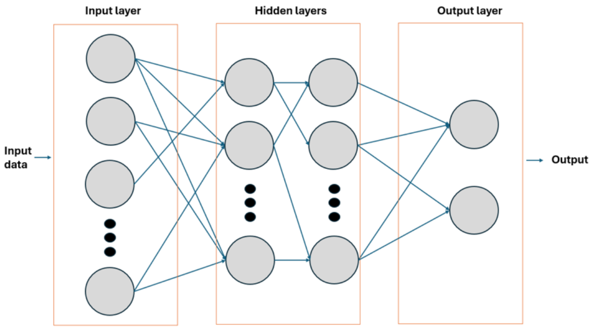

Figure 1. Structure of a neural network showing input features, hidden layers for pattern learning, and output neurons representing predicted quantities.

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-292-0

Among the main types of neural network models generally used in predicting, Multi-Layer Perceptron (MLP) stands out. It is one of the feedforward networks where the input values are multiplied with their corresponding weights and fed into the hidden layer while the hidden layer process transfers the weighted input to the output layer with values of multiplied weights corresponding to the output layer. MLP uses the backpropagation algorithm to recalculate the weights [9]. It uses commonly conventional activation functions such as the Sigmoid function or the Tanh function, however, any non-linear and continuous function, such as Rectified Linear Units (ReLU), is suitable for use in an MLP [11].

The major highlights of this model are as follows [12]:

- The neural network contains one or more intermediate layers between the input and the output nodes, which are hidden from both input and output nodes.
- Each neuron in the network includes a non-linear activation function that is differentiable.
- The neurons in each layer are connected with some or all the neurons in the previous layer.

Then, the objective of this paper is to train neural networks to predict interactions between wireless beams and objects in a three-dimensional environment. This approach involves simulating ray-surface interactions to estimate transmission-reception paths, considering characteristics such as time of flight and gain. The proposed neural network learns how surfaces impact wireless ray propagation, predicting factors such as attenuation and direction of the outgoing ray based on attributes of the incident ray. This approach offers the advantage of applying to new scenarios, improving its versatility to adapt to different situations.

Compared to existing neural ray tracing methods in the literature, the proposed work excels in scalability and flexibility, accommodating diverse levels of geometric complexity while maintaining high-quality channel prediction. This neural ray tracing framework is validated across indoor and outdoor scenes and shows potential for real-time or large-scale deployment, particularly in the context of next generation 6G communication systems and digital twin technologies.

The remainder of this paper is organized as follows: Section II gives a short overview of the work related to the idea to be put forward. Section III details how datasets are generated using the Sionna ray tracing engine in both indoor and outdoor 3D scenes. Section IV introduces the architecture of the proposed model. Section V explains the model training process. Section VI presents three metrics: Overall Error, Geometry Error, and Average Delay Mean Absolute Error (MAE), to assess how accurately the model predicts the physical behavior of wireless paths compared to ground-truth data. Section VII presents the results of the proposed neural ray tracing model in various scenarios and evaluates its performance using the defined metrics. Finally, Section VIII summarizes the main contributions of the paper and outlines potential directions for future research.

II. Related Work

Physically based simulation guided by neural networks is gaining popularity across various scientific domains. In the field of applied and computational electromagnetics, several approaches leveraging neural networks have been proposed to accelerate or approximate ray-based simulations [13]–[15]. For instance, Jin et al. [15] redefine ray trajectory generation as a sequential decision making problem, introducing the SANDWICH framework, a fully differentiable, scene aware neural architecture that jointly learns optical, physical, and signal properties of the environment.

On the other hand, other approaches from the domain of neural rendering and computer graphics also employ neural networks to model ray tracing and light transport. Knodt et al. [16] explicitly model light transport between scene surfaces using disentangled neural representations of geometry and reflectance, allowing for efficient inverse rendering. Zeng et al. [17] propose MirrorNeRF, a neural rendering framework capable of learning accurate geometry and mirror reflection, supporting scene manipulations such as adding new objects or modifying reflective surfaces and synthesizing corresponding reflections.

Many of these neural surrogates aim to learn the scattering process involving obstacles in free space.

