Digital Object Identifier

WiPy-RT: A Fast Ray Tracing Modeling Platform for RIS-assisted Wireless Channels

MOHAMMADREZA FARASHAHI[1], (Member, IEEE), BOON-CHONG SEET[1], (Senior Member, IEEE) and Xue Jun Li.[1], (Senior Member, IEEE)

[1]Department of Electrical and Electronic Engineering, Auckland University of Technology, Auckland, New Zealand.

Corresponding author: Mohammadreza Farashahi (e-mail: reza.farashahi@aut.ac.nz).

ABSTRACT

Recent advancements in wireless technologies, particularly in the context of the sixth generation (6G) mobile communications and Internet of Things (IoT) systems, have introduced a wide range of requirements and challenges in wireless communication. These developments necessitate comprehensive channel information, encompassing the three-dimensional (3D) features of electromagnetic (EM) signals. Such signals are now typically transmitted by antenna arrays, manipulated by Reconfigurable Intelligent Surface (RIS) structures, and received by another set of antenna arrays. This complex propagation environment demands sophisticated modeling techniques to accurately capture and predict channel behavior, essential for the design and optimization of next-generation wireless systems.

Ray tracing (RT) based simulators have gained significant traction in recent years, proving their worth in accurately and efficiently simulating EM propagation environments. These simulators have demonstrated remarkable accuracy in modeling complex wireless scenarios, making them invaluable tools for researchers and engineers. However, the intricate interactions between EM waves and the physical environment in three-dimensional space can still render the simulation process time-consuming, especially for scenarios with high complexity. Fortunately, recent state-of-the-art developments in Graphics Processing Unit (GPU) and Central Processing Unit (CPU) technologies have substantially mitigated this challenge. These hardware advancements have dramatically reduced the computational overhead associated with complex environmental simulations, making it feasible to conduct comprehensive and realistic RT-based analyses of sophisticated wireless environments.

This paper introduces WiPy-RT, a high-performance ray tracing simulator designed for modeling RIS-enabled wireless environments. WiPy-RT features an interactive interface that leverages highly configurable and open-source Python libraries for 3D structures, including PyVista, Trimesh, and PyTorch3D. This GPU-accelerated simulator excels in simulating first-order diffraction and multi-reflection (up to 6 orders of reflection) ray tracing scenarios, while being capable of modeling RIS behavior with large-scale unit cell arrays. Benchmarking against the state-of-the-art WiThRay simulator, WiPy-RT demonstrated an average simulation time of 0.8 seconds, compared to WiThRay's 16 minutes, achieving over 98% reduction in simulation time. Additionally, WiPy-RT provided higher resolution and accuracy in power distribution modeling for complex 3D urban environments. WiPy-RT introduces several novel features, including an intelligent ray launching method using Gaussian Kernel Density Estimation (KDE) and an efficient RIS modeling approach employing Random Forest Regression. Comparative analysis shows that WiPy-RT achieves up to 1000x speedup over existing solutions while maintaining high accuracy. WiPy-RT's fast execution and ability to handle complex 3D environments make it particularly suitable for advanced wireless communication research, including the simulation of massive MIMO systems and intelligent reflecting surfaces. The interdisciplinary nature of WiPy-RT, combining techniques from machine learning and computational electromagnetics, represents a significant step forward in addressing the challenges of modeling next-generation wireless systems.

INDEX TERMS

channel modeling, channel impulse response, ray tracing, reconfigurable intelligent surface, smart wireless environments

I. INTRODUCTION

THE development of beyond fifth-generation (5G) wireless networks marked a significant leap forward in connectivity, with ongoing advancements pushing the boundaries of what's possible in wireless communications [1]. With 5G technology becoming matured, the research community is already conceptualizing and developing sixth-generation (6G) systems, envisioning use cases such as holographic communications, extended reality (XR), and ultra-massive Internet of Things (IoT) [2]. The IoT landscape is growing exponentially, with recent projections suggesting nearly 7 billion connected devices by 2030, over 97% of which being Massive IoT and Broadband IoT [3]. This rapid proliferation of IoT devices, coupled with emerging technologies like Smart Cities Autonomous Vehicles, and Industry 4.0, is driving the need for more sophisticated wireless infrastructure [4]. To meet these demands, 6G systems have been anticipated to operate at higher frequency bands, including terahertz frequencies, utilize advanced technologies like RIS structures, and leverage artificial intelligence for network optimization [5].These advancements necessitate more accurate and efficient channel modeling techniques to capture the complex interactions in increasingly dense and dynamic wireless environments [6].

As wireless communications move towards higher frequency bands to meet increasing capacity demands, signal propagation faces significant challenges due to higher path loss [7]. In this high-frequency regime, beamforming techniques become crucial for focusing energy towards users and overcoming propagation losses [8]. Consequently, the accurate modeling of wireless systems and optimal equipment placement requires a shift from traditional stochastic models to more precise geometrical channel models that can better describe the complex wireless environment [9]. Ray tracing (RT) emerges as a particularly suitable candidate for this purpose, offering a deterministic approach that can accurately capture the intricate interactions between EM waves and the physical environment [10]. This method allows for detailed modeling of reflection, diffraction, and scattering phenomena, providing a comprehensive representation of the wireless channel [11]. Moreover, RT can incorporate the effects of beamforming and reconfigurable intelligent surfaces (RIS), making it invaluable for designing and optimizing next-generation wireless networks [12].

RT has emerged as a powerful technique for site-specific EM propagation modeling in wireless communications environments [13]. This method, based on geometrical optics and uniform theory of diffraction, allows for accurate prediction of radio wave propagation in complex urban, suburban, and indoor scenarios [14]. RT algorithms identify possible paths between transmitter and receiver, accounting for mechanisms such as direct line-of-sight propagation, specular reflections from surfaces, diffractions around edges, and transmissions through walls [15]. All RT-based propagation methods fundamentally adhere to Fermat's principle, which states that the path taken by a ray between two points is the one that can be traversed in the least time. This principle governs the behavior of rays as they interact with the environment, determining the angles of reflection and refraction, and the paths of diffracted rays [16]. By applying Fermat's principle, RT methods can efficiently compute the most significant propagation paths, providing a physically accurate representation of wave propagation in complex environments [11].

RT algorithms for propagation modeling generally fall into two primary categories: the Direct method (also known as Ray Launching) and the Inverse method [14], [17]. The Direct method involves launching a large number of rays from the transmitter in predefined directions, typically with a constant angular separation, and tracing their paths through the environment [18]. This method is computationally efficient, but may miss important paths for receivers far from the transmitter due to the discrete nature of the launched rays [19]. The Inverse method, on the other hand, determines all possible paths between a specific transmitter-receiver pair by considering all potential interactions with the environment [20]. While more computationally intensive, this method ensures that all significant paths are found, providing a more accurate representation of the channel, especially for long-distance propagation [21].

In [22], an efficient ray-tracing based channel model for wireless sensor networks in smart city environments was presented. The model uses two approaches: a horizontal method based on an optimized visibility graph, and a vertical over-rooftop technique for distant sensors. Validated against real-world measurements, the model demonstrates improved accuracy over statistical models when integrated into a Wireless Sensor Network (WSN) simulator, better estimating network performance in realistic environments.

Ziemann et al. introduced in a convolutional neural network approach for predicting radio frequency RT solutions in urban environments [23]. The authors generated a large synthetic dataset of urban scenes and corresponding RT simulations to train their model. By using a stepped prediction method and dilated convolutions, they were able to accurately predict propagation patterns for up to two reflections, though accuracy decreased for higher-order reflections.

In [24], Choi et al. presented WiThRay, an open-source 3D ray-tracing channel simulator for modeling smart wireless environments, including systems with reconfigurable intelligent surfaces. The authors introduced an efficient "bypassing on edge" algorithm and dynamic scattering area adjustment to accurately model complex propagation scenarios. This versatile simulator enables researchers to realistically evaluate various 5G/6G wireless techniques in advanced environments, particularly those involving RIS technology.

The authors in [25] investigated the design and applications of high-performance ray-tracing (RT) simulation platforms for 5G and beyond wireless communications. The authors provided a comprehensive tutorial reviewing state-of-the-art RT techniques, comparing existing simulators, and discussing requirements for supporting 5G scenarios. The work also introduced a cloud-based RT platform called CloudRT, demonstrating its applications through urban channel modeling and

MIMO beamforming case studies.

Lecci et al. in [26] explored simplification techniques for RT simulations of millimeter wave channels, aiming to improve the trade-off between accuracy and computational complexity. The authors proposed limiting the maximum number of reflections and discarding weak multipath components based on a relative power threshold. Their results showed that these simplifications can significantly reduce simulation time with minimal impact on accuracy in certain scenarios, identifying optimal configurations that balance performance and computational efficiency.

Authors in [27] developed a technique to integrate RIS functionality into a 3D RT simulator for indoor wireless communications. The authors studied RIS deployment strategies and channel characteristics in two non-line-of-sight indoor scenarios at 5.4 GHz and 28 GHz. Their results showed substantial coverage improvements with RIS and indicated that placing RIS in the middle of the wall facing both transmitter and receiver generally provides optimal performance when all are at the same height.

Liu and Sarris proposed in [28] a hybrid numerical method combining full-wave analysis with ray-tracing to efficiently model propagation in RIS-enabled communication channels. The authors used full-wave simulation to obtain the radiation pattern of the transmitter and complex radar cross section of the RIS, which was then imported into a ray-tracer as a secondary transmitter. The proposed technique was validated against full-wave finite element results in representative indoor propagation scenarios, demonstrating good accuracy while significantly reducing computational time compared to full finite element analysis.

A RT-based channel modeling method in [29] was used to simulate RIS assisted MIMO systems for 6G communications. The authors divided multipath components based on RIS interaction, implemented segmented RT simulations, and modeled RIS directional reflection as a secondary transmitter. This approach was validated against indoor measurements at 5.4 GHz, demonstrating good accuracy in predicting various channel characteristics for RIS-assisted scenarios.

Authors in [30] proposed to model wave propagation in RIS-assisted communication channels, combining full-wave analysis with RT. The authors used the equivalence principle to represent the RIS as a surface current source, which was then integrated into RT simulations. This approach demonstrated accuracy comparable to full-wave analysis in both near-field and far-field regions of the RIS, while maintaining computational efficiency for modeling realistic RIS-enabled environments.

While significant advancements have been made in RT-based propagation modeling for wireless communications, the incorporation of RIS technology into these models is still in its infancy. There exists a substantial research gap in accurately and efficiently modeling the behavior of RIS within complex wireless systems using RT techniques. The unique properties of RIS, including their ability to dynamically alter the electromagnetic environment, present novel challenges for traditional RT approaches. Furthermore, as wireless environments become increasingly complex, particularly with the integration of RIS, there is a pressing need for RT tools that can maintain high accuracy while also offering computational efficiency. This balance is important for handling the intricate scenarios expected in future wireless networks, especially in urban and dense IoT environments.

The key technical challenges in RIS-assisted RT modeling include: (1) efficiently handling the complex electromagnetic interactions between incident waves and numerous RIS unit cells, (2) accurately representing the phase-shifting behavior of RIS structures without prohibitive computational overhead, and (3) developing algorithms that maintain precision while dramatically reducing simulation time. These challenges are particularly acute in applications requiring rapid assessment of multiple RIS configurations or large-scale simulation environments.

Addressing these challenges would have significant practical impact across multiple domains. For network design, faster RIS-assisted RT simulation would enable rapid prototyping and optimization of network topologies. For device manufacturers, it would accelerate the development cycle of RIS hardware by providing quick feedback on different unit cell configurations. In real-world applications, such efficient modeling tools would facilitate the deployment planning of RIS-enhanced networks in smart cities, industrial IoT environments, and high-density urban areas where coverage optimization is critical.

In this paper, we propose WiPy-RT, an advanced Pythonic wireless propagation simulator with ray tracing. WiPy-RT addresses the aforementioned challenges by offering a novel approach to RT simulation that combines high accuracy with computational efficiency, particularly in modeling RIS-enabled environments. Our simulator leverages state-of-the-art machine learning techniques and efficient algorithms to overcome the limitations of traditional RT methods, especially when dealing with complex, RIS-enhanced wireless scenarios.

WiPy-RT is developed entirely in Python, leveraging efficient and versatile open-source libraries such as PyVista, Trimesh, PyTorch3D, and Scikit-learn. This choice of implementation not only ensures wide accessibility and ease of use for researchers but also allows for seamless integration with other Python-based tools commonly used in wireless communications research. At its core, WiPy-RT employs an advanced ray launching method, ensuring swift and efficient channel model realization for complex indoor and outdoor smart wireless environments. The simulator is capable of identifying direct, multi-order reflected, and one-order diffracted EM waves, providing a comprehensive representation of the wireless channel. For point-to-point channel modeling, WiPy-RT also incorporates the inverse method, offering flexibility in simulation approaches.

The key contributions of this paper are:

- Addressing the computational inefficiency gap in existing simulators through an intelligent ray launching

method strategy with Gaussian Kernel Density Estimation (KDE) ray emission model. This novel approach significantly improves the efficiency of ray tracing by adaptively focusing computational resources on the most relevant propagation paths. Compared to traditional uniform ray launching methods, our KDE-based approach achieves up to 40% reduction in simulation time while maintaining comparable accuracy.

- Solving the RIS unit cell interaction modeling challenge through an efficient method for RIS modeling using Random Forest Regression and Inverse RT for estimating travel distance between TX and RIS' unit cells. This hybrid approach allows for accurate modeling of RIS behavior without the computational overhead typically associated with full-wave analysis of large RIS structures. Our method demonstrates a 60% improvement in computation time for RIS modeling compared to conventional ray tracing techniques, while achieving less than 5% error in path loss estimation.
- Overcoming the platform accessibility limitations of existing solutions by implementing a comprehensive RT modeling platform in Python that outperforms existing solutions in terms of speed and scalability. WiPy-RT achieves simulation times that are up to 100 times faster than comparable MATLAB-based simulators for complex urban environments, making it particularly suitable for large-scale studies of 5G and 6G wireless networks.
- Demonstrating the effectiveness of WiPy-RT through extensive benchmarking against existing simulators and validation with real-world measurements. Our results show that WiPy-RT achieves an average accuracy improvement of 15% in predicting received signal strength in RIS-enabled environments, compared to traditional ray tracing methods.

These contributions collectively address the growing need for efficient and accurate modeling tools in the evolving landscape of wireless communications, particularly in the context of RIS-enabled 5G and 6G networks. By offering a unique combination of machine learning techniques, efficient algorithms, and a user-friendly Python implementation, WiPy-RT represents a significant advancement in the field of wireless channel modeling.

The remainder of this paper is organized as follows: Section II investigates the theoretical foundations underlying RT-based EM propagation modeling, elucidating the key principles and methodologies. A comprehensive discussion of the proposed WiPy-RT, including its architecture, algorithms, and novel features, is presented in Section III. Section IV offers a comparative analysis, presenting results that benchmark WiPy-RT against WiThRay, a comprehensive ray tracing propagation simulator, highlighting its performance and accuracy. Section V discusses the WiPy-RT's approach to modeling RIS behavior in the SISO system. Finally, Section VI concludes the paper, summarizing the key findings and discussing potential future research directions in this field.

II. THEORY BEHIND RAY TRACING

This section provides the foundational electromagnetic theory and ray tracing methodologies that underpin WiPy-RT's capabilities. Understanding these principles is helpful for appreciating the technical challenges WiPy-RT addresses in RIS-assisted wireless channel modeling. The section begins by examining the electromagnetic basis of ray tracing, derived from high-frequency approximations of Maxwell's equations, which forms the theoretical foundation for our intelligent ray launching method using Gaussian KDE. We then explore different ray types (direct, reflected, and diffracted), which sets a foundation for comprehending how WiPy-RT accurately models complex propagation environments where RIS elements interact with electromagnetic waves. Finally, we review direct and inverse ray tracing engines, highlighting their respective strengths and limitations that motivated our hybrid approach combining Random Forest Regression with traditional RT techniques. By establishing this theoretical groundwork, we provide context for WiPy-RT's key innovations: its adaptive ray distribution model that focuses computational resources on relevant propagation paths, and its efficient RIS modeling approach that dramatically reduces simulation time while maintaining high accuracy.

A. RAY TRACING AND ELECTROMAGNETICS

The ray concept in electromagnetics emerges from the high-frequency approximation of Maxwell's equations. This approximation is valid when the wavelength λ is much smaller than the characteristic dimensions of the problem geometry. In this regime, the electric and magnetic fields can be expressed as [31]:

$$\begin{aligned} \vec{E}(\vec{r}) &= \vec{e}(\vec{r})e^{-j\beta_0 S(\vec{r})} \\ \vec{H}(\vec{r}) &= \vec{h}(\vec{r})e^{-j\beta_0 S(\vec{r})}, \end{aligned} \tag{1}$$

where $\vec{E}(\vec{r})$ and $\vec{H}(\vec{r})$ are the vector electric and magnetic fields, respectively. $\vec{e}(\vec{r})$ and $\vec{h}(\vec{r})$ are slowly varying vector amplitude functions, $\beta_0 = \frac{2\pi}{\lambda}$ is the free-space wavenumber, $S(\vec{r})$ is the eikonal function, representing the optical path length, and $\vec{r}$ is the position vector in 3D space.

The exponential term $\exp(-j\beta_0 S(\vec{r}))$ describes the rapid phase variations of the wave, while $\vec{e}(\vec{r})$ and $\vec{h}(\vec{r})$ account for slower amplitude and polarization changes. Substituting these expressions into Maxwell's equations and considering the limit as $\beta_0 \to \infty$ leads to the eikonal equation [32]:

$$|\nabla S|^2 = n^2, \tag{2}$$

where n is the refractive index of the medium and can be calculated with $n = \sqrt{\mu_r \epsilon_r}$. This equation governs the behavior of the wavefronts (surfaces of constant phase) and, consequently, the ray trajectories, which are orthogonal to these wavefronts. The ray paths can be determined by solving the differential equation [32]:

$$\frac{d}{ds}\left(n\frac{d\vec{r}}{ds}\right) = \nabla n, \tag{3}$$

where s is the arc length along the ray path. In a homogeneous medium, where the permittivity ϵ, permeability μ, and refractive index n are constant throughout the space, the gradient of the refractive index vanishes ($\nabla n = 0$). Consequently, the general ray trajectory equation simplifies to [32]:

$$\frac{d^2\vec{r}}{ds^2} = 0 \tag{4}$$

The second-order differential equation derived has a general solution of the form [32]:

$$\vec{r} = \vec{a}s + \vec{b}, \tag{5}$$

where $\vec{r}$ is the position vector along the ray path, and $\vec{a}$ and $\vec{b}$ are constant vectors. This solution describes a straight line in the homogeneous medium, with $\vec{a}$ representing the direction of propagation and $\vec{b}$ providing an initial position.

The ray concept, as formulated above, provides a mathematical foundation for Fermat's principle of least time. This principle states that light travels between two points along the path that requires the least time. From this fundamental principle, one can derive the laws of reflection, refraction, and diffraction.

The high-frequency approximation of Maxwell's equations, as formulated above, allows us to model EM wave propagation using the concept of rays. This approach is particularly useful in complex environments where the wavelength is smaller than the characteristic dimensions of objects in the scene.

B. RAY TYPES

EM field distribution can be effectively described by considering three primary types of rays: direct rays, representing LoS propagation between the source and receiver, reflected rays, accounting for interactions with surfaces, where the incident wave is redirected according to the law of reflection, and diffracted rays, which describes the phenomenon of waves bending around obstacles or passing through apertures, extending the field into shadow regions.

1) Direct rays

Direct or LoS rays represent the simplest form of propagation, where EM waves travel directly from the transmitter to the receiver without encountering any obstacles. In an ideal free-space environment, the path loss for LoS propagation can be calculated using the Friis transmission equation:

$$L_{\text{LoS}} = 20\log_{10}\left(\frac{4\pi d}{\lambda}\right) \tag{6}$$

where L_{LoS} is the path loss in dB, d is the distance between the transmitter and receiver, and λ is the wavelength of the signal. This equation assumes isotropic antennas and no atmospheric effects.

2) Reflected Rays

Reflection occurs when EM waves encounter a smooth surface that is large compared to the wavelength. The behavior of reflected rays follows the law of reflection, where the angle of incidence equals the angle of reflection. The amplitude of the reflected wave depends on the reflection coefficient, which varies with the material properties and the polarization of the incident wave.