A recent work addressing this task is WiNeRT [2]. In the authors' approach, a neural surrogate to model wireless electromagnetic propagation effects in indoor environments is implemented. Such neural surrogates provide a fast, differentiable, and continuous representation of the environment and enable end-to-end optimization for downstream tasks. Specifically, they render the wireless signal (e.g., time of flight, power of each path) in an environment as a function of the sensor's spatial configuration (e.g., placement of transmit and receive antennas). That is to say, their approach inscribes within the ray tracing channel modeling paradigm, where wireless propagation is precisely modeled by tracing wireless rays.

Another work in this area is RayProNet [7]. The authors introduce a novel machine learning-empowered methodology for wireless channel modeling. The key ingredients include a point-cloud-based neural network and a Spherical Harmonics encoder with light probes. Their approach offers the flexibility to adjust antenna radiation patterns and transmitter/receiver locations, the capability to predict radio path loss maps, and the scalability of large-scale wireless scenes. This work is validated in various outdoor and indoor radio environments.

Additionally, widely adopted datasets such as DeepMIMO [18] support the training and benchmarking of data-driven channel models in ray-traced environments, and are instrumental in standardizing the evaluation of neural surrogates. More recently, physics-informed learning techniques have been proposed to embed propagation physics into deep models, improving generalizability and interpretability in scenario aware channel modeling [19].

III. Data Collection

The datasets in this project are generated using an open-source ray tracing simulator: Sionna [20]. Sionna Ray Tracing (RT) is a ray tracing extension for radio propagation modeling that is built on top of Mitsuba 3 [21] and TensorFlow [22]. Sionna RT relies on Mitsuba 3 for the rendering and scene handling, e.g., its XML-file format, as well as the computation of ray intersections with scene primitives, i.e., triangles forming a mesh modeling a surface [23]. Scene files for Mitsuba 3 may be created, edited, and exported using the popular open-source 3D content creation suite Blender [24] and the Mitsuba-Blender add-on. The dataset is generated in various scenes such as a cube (small indoor room scene) (Figure 2), and Munich (large urban city scene) (Figure 3).

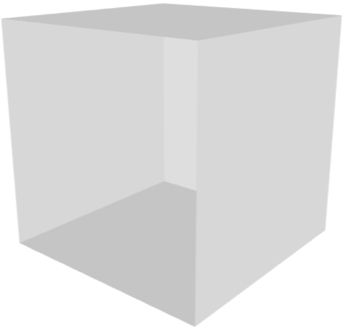

Figure 2. Cube: small indoor room scene which contains marble material.

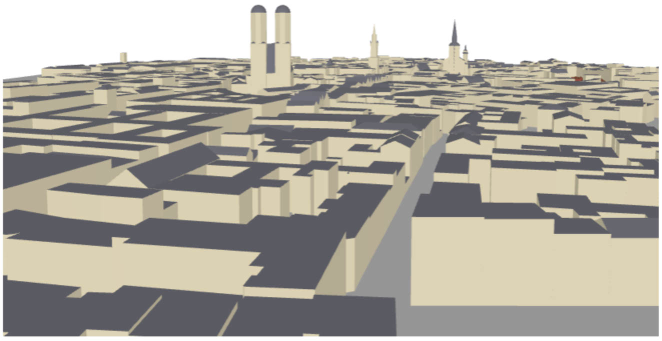

Figure 3. Munich: large urban city scene with a rich material composition, including both marble and metal.

The first scene contains marble material consistent with its simple indoor layout. In contrast, the second scene features a richer material composition, including both marble and metal, to better reflect the diversity of an urban landscape [25]. Table I shows the properties of these materials, as relative permittivity (ε_r) and conductivity (σ).

TABLE I. Constituent material properties (relative permittivity ε_r and conductivity σ)

Material name	ε_r	σ [S/m]
Marble	7.074	0.018
Metal	1	10^7

As described in Table II, datasets comprise 20 transmitter locations with Third Generation Partnership Project (3GPP) TR 38.901 pattern [26] and 40 sampled receiver locations with short dipole pattern with linear polarization pattern for each scene. To position the transmitters and receivers in the scene, a random uniform sampling strategy was adopted. Specifically, antenna positions were sampled uniformly in the plane within the bounding box of the scene, centered around the scene's geometric center. A fixed random seed was used to ensure reproducibility. This uniform spatial sampling avoids positional bias and ensures that the antennas are evenly distributed across the area of interest (671 m^2 for small indoor room scene and 4,082,653 m^2 for large urban city). The operating frequency is 3.5 GHz. Reflection is activated whereas diffraction and scattering are not activated.