For a single reflection, the electric field of the reflected wave can be expressed as:

$$E_r = E_i \cdot R \cdot \exp(-jkr) \tag{7}$$

where, E_r is the reflected field, E_i is the incident field, R is the reflection coefficient, k is the wavenumber $\left(\frac{2\pi}{\lambda}\right)$, and r is the total path length. The reflection coefficient R differs for perpendicular and parallel polarizations:

$$\begin{aligned} R_\perp &= \frac{\eta_2\cos(\theta_1) - \eta_1\cos(\theta_2)}{\eta_2\cos(\theta_1) + \eta_1\cos(\theta_2)} \\ R_\parallel &= \frac{\eta_2\cos(\theta_2) - \eta_1\cos(\theta_1)}{\eta_2\cos(\theta_2) + \eta_1\cos(\theta_1)} \end{aligned} \tag{8}$$

where η_1 and η_2 are the intrinsic impedances of the two media, and θ_1 and θ_2 are the angles of incidence and refraction, respectively.

3) Diffracted Rays

Diffraction occurs when EM waves encounter edges, corners, or small openings relative to the wavelength. This phenomenon allows waves to propagate into shadowed regions behind obstacles. To incorporate diffraction effects into ray-based models, two primary theories have been developed: the Geometrical Theory of Diffraction (GTD) and the Uniform Theory of Diffraction (UTD).

The Geometrical Theory of Diffraction, introduced by J.B. Keller [16], extends geometrical optics to include diffracted rays. GTD posits that when an incident ray strikes an edge or vertex, it produces diffracted rays emanating from these points. While GTD improved field predictions in shadow regions, it has a limitation of predicting infinite field strengths at shadow and reflection boundaries.

The Uniform Theory of Diffraction, developed by Kouyoumjian and Pathak [33], addresses GTD's limitations. UTD introduces transition functions that ensure the total field remains finite and continuous everywhere, including at shadow and reflection boundaries. This makes UTD particularly valuable for accurate field predictions across all regions of space.

The general form of the edge diffracted electric field is given by:

$$E^d = E^i(Q_E) \cdot \overline{D} \cdot A(s)\exp(-jks) \tag{9}$$

This formula encapsulates the key components of the diffracted field. $E^i(Q_E)$ represents the incident field at the diffraction point Q_E, which serves as the source of the diffracted ray. $\overline{D}$ is the dyadic diffraction coefficient, which

describes how the edge diffracts the incident field. As elaborated in [2], $\overline{D}$ can vary depending on the specific type of edge encountered. $A(s)$ is a spreading factor that accounts for the amplitude variation along the diffracted ray path. The exponential term $\exp(-jks)$ represents the phase change as the wave propagates, where k is the wavenumber and s is the distance from the diffraction point.

C. RT ENGINES

At the core of RT engines, there are two fundamental methods for tracing rays from the transmitter to the receiver: the direct method and the inverse method. These approaches form the basis for how ray paths are calculated in radio propagation modeling. Regardless of the method used, it is of great importance to take into account a key feature for all rays: from point A to point B, rays take the least amount of time possible to travel, following Fermat's principle. In the following, direct and inverse RT methods will be investigated in greater detail.

1) Direct RT

The direct method, also known as the Shooting and Bouncing Ray (SBR) method, is a fundamental approach in ray tracing for radio propagation modeling. In this method, a multitude of rays are emitted from the transmitter in various directions, sampling the 3D environment. The number and distribution of these rays as well as their separation angle are determined by the engine's configuration to ensure adequate coverage of the propagation space. As rays propagate through the environment, they interact with geometrical objects present in the 3D model, including reflection, diffraction, and scattering. For each ray's interaction and traveling distance, a loss coefficient is calculated. This coefficient accounts for the attenuation of the signal along its path and can be determined using appropriate propagation formulas, such as those discussed earlier in this paper.

The final step in the SBR method involves identifying and analyzing the rays that reach the vicinity of the receiver. These rays are considered valid propagation paths and form the basis for characterizing the radio channel. Each received ray carries important information: its delay, which represents the time taken to travel from the transmitter to the receiver; its total travel distance; its power level at the receiver, calculated by considering the initial transmission power and the cumulative losses incurred along its path; and its angle of arrival, which is crucial for spatial channel characterization. This angle of arrival information is particularly important for technologies such as smart antennas and MIMO systems. This collection of ray data provides a comprehensive multipath profile of the propagation environment, enabling detailed analysis of signal characteristics such as delay spread, angular spread, and received signal strength at the receiver location.It is worth noting that with this method, the computational overhead increases linearly with the number of reflections allowed in the engine, which can be a significant consideration in complex environments or when high accuracy is required.

2) Inverse RT

The inverse method, also known as the image method or ray backtracing, approaches the problem of ray tracing from a different perspective compared to the direct method. Instead of launching rays from the transmitter, this method starts at the receiver and works backwards to determine possible paths to the transmitter.

The process begins by considering the receiver location and identifying potential reflection points in the environment. For each reflective surface, an image of the transmitter is created by mirroring its position across the surface. These image points represent potential sources from which rays could have originated to reach the receiver after a single reflection.

The method then proceeds to check if there is a clear, unobstructed path between the receiver and each image point. If such a path exists, it represents a valid ray path from the transmitter to the receiver via that reflection point. This process is repeated for higher-order reflections, creating images to account for multiple reflections.

For diffraction, the method identifies potential diffraction points, such as building edges or corners, and determines if there are valid paths from the transmitter to the diffraction point and from the diffraction point to the receiver. The same principle applies to other propagation mechanisms like scattering, where potential scattering points are identified and analyzed. Each valid path discovered through this process is then characterized in terms of its total path length, delay, angle of arrival at the receiver, and signal strength. The signal strength is calculated by considering the transmission power and the losses incurred due to propagation distance and interactions with the environment.

One of the key advantages of the inverse method is its efficiency in scenarios with a limited number of reflections and a complex receiver environment. It can be particularly effective when the number of receivers is small compared to the number of transmitters or when analyzing specific receiver locations in detail.

However, it is important to note that the computational complexity of the inverse method increases exponentially with the number of reflections considered, as the number of image points grows rapidly. This can make it less suitable for environments requiring many reflections or for large-scale simulations with numerous receivers.

III. PROPOSED RAY TRACER

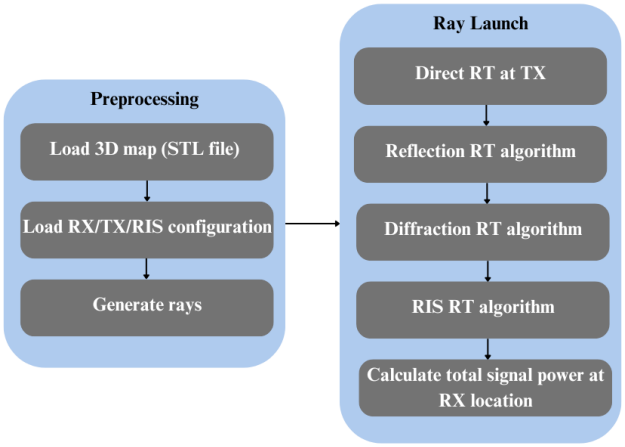


FIGURE 1: Overview of proposed WiPy-RT's architecture.

This section delves into the intricacies of WiPy-RT's architecture and functionality, investigating step-by-step its approach to modeling complex EM propagation scenarios. The following subsections provide a comprehensive overview of the simulator's core components, algorithms, and methodologies. The discussion begins with an examination of the preprocessing stage, where the environment is imported and structured, key features are extracted, and the wireless system is defined. Next, the ray launching process, the heart of the simulator, is explored, detailing how rays are tracked, labeled, and organized as they propagate through the environment. Finally, the RIS implementation stage will be discussed, where unique rays between TX-RIS and RIS-RX are efficiently calculated. A general overview of the WiPy-RT's architecture is shown in Figure 1

A. PREPROCESSING

The preprocessing stage in WiPy-RT is a critical phase that sets the foundation for accurate and efficient ray tracing simulations. This stage begins with the loading of environment data, typically in the form of STL formatted files, which contain the 3D geometric information of the simulation environment. These files are then imported and processed using a combination of powerful Python libraries: PyVista, Trimesh, and PyTorch3D. Each of these libraries plays a distinct role in preparing the environment data for the subsequent ray tracing operations.

PyVista serves as an excellent library for structuring and visualizing the 3D environment. It provides robust tools for organizing and rendering complex 3D geometries, enabling a clear and intuitive representation of the simulation space. Complementing PyVista, the Trimesh library offers powerful modules for data manipulation of the environment. Trimesh's approach is to view the entire environment as a collection of triangular mesh cells. Each of these cells contains critical information: the coordinates of its points, the normal vector to the surface it represents, and data about adjacent cells. This detailed representation allows for efficient access to geometric features and facilitates the identification and labeling of objects with sharp edges, which is crucial for accurate modeling of diffraction effects. The combination of PyVista's visualization capabilities and Trimesh's detailed geometric representation provides a comprehensive foundation for the subsequent ray tracing operations. PyTorch3D, in particular, enhances the simulator's performance by leveraging multi-dimensional tensors, enabling highly efficient vector computations. This approach significantly accelerates the process of identifying intersection points between rays and surfaces, a computationally intensive task that forms the core of ray tracing algorithms. The synergy of these libraries—PyVista's visualization capabilities, Trimesh's detailed geometric representation, and PyTorch3D's GPU-accelerated computations—provides a comprehensive and efficient foundation for the subsequent ray tracing operations in WiPy-RT.

Following the environment import and structuring, WiPy-RT proceeds to identify and catalog key features of the simulation space. This process includes detecting and marking sharp edges, which is crucial for accurate diffraction modeling. The preprocessing stage then involves assigning material properties and free space loss coefficients to the objects in the environment, essential for accurately modeling EM wave interactions with different surfaces and materials. This detailed material characterization allows for more realistic simulations of complex urban or indoor environments.

Subsequently, the simulator configures the RIS and antennas, specifying their positions, the number of unit cells in the RIS, and the antennas' farfield patterns. This detailed setup ensures that the unique characteristics of wireless systems, particularly those incorporating RIS technology, are accurately represented. The ability to precisely model RIS structures with varying numbers of unit cells sets WiPy-RT apart from many existing simulators, allowing for more accurate predictions of RIS-assisted communication performance.

Finally, the ray tracing parameters are configured, including the number of reflections to be considered and whether diffraction effects should be enabled. This flexibility in configuration allows users to balance between simulation accuracy and computational efficiency based on their specific requirements.

B. RAY LAUNCH

The Ray Launch stage in WiPy-RT represents the core of the ray tracing process, where intelligent ray distribution is implemented to efficiently simulate EM wave propagation. To comprehensively model all possible paths from the transmitter to the receiver, an infinite number of rays with zero angle separation would theoretically be required. However, given that computational resources are finite, it is impractical to launch an infinite number of rays. Instead, WiPy-RT employs an intelligent ray launching strategy to balance accuracy and computational efficiency.

This intelligent ray launching strategy represents a significant advancement over traditional uniform ray distribution

methods. By adapting the ray density to the environment's complexity, WiPy-RT achieves superior accuracy with fewer rays, leading to substantial computational savings.

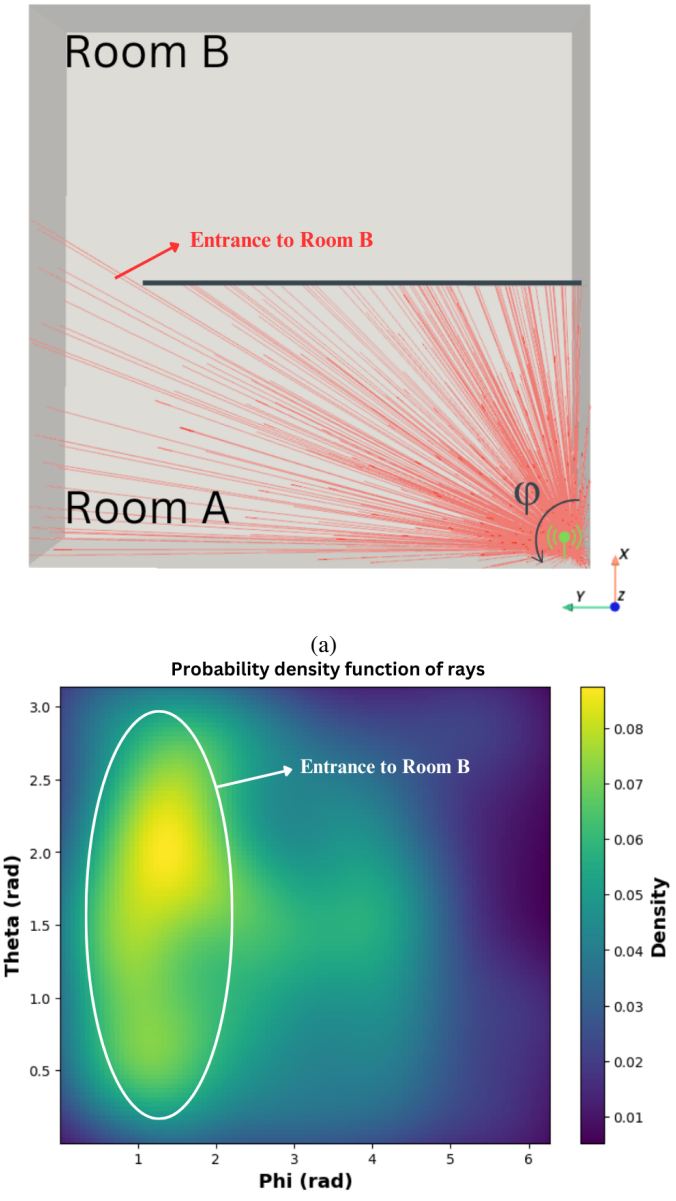


FIGURE 2: (a) Indoor environment divided into two rooms, with a source point located in the bottom right of room A. (b) Ray probability density function (PDF) distribution calculated based on the KDE model for the current source location with respect to θ and ϕ.

The simulator's ray launching process can be broken down into several key steps:

1. Initial Ray Generation: The simulator initiates the process by generating a limited number of rays (typically 1000-5000) with random azimuth (θ) and elevation (ϕ) angles. This initial set serves as a sampling of the environment.

2. KDE Model Application: These angles, along with the distance each ray has traveled, are then fed into a Kernel Density Estimation (KDE) model. The KDE model estimates the probability density function of the ray distribution based on this initial sample.

3. Adaptive Ray Generation: Based on the ray density with respect to θ and ϕ derived from the KDE model, WiPy-RT generates a larger number of rays (typically 50,000-200,000) for the actual ray tracing process. This adaptive approach ensures that more rays are allocated to directions with higher importance in the simulation.

4. Ray Tracing and Analysis: WiPy-RT then performs ray tracing with these generated rays and identifies the unique rays emitting from the source and reaching the receiver.

The choice of Gaussian KDE is deliberate, as it provides a robust statistical model for estimating the probability density function of the ray distribution. This approach allows for a more nuanced and adaptive ray launching strategy compared to uniform distribution methods.

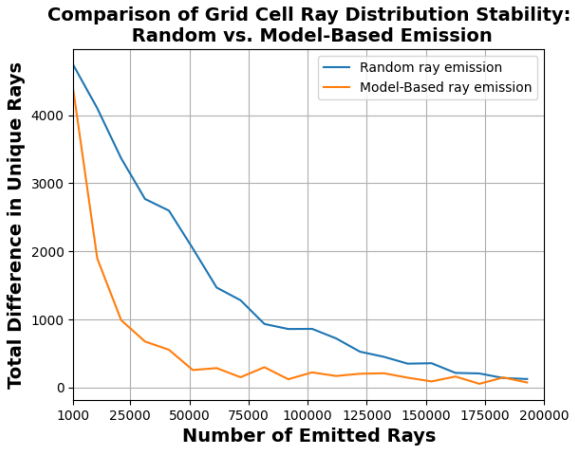


FIGURE 3: Convergence of unique ray distribution across grid cells for random vs. model-based ray emission methods, showing the total difference in unique rays as the number of emitted rays increases from 1,000 to 200,000.

The WiPy-RT's intelligent ray launching strategy is visually demonstrated in Figure 2 (a) depicts an indoor environment divided into two rooms, Room A and Room B, connected by a gate, with the antenna serving as the source point strategically positioned at the bottom right corner of Room A. The red lines emanating from this source represent the initial set of randomly generated rays, illustrating their diverse propagation paths throughout the environment. Figure 2 (b) showcases the ray density distribution derived from the KDE model, representing the probability density of rays with respect to azimuth (θ) and elevation (ϕ) angles. The color gradient, ranging from dark blue (lowest density) to bright yellow (highest density), illustrates the non-uniform distribution of rays, with a higher concentration in the direction of the gate connecting Room A and Room B. This increased density corresponds to rays with larger travel distances, critical for

accurately modeling propagation between the two rooms. This adaptive distribution ensures that WiPy-RT allocates more computational resources to ray directions of greater importance in the simulation, thereby enhancing both efficiency and accuracy of the ray tracer engine. It should be noted that the number of rays used to create the ray density distribution is dependent on the size and complexity of the environment, allowing the simulator to adapt its computational resources according to the specific characteristics of each simulation scenario.

Following the establishment of the ray density distribution, the ray tracer engine in WiPy-RT proceeds to emit rays in accordance with this derived distribution. The number of rays selected for this stage is highly dependent on the complexity of the environment, particularly the number of mesh elements present in the 3D model. To determine the optimal number of rays for the ray tracer, WiPy-RT employs an adaptive approach. This involves emitting different numbers of rays from multiple randomized points within the environment and calculating a matrix of unique ray distances to a predefined grid surface. As the number of emitted rays increases, the error between successive snapshots of this distance matrix is evaluated. The process continues until the error falls below a specified tolerance threshold. This adaptive method ensures that the simulation strikes an optimal balance between computational efficiency and accuracy, tailoring the ray count to the specific complexities of each unique environment.

To illustrate this adaptive ray selection process, consider the environment depicted in Figure 1(a). This room has a dimension of $10 \times 10 \times 2\ m^3$, with a separating wall positioned along the line from (5, 0) to (5, 8) in the XY plane, effectively dividing the space into two connected rooms. For this example, a grid is established 50 centimeters above the floor, covering the entire floor area of both rooms. This grid serves as the reference surface for calculating ray distances.

Figure 3 illustrates the convergence behavior of the ray tracing process for both random and model-based ray emission methods. The x-axis represents the number of emitted rays, ranging from 1,000 to 200,000, while the y-axis shows the total difference in unique rays received by each grid cell between consecutive simulations. The blue curve represents the random emission method, while the orange curve depicts the model-based emission approach. Both curves demonstrate a general downward trend as the number of emitted rays increases, indicating that the ray distribution across the grid cells becomes more stable with higher ray counts. Notably, the model-based emission (orange curve) converges more rapidly and maintains lower difference values throughout the range, suggesting superior efficiency and stability compared to the random emission method. This graph underscores the effectiveness of the model-based approach in achieving a consistent ray distribution with fewer emitted rays, thereby optimizing computational resources while maintaining simulation accuracy.

As demonstrated in *Figure* 3, the KDE-based ray launching strategy provides significant computational cost savings compared to uniform distribution approaches. The model-based method converges to a stable ray distribution with approximately 50,000 rays, while the uniform distribution requires almost 200,000 rays to achieve comparable stability—a 75% reduction in required rays. This translates directly to reduced computational load, with our benchmarks showing an average 70% decrease in computation time for equivalent simulation accuracy. Furthermore, the KDE approach maintains consistently lower variance in ray distribution (shown by the lower difference values at each sample point), resulting in more reliable simulations with fewer resources.

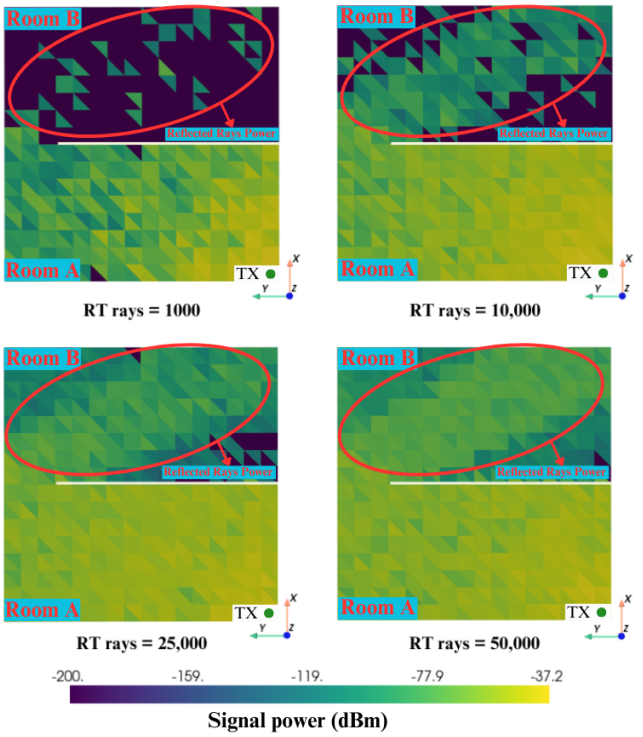


FIGURE 4: Signal power distribution on a triangular grid 0.5 meter above the floor of the two-room environment for varying numbers of emitted rays using the KDE ray distribution model, ranging from 1,000 to 50,000 rays.