TABLE II. Data collection: configuration details of datasets

Dataset	Cube	Munich
Scale	Small indoor room scene	Large urban city
Covered area (m^2)	671	4,082,653
Transmitters	20	20
Receivers	40	40
Antenna pattern	Transmitter: 3GPP TR 38.901[26] Receiver: short dipole pattern with linear polarization	Transmitter: 3GPP TR 38.901[26] Receiver: short dipole pattern with linear polarization
Frequency (GHz)	3.5	3.5
Number of bounces	3	3
Number of rays to trace	10^6	10^6
Reflected paths are computed	True	True
Diffracted paths are computed	False	False
Scattered paths are computed	False	False

IV. Neural Model

The general goal of the project is to contribute to the advancement of next generation networks by designing, developing, and testing an innovative neural network based ray tracer. The model takes three configuration parameters as input: a 3D representation of the environment and the spatial coordinates of the transmitter and receiver devices. The model predicts the wireless scene where the output is a variably-sized set of K paths. A path consists of a sequence of ray segments (r, r+1 ...) connecting a transmitter to a receiver. Each path encodes three channel attributes: gain (a_k), time-of-flight (τ_k) and angles (Φ_k). This approach allows for the effective encoding of interacting objects, with a particular emphasis on learning geometric features. Therefore, the final state is modeled as an evaluation of the interactions the ray experiences with its environment which is represented as a 3D mesh composed of F faces and V vertices, where each face corresponds to some surface on a wall.

An MLP model is built to predict the transformation to the incident ray (the new direction ($d_k^{(r+1)}$) and gain ($a_k^{(r+1)}$)). The network predicts an attenuation factor s and a rotation matrix A (4-dim Euler-Rodrigues parameterization), which is then used to determine the updated gain and direction.

Specifically, this neural model consists of two MLP networks, an MLP spatial network with 2 hidden layers, each with 64 hidden units and ReLU activation to encode EM properties specific to a spatial region, but independent of the incidence direction and an MLP directional network with 1 hidden layer with 64 hidden units and ReLU activation which predicts the rotation a ray incident taking into account the direction.

The first network takes as inputs: f_i, a one-hot encoded identifier of the face where the relay point $x_k^{(r+1)}$ lies; n_i, the surface normal vector at that face, representing the geometric orientation of the surface; and, b_i, a 3D conditioning vector based on signed distances (sdf). These signed distance functions measure how far a given coordinate (e.g., the transmitter, receiver, or relay point) is from the face f_i, taking into account whether the point is inside or outside the face. This helps the network contextualize the interaction based on geometry.

The second network takes as input the spacial encoding v_i (output of the first network) and incorporates the direction of incidence $d_k^{(r)}$ (the direction of the incoming ray) to model how the ray interacts with the surface, including how much of it is reflected or transmitted, and how its direction changes.

The final output is scaling and additive coefficients s for the gain magnitude ($a_k^{(r+1)} = s_1 a_k^{(r)} + s_2$) and 4-dim parameters for rotation (based on Euler-Rodrigues formulation). The rotation parameters ρ_i are mapped to a 3×3 rotation matrix A to transform the incident to outgoing ray $d_k^{(r+1)} = A d_k^{(r)}$.

The new angles and time-of-flight have been calculated from the new direction. The new angle $\Phi_k^{(r+1)}$ describes the horizontal direction in which the ray travels after the interaction, measured in the XY plane, then it has been calculated using the arctangent function which gives the angle between the Y and X components of the direction vector. The new time-of-flight $\tau_k^{(r+1)}$ represents the time it takes for the ray to travel from the current point of interaction to the next point where it hits an object or reaches the receiver. It is a measure of propagation delay. To calculate it, the path of the ray is simulated using the Mitsuba renderer. For each predicted outgoing direction: a new ray is created starting from the interaction point; this ray is traced through the 3D scene using Mitsuba's *ray_intersect* method; if the ray intersects with an object, the distance along the ray to that point is recorded; and, this distance is then divided by the speed of light (3x10^8 m/s) to convert it into time. Only rays that intersect with valid surfaces are used; rays that go off to infinity are ignored.

V. Training

In this section, the implementation details are introduced in the training settings of this project.

In these experiments, K rays are initially launched omnidirectionally from the transmitter location, agnostic to the environment and location of the receiver location. For each ray, its interaction with the environment is evaluated. These data are separated into training and validation sets. Among them, about 85% are used for training, with the remaining 15% reserved for validation.