Figure 4 illustrates the evolution of signal power distribution in the two-room environment as the number of emitted rays increases using the KDE ray distribution model. The figure presents four panels, each depicting the signal power distribution on a triangular grid positioned 0.5 meters above the floor for different number of emission rays: 1,000, 10,000, 25,000, and 50,000 rays, respectively. For these simulations, the ray tracing engine is configured to account for up to 5 reflections and 1 diffraction in the environment. As the number of rays increases, a clear progression in the detail and accuracy of the power distribution is observable. With 1,000 rays, the distribution appears coarse and incomplete, particularly in Room B as there is almost no trace of diffracted and diffracted rays. As the ray count increases to 10,000 and then to 25,000, the power distribution becomes more refined,

revealing intricate patterns and variations across both rooms. The simulation with 50,000 rays provides the most detailed and comprehensive representation of the signal power distribution, capturing subtle nuances in signal strength throughout the entire environment.

Having established an optimal number of emission rays through the adaptive process, WiPy-RT proceeds to simulate the behavior of RIS structures within the environment. The first step in this process involves a comprehensive analysis of all direct and first-order reflected rays that reach the RIS's surface. These rays are then systematically grouped based on their reflection history, creating distinct categories that represent different propagation paths.

For each group of rays incident on the RIS surface, it is crucial to determine the travel distance of the rays to each unit cell of the RIS. However, directly calculating this for every possible ray-cell combination would be computationally intensive. To address this challenge, WiPy-RT employs a Random Forest Regressor. This model estimates the distance between the transmitter and each unit cell of the RIS based on the populated rays within the same group. By leveraging the data from the rays that have been traced, the Random Forest Regressor can accurately predict the travel distances for all unit cells, including those not directly hit by traced rays. This approach significantly reduces computational overhead while maintaining a high degree of accuracy in modeling the RIS behavior.

While Random Forest Regression provides excellent accuracy for distance estimation between TX/RX and RIS unit cells, we also evaluated alternative regression methods to address potential computational complexity concerns. Our investigation included Linear Regression, which offers simplicity but struggles with non-linear relationships; K-Nearest Neighbors (KNN), which performs well for intermediate-sized RIS structures with moderate computational requirements; Multi-Layer Perceptron (MLP), which can capture complex relationships but requires careful architecture design; and direct numerical interpolation methods, which provide the fastest computation but with reduced accuracy for irregular ray distributions. For the 70×70 RIS configuration, the Random Forest Regressor achieved the lowest estimation error (2.3%) compared to KNN (3.1%), MLP (4.2%), and direct interpolation (5.7%), while the difference in computation time between these methods remained within a few hundred microseconds—negligible relative to the overall simulation time.

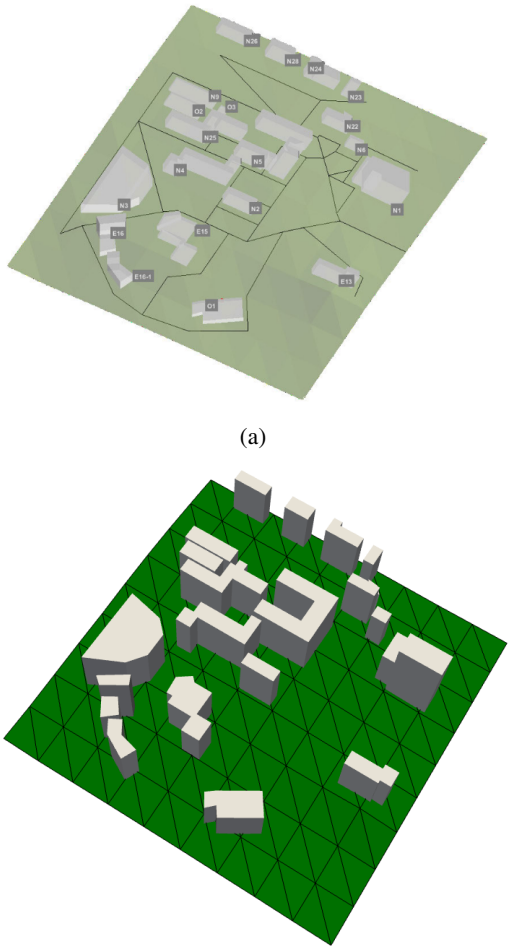
Following this estimation process, WiPy-RT then traces the rays from each RIS unit cell to the receiver. This crucial step takes into account the individual phase shift of each unit cell, allowing for a precise representation of how the RIS manipulates the incident EM waves. By considering these phase shifts, the simulator can accurately model the RIS's ability to control and optimize the propagation environment, reflecting its true impact on the overall channel characteristics. Finally, the signal power of the rays interacting with the RIS is calculated with the formula [34]

$$P_r^{TX-IRS-UE} = P_t G_t G_r \left(\frac{A}{4\pi}\right)^2 \left|\sum_{n=1}^{N} \frac{R_n e^{-j\Phi_n}}{d_{t,n} d_{r,n}}\right|^2. \quad (10)$$

where, P_t represents the transmit power of the transmitter, while G_t and G_r are the antenna gains of the transmitter and receiver, respectively. The term A denotes the surface area of the unit cell that captures and reflects each ray's transmit power. The variable N is the number of cells involved in the process. Each ray experiences a reflection loss R_n and a total phase change Φ_n, which is given by

$$\Phi_n = \frac{2\pi}{\lambda}(d_{t,n} + d_{r,n}) + \phi_n. \quad (11)$$

The distances $d_{t,n}$ and $d_{r,n}$ are the distances between the n-th IRS cell and the transmitter, and the n-th IRS cell and the receiver, respectively. The wavelength is denoted by λ, and ϕ_n represents the phase shift of the unit cells in the n-th cell.



(a)

(b)

FIGURE 5: 3D map of the surrounding of KAIST [24] (a) visualized in MATLAB, (b) imported and visualized in Python.

IV. PERFORMANCE BENCHMARKING

In this section, the performance of WiPy-RT is evaluated as compared to the research work by Choi et al. [24], which introduced the WiThRay RT engine. The chosen map, depicted in Figure 5 (a), represents a three-dimensional environment surrounding the Korea Advanced Institute of Science and Technology (KAIST) in South Korea. This particular environment was selected for its complexity and diversity of structures, providing a test scenario for assessing the capabilities of WiPy-RT. WiThRay was fully implemented in MATLAB, and consequently, the 3D map data was originally detailed in a Microsoft Excel Worksheet file (xlsx format). To adapt this data for use in the Python-based WiPy-RT engine, the Pandas library, a data processing tool in Python, is employed to import the Excel data. Following the import, the VTK library is utilized to convert the imported data into an STL format, which is compatible with the simulation environment.

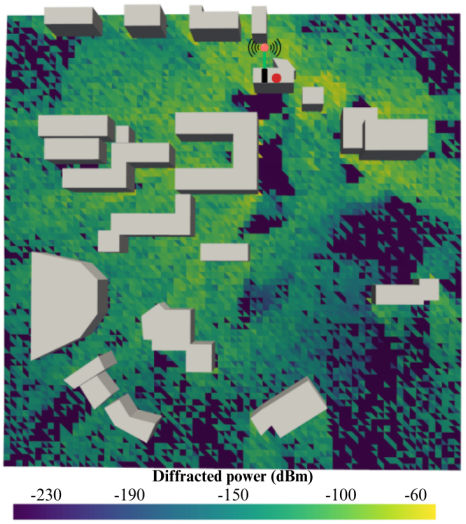
By utilizing this benchmark, the aim is to provide a comparison of WiPy-RT's performance against a state-of-the-art ray tracing engine. The KAIST map offers an urban landscape with various building heights, and shapes. This comparison will help demonstrate the strengths and potential areas for improvement in WiPy-RT, particularly in the context of modeling advanced wireless communication systems.

To evaluate WiPy-RT's performance, a ray tracing scenario using the KAIST map is set up. A TX is positioned at a fixed location in close proximity to a building. Regarding the material properties of the environment, the buildings' material is set to concrete, a common construction material in urban environments. To account for terrestrial reflections, a loss of 6 dB is implemented. As for the RT configuration, up to 6 ray reflections are allowed, enabling the modeling of complex multi-path propagation through the urban landscape. Additionally, 1 ray diffraction is included to account for the bending of waves around obstacles, modeling propagation around building edges. Furthermore, 1 ray reflection-diffraction is incorporated, which accounts for rays that are reflected first and then diffracted from an edge. Finally, RT is performed using 200,000 rays at the TX's location. The antennas used for propagation are considered as omnidirectional antennas. The signal power distribution is calculated on a surface 1 meter above the ground covering the whole map.

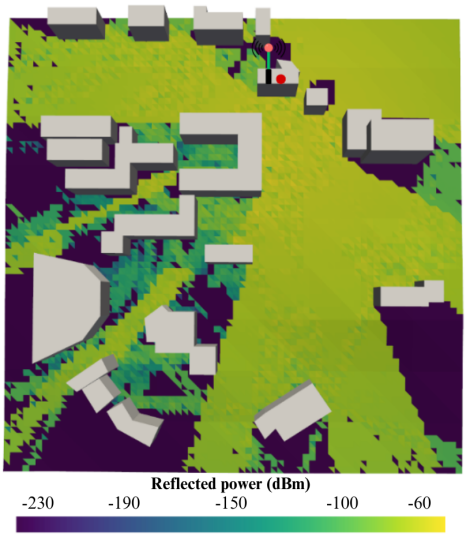
Figure 6 illustrates the distribution of signal power for both diffracted and reflected rays simulated by WiPy-RT. The simulation results are visualized on a surface positioned 1 meter above the ground, providing a view of signal propagation across the urban environment.

Figure 6 (a) depicts the power distribution of diffracted rays, while Figure 6 (b) shows the power distribution of reflected rays. The color scale, ranging from dark blue to yellow, represents signal strength in dBm, with yellow indicating stronger signals and dark blue representing weaker signals.

To calculate the signal power distribution on the map, a surface 1 meter above the ground was created, composed of small cells. Both simulators were configured to identify unique rays emitting from the Tx and reaching each cell.



(a)



(b)

FIGURE 6: Distribution of signal power of (a) diffracted rays, (b) reflected rays simulated by WiPy-RT.

The power of intersecting rays with each surface cell was then calculated, with the total signal power of each cell being the sum of all rays' power intersecting with that cell. For benchmarking purposes, 20 RT simulations were performed for different Tx locations, calculating signal power for all surface cells.

These simulations were conducted using a high-performance computer equipped with an Intel(R) Xeon(R) W-2265 CPU featuring 24 processing cores, 128 GB of memory, and an NVIDIA GeForce RTX 3080 graphics card. The average simulation time for WiPy-RT across the 20 simulations was 0.8 seconds with a variance of 0.2 seconds. In contrast, WiThRay's mean simulation time was recorded at 16 minutes and 23.9 seconds with a variance of 112.2 seconds. This substantial difference in simulation time underscores

the superior computational efficiency of WiPy-RT, making it particularly suitable for rapid, iterative analyses of complex wireless environments.

Table 1 presents a comparative analysis of the simulation performance between WiPy-RT and WiThRay. The data highlights the significant computational efficiency of WiPy-RT, particularly when utilizing GPU acceleration. WiPy-RT's GPU-enabled simulations demonstrate a remarkable average total simulation time of just 0.8 seconds, with a minimal variance of 0.2 seconds squared. This performance represents a substantial improvement over WiPy-RT's CPU-based simulations, which average 10.8 seconds with a variance of 1.1 seconds squared. The contrast becomes even more pronounced when compared to WiThRay, which requires an average of 983.9 seconds (approximately 16 minutes and 23 seconds) to complete the same simulations, with a considerably higher variance of 112.2 seconds squared.

Figure 7 presents a comparative visualization of the total signal power distribution simulated by WiPy-RT and WiThRay with a fixed location for TX (red dot in the map). The total signal power encompasses both reflected and diffracted rays, providing a view of the EM propagation in the urban environment.

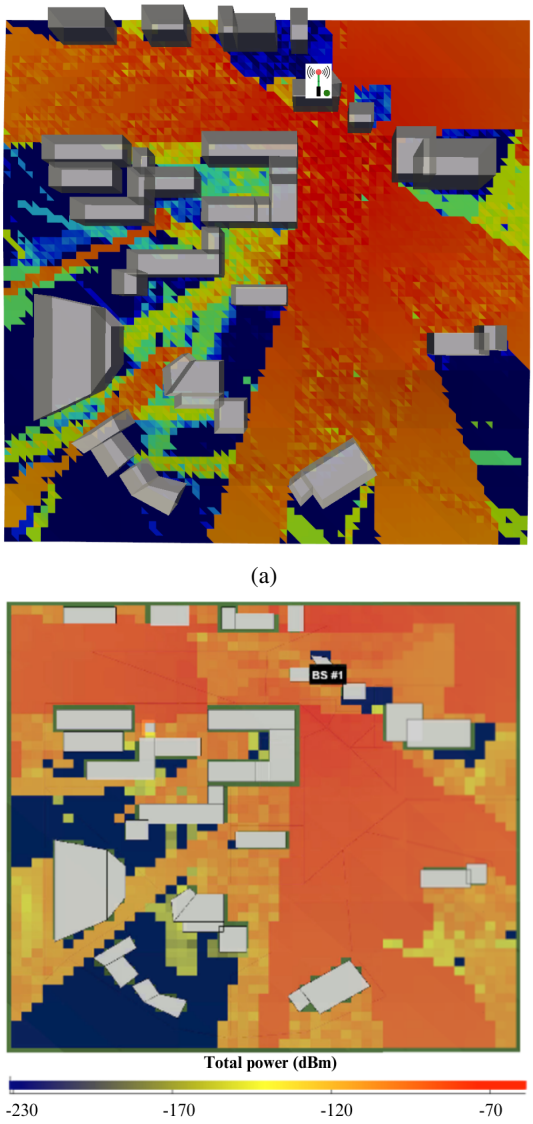
Figure 7 (a) displays the results from WiPy-RT, while Figure 7 (b) shows the simulation output from WiThRay. A striking difference between the two simulations is the computational efficiency. WiPy-RT completed the entire simulation in just 0.7 seconds, which is the sum of the diffraction and reflection calculation times mentioned earlier. In contrast, WiThRay required 15 minutes and 14 seconds to complete the same simulation, highlighting the significant performance advantage of WiPy-RT.

Moreover, the power distribution simulated by WiPy-RT exhibits a higher level of detail compared to WiThRay. This is evident in the more nuanced variations in signal strength across the map in Figure 7 (a). The WiPy-RT simulation captures finer gradations in signal power, particularly in areas around buildings and in open spaces, potentially offering a more accurate representation of real-world signal propagation.

This comparison not only demonstrates the computational efficiency of WiPy-RT but also suggests its capability to produce more detailed and potentially more accurate simulations of complex urban EM environments. The higher resolution in power distribution could be particularly valuable for precise planning and optimization of wireless networks, especially in scenarios involving RIS.

TABLE 1: Comparison of Simulation Times and Variance for WiPy-RT and WiThRay

Simulator	Total Simulation Time (s)	Variance (s^2)
WiPy-RT CPU	10.8	1.1
WiPy-RT GPU	0.8	0.2
WiThRay	983.9	112.2



(a)

(b)

FIGURE 7: Distribution of total signal power simulated by (a) WiPy-RT and (b) WiThRay.

V. RIS IMPLEMENTATION

In the previous section, WiPy-RT's engine for modeling ray reflection and diffraction between Tx and Rx was examined. This section focuses on the engine's approach to simulating RIS behavior. The integration of RIS into wireless communication models presents unique challenges due to its ability to dynamically alter the propagation environment. WiPy-RT addresses these challenges by incorporating specific algorithms and methods to accurately represent RIS interactions within the ray tracing framework. The following paragraphs detail the implementation strategy, including how RIS elements are modeled, and the computational techniques used to efficiently process these interactions.

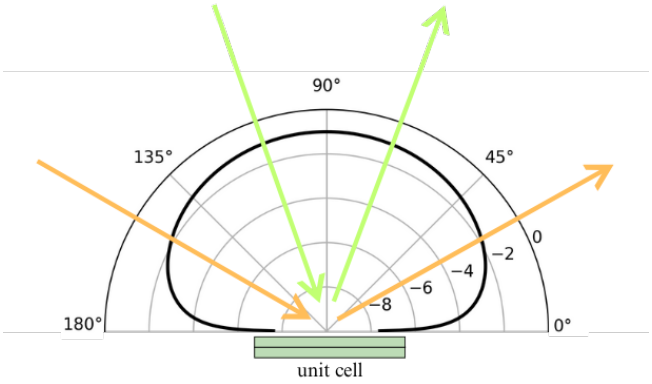


FIGURE 8: Example for reflection pattern of a RIS unit cell.

A. UNIT CELL REFLECTION PATTERN

Unit cells with varying structures exhibit distinct behaviors when reflecting incident EM waves with different Angle of Arrival (AoA) and Angle of Departure (AoD). The unit cell's reflection pattern is a factor considered by WiPy-RT in its RIS modeling approach. This reflection pattern characterizes how the unit cell responds to incoming waves from various angles and how it redirects these waves. Figure 8 provides an example of a unit cell reflection pattern, illustrating the variation in reflection characteristics for different incident angles. The unit cell reflection behavior can be simulated by high frequency simulation software such as CST Studio Suite, Ansys HFSS, etc. This reflection pattern can be inputted into WiPy-RT to create models that more closely approximate real-world scenarios.

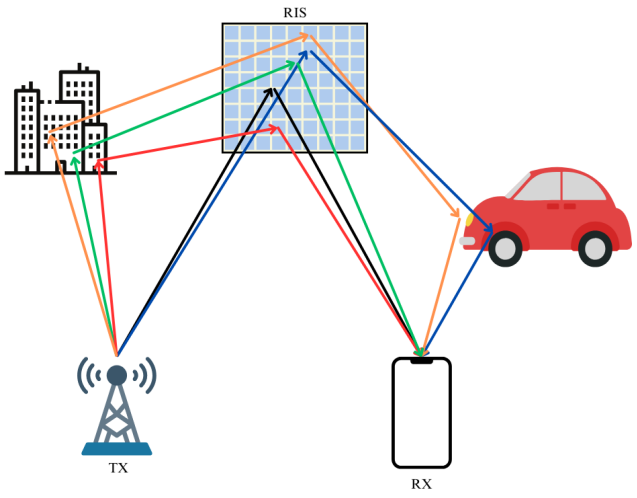


FIGURE 9: WiPi-RT's path modeling for up to 2 reflections for TX-RIS-RX ray interactions.

B. RIS MODEL

WiPy-RT incorporates a hybrid method to accurately and efficiently model the RIS structure within wireless systems. This approach balances computational efficiency with simulation accuracy, enabling the platform to handle complex RIS interactions. The model is designed to simulate TX-RX-RIS interactions for up to two reflections, as illustrated in Figure 9. This capability allows for the examination of both direct RIS reflections and secondary reflections involving the RIS, providing a more comprehensive understanding of the RIS's impact on signal propagation. The hybrid method takes into account the unique properties of RIS elements, including their ability to modify phase and amplitude, while maintaining manageable computational requirements.

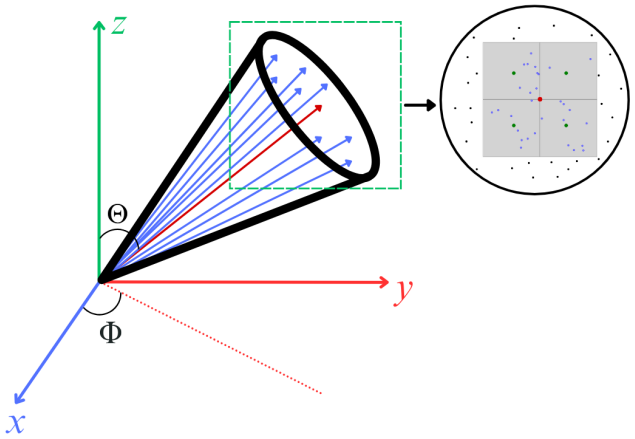


FIGURE 10: Ray tube with central ray (red) and surrounding rays (blue), with the top right subset showing a 2x2 RIS showing intersected rays (blue), unit cell centers (green), missed rays (black), and RIS center (red).