The MLP architecture models were coded in Python 3 (v3.10.6), using the PyTorch framework (v2.7.0), while Mitsuba 3 (v3.5.2) was employed for physically based rendering and ray intersection computations. Supporting tools include NumPy (v1.23.5) and SciPy (v1.15.2) (Rotation module) for quaternion operations.

The models were trained and tested on a laptop computer with a GPU environment NVIDIA GeForce RTX 2070 and 8 GB of RAM. Each of these models is trained in a supervised setting for 100 epochs with a learning rate of 0.001 and batch size of 1. Adam optimizer and the Mean Square Error (MSE) loss function for scalar-valued attributes and cosine distances loss function between angular attributes are utilized for received path loss optimization in this project. Set based Channel Loss compares two sets of multi-path channels: the predicted set and the ground-truth set. This comparison provides feedback to improve model training. To compare two paths the difference between each pair is measured. For scalar attributes, the differences directly are calculated whereas for angular attributes, the angular difference by treating the angles as unit vectors in Cartesian coordinates is measured and using a cosine-based distance.

This approach ensures that the loss considers both the matching of paths and the accuracy of their attributes, leading to meaningful guidance for training the model.

Training time for the model was approximately 4.70 seconds per epoch for indoor scenes and 0.03 seconds per epoch for outdoor scenes on an NVIDIA RTX 2070 GPU. The computational complexity scales linearly with the number of rays and interactions due to the feedforward architecture of the MLP. This efficiency enables the training of larger models or generalization to new environments using standard computational resources.

The inputs include normalized geometric features such as surface normals and signed distance functions, while face identifiers are one-hot encoded. The model architecture, described in Section IV, comprises two MLPs with ReLU activations, trained jointly to predict path direction and gain.

VI. Evaluation Metric

The evaluation metric serves as a quantitative measure to assess the performance of the proposed method in the prediction. In this work, absolute error metrics are used rather than relative errors, because some channel parameters, such as gain and delay, can be close to zero, which would make relative errors unstable or less meaningful. Therefore, absolute errors provide a more reliable and interpretable assessment of the accuracy of the prediction. Three evaluation metrics are considered to evaluate this approach:

- Overall prediction error ('Overall'): This metric measures how well the entire set of predicted paths matches the set of ground-truth paths. To compare the two sets of paths (predicted vs. ground truth), correspondences between them are established using a linear sum assignment problem (also known as the Hungarian algorithm). This algorithm finds the best one-to-one matching between predicted and ground-truth paths that minimizes the total error. The final error considers all relevant path attributes, including gain, angles and time-of-flight. A lower value indicates better overall alignment between predicted and true multipath components.

$$\text{Err}_{\text{overall}} = \frac{1}{N}\sum_{k=1}^{N}\left(|a_k - \hat{a}_k| + |\tau_k - \hat{\tau}_k| + |\Phi_k - \hat{\Phi}_k|\right)$$

  where:
  - N is the total number of predicted paths.
  - a_k and $\hat{a}_k$ are the true and predicted gain of path k, respectively.
  - τ_k and $\hat{\tau}_k$ are the true and predicted time-of-flight of path k.
  - Φ_k and $\hat{\Phi}_k$ are the true and predicted angles of path k.
- Geometry prediction error ('Geometry'): This is a more focused version of the overall error. It still uses the matching mechanism from metric, but instead of evaluating all attributes, it specifically looks at two that describe the geometry of the path: angles and time-of-flight. This metric evaluates whether the predicted rays follow the same geometric routes as the ground truth, meaning they bounce off the same surfaces and follow similar trajectories between the transmitter and receiver. As with the overall error, lower values indicate better geometric consistency.

$$\text{Err}_{\text{geometry}} = \frac{1}{N}\sum_{k=1}^{N}\left(|\tau_k - \hat{\tau}_k| + |\Phi_k - \hat{\Phi}_k|\right)$$

  where:
  - N is the total number of predicted paths.
  - τ_k and $\hat{\tau}_k$ are the true and predicted time-of-flight of path k.
  - Φ_k and $\hat{\Phi}_k$ are the true and predicted angles of path k.
- Average Delay Time - MAE ('AvgDelay'): This metric summarizes the average time delay (τ) of all the predicted paths in a channel and compares it to the average delay of the ground-truth paths. For each path, its average time-of-flight is calculated and weighted by the linear power of the path. Then, the Mean Absolute Error (MAE) between the predicted average delay and the true average delay is computed. Lower values here indicate that the temporal structure of the predicted channel closely matches the true one.