The WiPy-RT model aims to identify ray paths traveling from TX, reflecting from each center point of RIS's unit cells, and ultimately reaching the RX. Finding these exact paths poses a significant computational challenge, particularly for RIS structures with numerous unit cells. To address this, an approximate method has been implemented, offering improved efficiency while maintaining reasonable accuracy.

The process begins with inverse ray tracing at both TX and RX locations to determine accurate paths from TX/RX to the center of the RIS. Following path identification, ray tubes are generated around the paths' directions using uniform random distribution in θ and ϕ. Direct ray tracing is then performed at RX and TX locations using these constructed ray tubes.

The rays that intersect with the RIS surface are collected into separate arrays, each containing the rays' travel distances from TX/RX to the RIS surface and their corresponding intersection points. Subsequently, a Random Forest Regressor is employed to estimate path distances from TX/RX to the central points of each RIS unit cell. Once all path distances from TX/RX to the RIS unit cells are estimated, the total TX-RIS-RX paths are calculated. Finally, the RIS signal power at the RX is computed using Equations (10) and (11). Figure 10 illustrates an example of a ray tube with a central red ray surrounded by randomly distributed blue rays. The top-right

subset depicts a 2×2 RIS, where blue dots represent intersected tube rays with the surface, green dots indicate unit cell centers, black dots show missed rays, and the red dot marks the RIS surface center point.

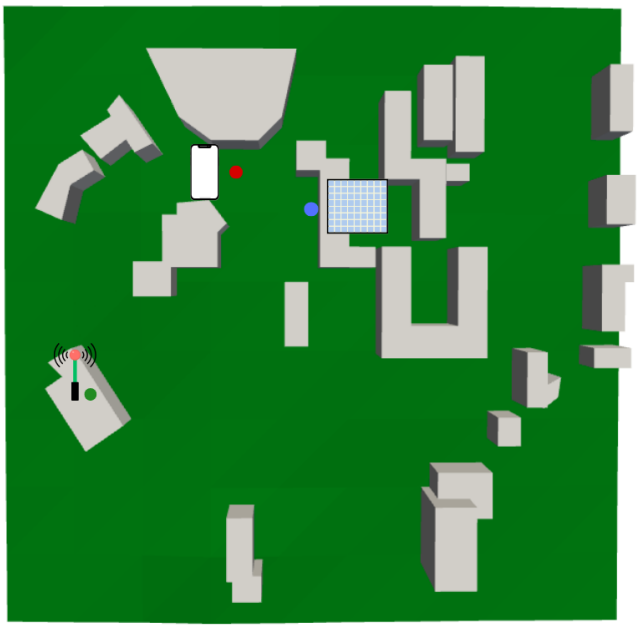


FIGURE 11: Map layout of the RIS-assisted SISO communication scenario. The TX (green circle), RIS (blue circle), and RX (red circle) positions are shown.

C. RESULTS

This section presents WiPy-RT's simulation results for an RIS-assisted SISO communication scenario. Figure 11 depicts the map layout, with TX, RIS, and RX locations marked by green, blue, and red circles, respectively. The configuration lacks a LOS path between TX and RX, while the RIS maintains visibility to both transceivers. Three RIS configurations were considered: 10×10, 30×30, and 70×70 unit cells that are installed on the building's wall shown in Figure 11. The communication system operates at 5.8 GHz, with RIS unit cell spacing set to $\lambda/2$. The primary objective was to maximize signal power at the RX by optimizing the RIS phase distribution.

For the WiPy-RT engine simulation, 1,000,000 rays were emitted from the TX, and an equal number from the RIS's center. Figure 11 illustrates the optimized RIS phase distribution and its corresponding signal power distribution. For clarity, only the signal power associated with the RIS has been plotted. This visualization allows for a focused examination of the RIS's impact on signal propagation and strength in the given environment.

Figure 12 displays the RIS reflection signal power distribution in the map for optimized RIS phase configurations. The figure presents results for three different RIS sizes with varying numbers of unit cells. On the right side of the figure,

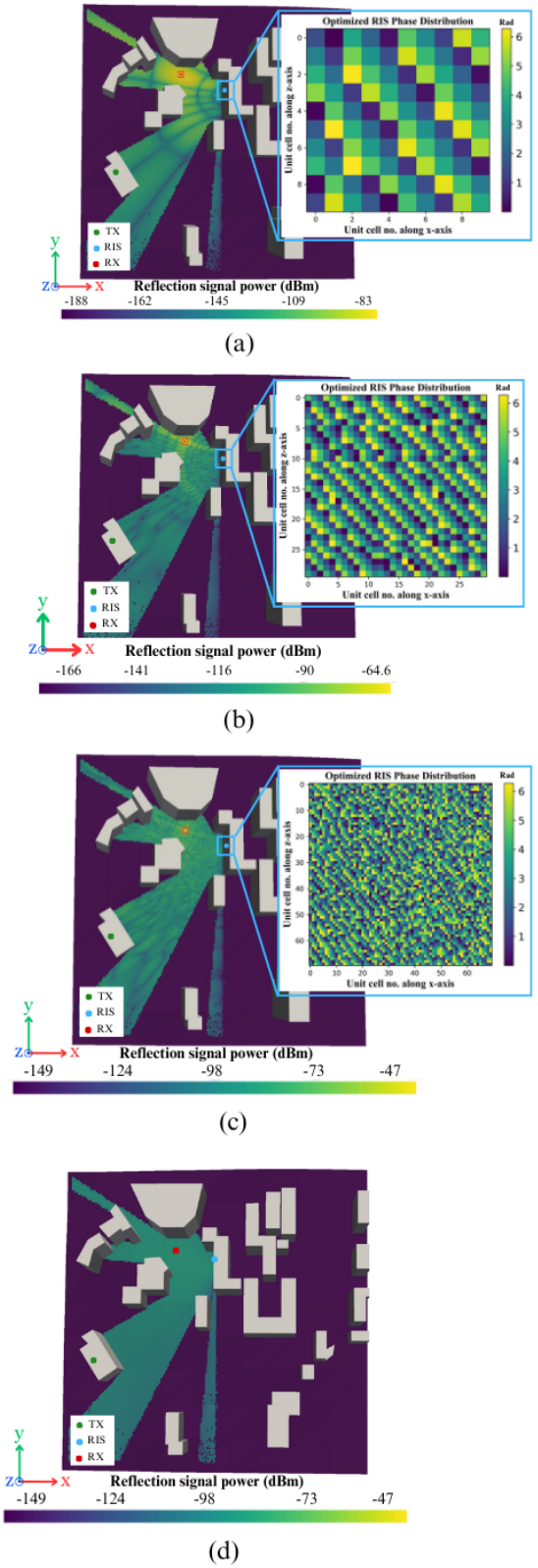


FIGURE 12: RIS reflection signal power distributions and corresponding optimized phase configurations (subsets) for (a) 10×10, (b) 30×30, and (c) 70×70 RIS structures. (d) Depicts RIS reflection signal power distribution for a 70×70 RIS with all unit cells being switched off.

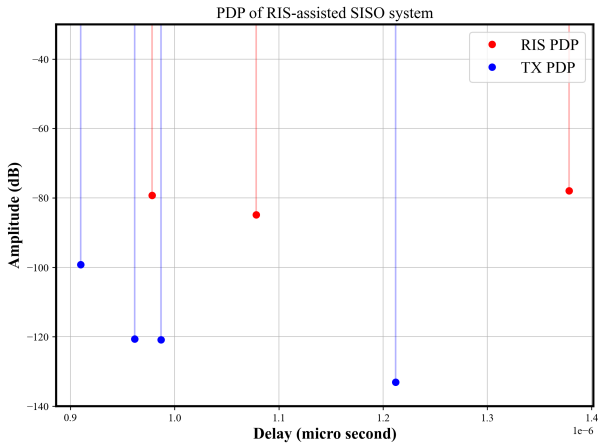


FIGURE 13: Power delay profile of direct (Tx-Rx) rays and RIS-reflected rays for the RIS-assisted SISO system.

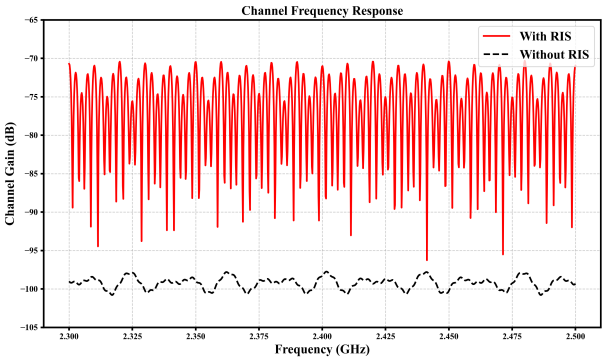


FIGURE 14: Channel frequency response comparison between scenarios with and without the optimized 70 × 70 RIS across the 2.4-2.5 GHz frequency band.

the optimized RIS phase distributions for maximum signal power at the RX are shown, while the left side illustrates the corresponding signal power distribution on the map. The results demonstrate a significant increase in signal power at the RX, ranging from -83 dBm for the 10×10 RIS to -47 dBm for the 70×70 RIS. This substantial improvement clearly indicates the effectiveness of the RIS in enhancing signal strength within the system. The simulation results suggest that for an RIS structure to be highly effective, a large number of unit cells is required. This observation underscores the importance of RIS size in achieving significant performance gains in wireless communication systems.

To further validate our RIS channel modeling approach, we analyzed both the temporal and frequency domain characteristics of the simulated channel. *Figure* 13 presents the power delay profile of the RIS-assisted SISO system, clearly distinguishing between direct TX-RX paths (blue) and RIS-reflected paths (red). The direct paths exhibit lower power levels (around -100 to -135 dB) compared to the RIS-reflected paths (approximately -80 dB), demonstrating the significant power enhancement achieved through the RIS. The temporal distribution of multipath components reveals that RIS-reflected signals arrive with delays between 0.95 and 1.35 microseconds, creating a distinct cluster of strong multipath components. *Figure* 14 illustrates the dramatic improvement in channel frequency response when using an optimized 70 × 70 RIS compared to the system without RIS. The RIS-assisted channel (solid red line) shows a gain improvement of approximately 30 dB across the entire 2.5 GHz frequency band compared to the non-RIS channel (dashed black line). The frequency-selective nature of the channel is evident in both cases, but the RIS significantly enhances the overall channel gain while maintaining a similar frequency-selective pattern.

VI. CONCLUSION

This paper presented WiPy-RT, a fast and efficient ray tracing platform specifically designed for modeling RIS-assisted wireless channels. The platform introduces several key innovations, including an intelligent ray launching method using Gaussian Kernel Density Estimation and an efficient RIS modeling approach employing Random Forest Regression. The platform's hybrid approach to RIS modeling, combining inverse and direct ray tracing methods with advanced regression techniques, offers a balance between computational efficiency and simulation accuracy.

WiPy-RT demonstrates significant performance improvements over existing solutions, with simulation times reduced from minutes to less than a second compared to state-of-the-art platforms like WiThRay. WiPy-RT's capabilities were demonstrated through a case study of a RIS-assisted SISO communication system. The results showed significant improvements in signal power at the receiver when using RIS, with larger RIS structures (up to 70×70 unit cells) providing substantial gains in signal strength.

The simulation platform proved effective in modeling complex wireless environments, including the intricate interactions between transmitters, receivers, and RIS elements. The novel approach to RIS modeling allows for accurate simulation of large-scale RIS structures, addressing a critical need in the design and optimization of next-generation wireless systems. The ability to optimize RIS phase distributions for maximum signal power highlights the potential of RIS technology in enhancing wireless communication systems.

WiPy-RT's innovative features and performance improvements contribute significantly to the field of wireless channel modeling, offering researchers and engineers a powerful tool for exploring advanced RIS-enabled communication scenarios. Future work could focus on extending WiPy-RT to support more complex scenarios, including multiple RIS structures, dynamic environments, higher-order reflections, and models that account for surface material dispersion and polarization effects in ray reflections/diffractions. Additionally, integrating more advanced optimization algorithms for RIS configuration could further improve the platform's utility in designing next-generation wireless systems.

REFERENCES

[1] Anutusha Dogra, Rakesh Kumar Jha, and Shubha Jain. A survey on beyond 5g network with the advent of 6g: Architecture and emerging technologies. *IEEE Access*, 9:67512–67547, 2020.

[2] Guan Gui, Miao Liu, Fengxiao Tang, Nei Kato, and Fumiyuki Adachi. 6g: Opening new horizons for integration of comfort, security, and intelligence. *IEEE Wireless Communications*, 27(5):126–132, 2020.

[3] Ericsson. Ericsson mobility report. Technical report, Ericsson, June 2023.

[4] Wei Jiang, Bin Han, Mohammad Asif Habibi, and Hans Dieter Schotten. The road towards 6g: A comprehensive survey. *IEEE Open Journal of the Communications Society*, 2:334–366, 2021.

[5] Shuping Dang, Osama Amin, Basem Shihada, and Mohamed-Slim Alouini. What should 6g be? *Nature Electronics*, 3(1):20–29, 2020.

[6] Carlos De Lima, Didier Belot, Rafael Berkvens, Andre Bourdoux, Davide Dardari, Maxime Guillaud, Minna Isomursu, Elena-Simona Lohan, Yang Miao, Andre Noll Barreto, et al. Convergent communication, sensing and localization in 6g systems: An overview of technologies, opportunities and challenges. *IEEE Access*, 9:26902–26925, 2021.

[7] Theodore S Rappaport, Shu Sun, Rimma Mayzus, Hang Zhao, Yaniv Azar, Kevin Wang, George N Wong, Jocelyn K Schulz, Mathew Samimi, and Felix Gutierrez. Millimeter wave mobile communications for 5g cellular: It will work! *IEEE Access*, 1:335–349, 2013.

[8] Wonil Roh, Ji-Yun Seol, Jeongho Park, Byunghwan Lee, Jaekon Lee, Yungsoo Kim, Jaeweon Cho, Kyungwhoon Cheun, and Farshid Aryanfar. Millimeter-wave beamforming as an enabling technology for 5g cellular communications: Theoretical feasibility and prototype results. *IEEE Communications Magazine*, 52(2):106–113, 2014.

[9] Jie Huang, Cheng-Xiang Wang, Lu Bai, Jian Sun, Yang Yang, Jie Li, Olav Tirkkonen, and Ming-Tuo Zhou. A big data enabled channel model for 5g wireless communication systems. *IEEE Transactions on Big Data*, 6(2):211–222, 2018.

[10] Vittorio Degli-Esposti, Franco Fuschini, Enrico M Vitucci, Marina Barbiroli, Marco Zoli, Li Tian, Xuefeng Yin, Diego Andres Dupleich, Robert Müller, Christian Schneider, et al. Ray-tracing-based mm-wave beamforming assessment. *IEEE Access*, 2:1314–1325, 2014.

[11] Zhengqing Yun and Magdy F Iskander. Ray tracing for radio propagation modeling: Principles and applications. *IEEE Access*, 3:1089–1100, 2015.

[12] Chongwen Huang, Sha Hu, George C Alexandropoulos, Alessio Zappone, Chau Yuen, Rui Zhang, Marco Di Renzo, and Merouane Debbah. Holographic mimo surfaces for 6g wireless networks: Opportunities, challenges, and trends. *IEEE Wireless Communications*, 27(5):118–125, 2020.

[13] Manuel F Catedra, J Perez, F Saez De Adana, and O Gutierrez. Efficient ray-tracing techniques for three-dimensional analyses of propagation in mobile communications: application to picocell and microcell scenarios. *IEEE Antennas and Propagation Magazine*, 40(2):15–28, 1998.

[14] F Aguado Agelet, Arno Formella, JM Hernando Rabanos, F Isasi De Vicente, and F Perez Fontan. Efficient ray-tracing acceleration techniques for radio propagation modeling. *IEEE Transactions on Vehicular Technology*, 49(6):2089–2104, 2000.

[15] Scott Y Seidel and Theodore S Rappaport. Site-specific propagation prediction for wireless in-building personal communication system design. *IEEE Transactions on Vehicular Technology*, 43(4):879–891, 1994.

[16] Joseph B Keller. Geometrical theory of diffraction. 2016.

[17] Manuel F Cátedra and Jesus Perez. *Cell planning for wireless communications*. Artech House, Inc., 1999.

[18] Allan Wainaina Mbugua, Yun Chen, Leszek Raschkowski, Lars Thiele, Stephan Jaeckel, and Wei Fan. Review on ray tracing channel simulation accuracy in sub-6 ghz outdoor deployment scenarios. *IEEE Open Journal of Antennas and Propagation*, 2:22–37, 2020.

[19] GE Athanasiadou, AR Nix, and JP McGeehan. A ray tracing algorithm for microcellular and indoor propagation modelling. In *1995 Ninth International Conference on Antennas and Propagation, ICAP'95 (Conf. Publ. No. 407)*, volume 2, pages 231–235. IET, 1995.

[20] Steven J Fortune, David M Gay, Brian W Kernighan, Orlando Landron, Reinaldo A Valenzuela, and Margaret H Wright. Wise design of indoor wireless systems: practical computation and optimization. *IEEE Computational Science and Engineering*, 2(1):58–68, 1995.

[21] Michael C Lawton and JP McGeehan. The application of a deterministic ray launching algorithm for the prediction of radio channel characteristics in small-cell environments. *IEEE Transactions on Vehicular Technology*, 43(4):955–969, 1994.

[22] Taha Alwajeeh, Pierre Combeau, and Lilian Aveneau. An efficient ray-tracing based model dedicated to wireless sensor network simulators for smart cities environments. *IEEE Access*, 8:206528–206547, 2020.

[23] Matthew R Ziemann, John S Hyatt, and Michael S Lee. Convolutional neural networks for radio frequency ray tracing. In *MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM)*, pages 618–622. IEEE, 2021.

[24] Hyuckjin Choi, Jaeky Oh, Jaehoon Chung, George C Alexandropoulos, and Junil Choi. Withray: A versatile ray-tracing simulator for smart wireless environments. *IEEE Access*, 11:56822–56845, 2023.

[25] Danping He, Bo Ai, Ke Guan, Longhe Wang, Zhangdui Zhong, and Thomas Kürner. The design and applications of high-performance ray-tracing simulation platform for 5g and beyond wireless communications: A tutorial. *IEEE Communications Surveys & Tutorials*, 21(1):10–27, 2018.

[26] Mattia Lecci, Paolo Testolina, Marco Giordani, Michele Polese, Tanguy Ropitault, Camillo Gentile, Neeraj Varshney, Anuraag Bodi, and Michele Zorzi. Simplified ray tracing for the millimeter wave channel: A performance evaluation. In *2020 Information Theory and Applications Workshop (ITA)*, pages 1–6. IEEE, 2020.

[27] Jialing Huang, Cheng-Xiang Wang, Yingzhuo Sun, Jie Huang, and Fu-Chun Zheng. A novel ray tracing based 6g ris wireless channel model and ris deployment studies in indoor scenarios. In *2022 IEEE 33rd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 884–889. IEEE, 2022.

[28] Yuanzhi Liu and Costas D Sarris. Efficient propagation modeling for communication channels with reconfigurable intelligent surfaces. *IEEE Antennas and Wireless Propagation Letters*, 21(10):2120–2124, 2022.

[29] Jialing Huang, Cheng-Xiang Wang, Songjiang Yang, Yinghua Wang, Yingjie Xu, Yingzhuo Sun, Jie Huang, and Fu-Chun Zheng. Ray tracing based 6g ris-assisted mimo channel modeling and verification. In *2023 IEEE/CIC International Conference on Communications in China (ICCC)*, pages 1–6. IEEE, 2023.

[30] Yuanzhi Liu and Costas D Sarris. Propagation modeling for ris-enabled channels based on ray-tracing and the equivalence principle. In *2023 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (USNC-URSI)*, pages 19–20. IEEE, 2023.

[31] Akira Ishimaru. *Electromagnetic wave propagation, radiation, and scattering: from fundamentals to applications*. John Wiley & Sons, 2017.

[32] Max Born and Emil Wolf. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. Elsevier, 2013.

[33] Robert G Kouyoumjian and Prabhakar H Pathak. A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface. *Proceedings of the IEEE*, 62(11):1448–1461, 1974.

[34] Mohammadreza Farashahi, Boon-Chong Seet, and Xuejun Li. Framework for propagation modeling of irs-assisted communication based on ray tracing. *Physical Communication*, 63:102301, 2024.