$$\text{Err}_{\text{avg\_delay}} = \left|\frac{\sum_{k=1}^{N} a_k \cdot \tau_k}{\sum_{k=1}^{N} a_k} - \frac{\sum_{k=1}^{N} \hat{a}_k \cdot \hat{\tau}_k}{\sum_{k=1}^{N} \hat{a}_k}\right|$$

  where:
  - N is the total number of predicted paths.
  - a_k and $\hat{a}_k$ are the true and predicted linear gains of path k.
  - τ_k and $\hat{\tau}_k$ are the true and predicted time-of-flight of path k.

VII. Experimental Results

This section presents the performance evaluation of the proposed neural network based ray tracing model through experiments conducted in both indoor and outdoor environments. The model's ability is evaluated to predict key wireless propagation characteristics, including path gain, direction or time-of-flight, using the datasets generated with the Sionna ray tracing simulator. Performance metrics, as described in previous section, are used to evaluate the model's accuracy, generalization, and ability to learn complex interactions in realistic 3D scenarios.

Table III presents the quantitative evaluation of the model's performance in two distinct scenarios: a small indoor room and a large urban city environment. The results are reported across three metrics: Overall, Geometry and AvgDelay.

TABLE III. Quantitative Results. Comparing errors of this approach in two different scenarios

Metrics	Small indoor room scene	Large urban city
Overall	2.374163	2.313505
Geometry	2.372437	2.312989
AvgDelay	0.001727	0.000516

The Overall error reflects the model's ability to predict complete ray paths accurately. The values are quite similar across both scenes (2.37 for the indoor scene vs. 2.31 for the city), indicating consistent overall performance regardless of scene complexity.

The Geometry error focuses specifically on the accuracy of the predicted ray geometry. Again, the model shows similar performance in both environments (2.37 vs. 2.31), suggesting that it effectively captures the geometric characteristics of the propagation paths.

The AvgDelay error, measured as MAE, shows a greater difference between scenes. The model achieves better delay prediction in the large urban city (0.000516) compared to the indoor room (0.001727). This may be attributed to the richer variety of multipath effects in urban settings, which enhance the model's ability to learn delay patterns effectively.

When compared to WiNeRT [2], the proposed model demonstrates competitive performance. Although WiNeRT achieves a superior geometry error of 0.084 in controlled indoor settings, the AvgDelay error of this model (0.0005 in urban environments) is significantly lower than WiNeRT's best-case error of 0.828. These results underscore the model's strong predictive accuracy, particularly in complex outdoor scenarios with greater environmental variability.

A time performance evaluation is also proposed comparing this model to traditional ray tracing (Table IV).

TABLE IV. RUNTIME COMPARISON BETWEEN THE PROPOSED MODEL AND SIONNA RAY TRACING SIMULATOR

Dataset	**Small indoor room scene**	**Large urban city**
Runtime (neural model)	470.07 s	2.97 s
Runtime (Sionna ray tracing)	1.07 s	14.39 s

The runtime comparison between the proposed neural model and the traditional Sionna ray tracing simulator reveals interesting behavior across different environments. As shown in Table IV, the neural model achieves significantly faster inference in the large urban city scenario (2.97 seconds vs. 14.39 seconds for Sionna), demonstrating its potential for efficient large-scale deployment. However, in the small indoor room, the neural model exhibits a notably higher runtime (470.07 seconds compared to 1.07 seconds with Sionna). This disparity is likely due to the higher density of multipath reflections in indoor environments, increasing the computational load for the neural network.

When compared to RayProNet [7], which leverages continuous neural point-field representations for efficient runtime performance, the proposed model exhibits a more scene dependent behavior. RayProNet achieves inference times under 100s for complex indoor and outdoor environments, while the proposed model excels in outdoor scenarios with sparse geometries.

Overall, the proposed neural network based model demonstrates robust generalization across diverse environments, with notable strengths in predicting propagation delays and efficient runtime performance in large-scale urban settings. These findings suggest that the model is well-suited for applications requiring high accuracy and scalability in wireless environments.

VIII. CONCLUSION AND FUTURE WORK

In this work, a neural network based surrogate model for ray tracing in wireless communication environments has been presented. By learning how electromagnetic rays interact with 3D surfaces, this proposed model effectively predicts critical channel attributes such as gain, angle of departure/arrival, and time-of-flight. This approach has been validated using the Sionna ray tracing simulator in both indoor and outdoor settings, demonstrating consistent performance across different levels of scene complexity. Notably, this model shows strong generalization capabilities and achieves low error in average delay prediction, especially in urban environments where multipath effects are more diverse.

These findings suggest that neural ray tracing offers a scalable and efficient alternative to traditional physics-based simulators, with the potential for real-time or large-scale deployment in the context of 6G and digital twin technologies.

Despite the promising results, this study has several limitations that should be addressed in future work: simplified material properties, the materials used in the dataset have fixed relative permittivity and conductivity values. While this ensures consistency, it does not capture the variability found in real-world materials; limited scene diversity, only two scenes were considered for dataset generation. These environments, while representative of some scenarios, do not encompass the full range of conditions encountered in practical wireless communication system; exclusion of diffraction and scattering, the dataset was generated without accounting for diffraction and scattering effects. These phenomena, however, can significantly influence propagation characteristics, particularly in environments with sharp edges or complex surfaces; and, fixed frequency, all experiments were conducted at a single operating frequency of 3.5 GHz.

Future work could explore the scalability to larger and more diverse scenes, extend to additional propagation phenomena, and use different frequency ranges to further enhance the robustness and applicability of the model. In particular, future extensions of this work could target environments such as multi-floor indoor settings and dense urban areas with complex obstructions. The inclusion of additional physical effects like diffraction and scattering would improve modeling in scenarios with sharp edges and rough surfaces. Evaluating the model at other frequency bands, especially sub-THz ranges relevant to 6G, would further validate its generalizability. Moreover, integrating neural surrogates with physics-based modules could help balance efficiency with physical interpretability. Finally, online learning or reinforcement learning approaches could enable the model to adapt in real time within digital twin applications.

ACKNOWLEDGMENT

This work was supported by the program "Programa de Ayudas para la Realización de Proyectos de Investigación UAH" of the Vice-Rectorate for Research and Knowledge Transfer of the University of Alcala (Spain) through project PIUAH24/IA-076, and by the contract Programa Investigo ref. 04-UAH-INV2024 provided by the Comunidad de Madrid.

REFERENCES

[1] T. K. Sarkar, *The physics and mathematics of electromagnetic wave propagation in cellular wireless communication*, 1st edition. 2018, ISBN: 9781119393139. [Online]. Available: https://onlinelibrary.wiley.com/doi/book/10.1002/9781119393146.

[2] T. Orekondy *et al.*, "WineRT: Towards neural ray tracing for wireless channel modelling and differentiable simulations," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=tPKKXeW33YU.

[3] F. Aguado Agelet, A. Formella, J. Hernando Rabanos, F. Isasi de Vicente, and F. Perez Fontan, "Efficient ray-tracing acceleration techniques for radio propagation modeling," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 6, pp. 2089–2104, 2000. DOI: 10.1109/25.901880.

[4] Z. Ji, B.-H. Li, H.-X. Wang, H.-Y. Chen, and T. Sarkar, "Efficient ray-tracing methods for propagation prediction for indoor wireless communications," *IEEE Antennas and Propagation Magazine*, vol. 43, no. 2, pp. 41–49, 2001. DOI: 10.1109/74.924603.

[5] D. He *et al.*, "The design and applications of high-performance ray-tracing simulation platform for 5g and beyond wireless communications: A tutorial," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 10–27, 2019. DOI: 10.1109/COMST.2018.2865724.

[6] F. Saez de Adana, O. Gutierrez Blanco, I. Gonzalez Diego, J. Perez Arriaga, and M. Catedra, "Propagation model based on ray tracing for the design of personal communication systems in indoor environments," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 6, pp. 2105–2112, 2000. DOI: 10.1109/25.901882.

[7] G. Cao and Z. Peng, "Raypronet: A neural point field framework for radio propagation modeling in 3d environments," *IEEE Journal on Multiscale and Multiphysics Computational Techniques*, vol. PP, pp. 1–12, Jan. 2024. DOI: 10.1109/JMMCT.2024.3464373.

[8] N. Purkait, *Hands-on neural networks with Keras : design and create neural networks using deep learning and artificial intelligence principles*, 1st edition. 2019, ISBN: 9781789533347. [Online]. Available: https://github.com/PacktPublishing/Hands-On-Neural-Networks-with-Keras?tab=readme-ov-file.

[9] V. Renganathan, "Overview of artificial neural network models in the biomedical domain," vol. 120, pp. 536–540, Jul. 2019. DOI: 10.4149/BLL_2019_087.

[10] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun, *Doctor ai: Predicting clinical events via recurrent neural networks*, 2016. arXiv: 1511.05942 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1511.05942.

[11] A. Nguyen, K. Pham, D. Ngo, T. Ngo, and L. Pham, "An analysis of state-of-the-art activation functions for supervised deep neural network," *CoRR*, vol. abs/2104.02523, 2021. arXiv: 2104.02523. [Online]. Available: https://arxiv.org/abs/2104.02523.

[12] S. Dutt, *Machine learning*, [First edition]. 2019, ISBN: 9789353067373. [Online]. Available: https://www.oreilly.com/library/view/machine-learning/9789389588132/.

[13] Y. Ge, L. Guo, and M. Li, "Physics-informed deep learning for time-domain electromagnetic radiation problem," in *2022 IEEE MTT-S International Microwave Biomedical Conference (IMBioC)*, 2022, pp. 114–116. DOI: 10.1109/IMBioC52515.2022.9790302.

[14] L. Li *et al.*, "Deepnis: Deep neural network for nonlinear electromagnetic inverse scattering," *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 3, pp. 1819–1825, 2019. DOI: 10.1109/TAP.2018.2885437.

[15] Y. Jin *et al.*, *Sandwich: Towards an offline, differentiable, fully-trainable wireless neural ray-tracing surrogate*, 2025. arXiv: 2411.08767 [cs.NI]. [Online]. Available: https://arxiv.org/abs/2411.08767.

[16] J. Knodt, S. Baek, and F. Heide, "Neural ray-tracing: Learning surfaces and reflectance for relighting and view synthesis," *CoRR*, vol. abs/2104.13562, 2021. arXiv: 2104.13562. [Online]. Available: https://arxiv.org/abs/2104.13562.

[17] J. Zeng *et al.*, "Mirror-nerf: Learning neural radiance fields for mirrors with whitted-style ray tracing," in *Proceedings of the 31st ACM International Conference on Multimedia*, ser. MM '23, ACM, Oct. 2023, pp. 4606–4615. DOI: 10.1145/3581783.3611857. [Online]. Available: http://dx.doi.org/10.1145/3581783.3611857.

[18] A. Alkhateeb, "Deepmimo: A generic deep learning dataset for millimeter wave and massive MIMO applications," *CoRR*, vol. abs/1902.06435, 2019. arXiv: 1902.06435. [Online]. Available: http://arxiv.org/abs/1902.06435.

[19] E. Zhu, H. Sun, and M. Ji, *Physics-informed generalizable wireless channel modeling with segmentation and deep learning: Fundamentals, methodologies, and challenges*, 2024. arXiv: 2401.01288 [cs.IT]. [Online]. Available: https://arxiv.org/abs/2401.01288.

[20] J. Hoydis *et al.*, *Sionna*, version 1.1.0, https://nvlabs.github.io/sionna/, 2022.

[21] W. Jakob *et al.*, *Mitsuba 3 renderer*, version 3.1.1, https://mitsuba-renderer.org, 2022.

[22] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *CoRR*, vol. abs/1603.04467, 2016. arXiv: 1603.04467. [Online]. Available: http://arxiv.org/abs/1603.04467.

[23] J. Hoydis *et al.*, "Sionna RT: Differentiable Ray Tracing for Radio Propagation Modeling," *arXiv preprint*, Mar. 2023.

[24] Blender Online Community, *Blender - a 3d modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org.

[25] Recommendation ITU-R, *Effects of building materials and structures on radiowave propagation above about 100 mhz p series radiowave propagation*, 2015. [Online]. Available: https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.2040-1-201507-S!!PDF-E.pdf.

[26] ETSI TR 138 901, *Study on channel model for frequencies from 0.5 to 100 ghz (3gpp tr 38.901 version 16.1.0 release 16)*, 2020. [Online]. Available: https://www.etsi.org/deliver/etsi_tr/138900_138999/138901/16.01.00_60/tr_138901v160100p.pdf.