# **DiffeRT: An Open-Source Library for Optimizing Radio Networks with Differentiable Ray Tracing**

# Jérome Eertmans, Claude Oestges, Laurent Jacques

ICTEAM (Université catholique de Louvain) — jerome.eertmans@uclouvain.be



## Highlights

We present *DiffeRT* [1], a Differentiable Ray Tracing (RT) Python library for radio propagation that is:

• Fast: just-in-time compilation and native GPU/TPU support (thanks to JAX [2]), and Rust backend

- Scalable: support for any number of batch dimensions

**UCLouvain** 

Most existing RT tools are either **closed-source** or not differentiable.

**Ray Tracing Tools** 

Automatic differentiation is a key feature for optimization and Machine Learning (ML) applications.

To our knowledge, *Sionna* [3] is the only other open-source RT library that is both differentiable and open-source.

# **Visualizing Results**

A RT simulator is **nothing** without a good visualization tool. *DiffeRT* provides with three plotting backends:

• Matplotlib: for highly customizable plots



- **Tested**: code coverage over 80%
- Well-documented: comprehensive documentation and examples
- Flexible: extensible to any scene, radio-material, and antenna model
- **Open-source**: under the MIT license

## Context

Ray Tracing (RT) is a deterministic technique to model wave propagation (Fig. 1).



Figure 1: Ray Tracing in an urban scene, scene from [3].

Computing electromagnetic (EM) fields is usually performed in five steps (Fig. 2).

#### **Our Approach**

#### Like Sionna, DiffeRT is:

- **Differentiable** thanks to automatic differentiation
- **Open-source** and (mostly) written in Python
- Fast and scalable to GPU/TPU
- Well-documented and tested
- Includes models for ITU materials from recommendations P.2040-3
- **Unlike** *Sionna*, *DiffeRT*:
- Focuses on RT and is not a complete link-level simulation framework
- Splits RT into many small steps, avoiding the need for a monolithic simulator
- Separates the path tracing from the EM field computation
- Exposes advanced features (e.g., Fermat-based RT, graph-based visibility matrices, etc.)

#### Figure 4: Dipole antenna pattern.

## • **Plotly**: for beautiful interactive visuals



Figure 5: Ray Tracing in a simple scene.

• **VisPy**: for high-performance 3D rendering

## **Getting Started**

*DiffeRT* is available on PyPI and can be installed with pip: pip install differt.



Figure 2: Typical Ray Tracing Pipeline.

DiffeRT decouples each step of the pipeline, allowing for a flexible and modular design, while benefiting from **automatic differentiation**.

# **Importing Data**

We believe that **reusing existing material** should be easy and fast:

- Support meshes written in different formats (e.g., OBJ, PLY)
- Readers written in Rust for speed
- Utilities to load scenes compatible with Sionna [3] (Listing 1)

## **Use Cases**

We develop *DiffeRT* with a focus on researchoriented projects.

**Radio Network Optimization**: optimize the position of a base station (BS) and user equipment (UE) to maximize the received power

**Develop Machine Learning model**: in [4], we trained a model with *DiffeRT* to learn how to sample path candidates from a **differentiable** RT simulator

**Study scene dynamics**: in [5], we introduce a new tool, the **Multipath Lifetime Map**, to study the coherence time of multipath components



Pre-built wheels are available on Linux, macOS, and Windows, for Python 3.10+.

#### Source Code & Documentation

DiffeRT's code is published on GitHub and provides with a comprehensive documentation. Don't hesitate to star the project on GitHub!





Documentation

GitHub

## **Future Features**

As of v0.1, *DiffeRT* is still under development. For v1.0, we are working on:

- Implementing diffraction coefficients
- Support for point clouds

# Download Sionna scenes download\_sionna\_scenes() # Find path to the scene file scene\_file = get\_sionna\_scene( "simple\_street\_canyon"

#### # Load the scene

scene = TriangleScene.load\_xml(scene\_file) # Optional: plot the scene fig = scene.plot(backend="plotly") fig.show()

Listing 1: Loading a scene from *Sionna* [3].

Figure 3: Multipath Lifetime Map.

Educational purposes: teach students how RT works with comprehensive examples and interactive visualizations



Generative ML model

Improved Ray Launching algorithms

We also welcome contributions or suggestions from the community, so don't hesitate to reach out on GitHub!

## References

- Jérome Eertmans. *DiffeRT*. Version 0.1.0. https://github.com/jeertmans/DiffeRT. 2025.
- [2] James Bradbury, Roy Frostig, Peter Hawkins, et al. JAX: composable transformations of Python+NumPy programs. Version 0.6.1. 2018. URL: http://github.com/jaxml/jax.
- Jakob Hoydis, Sebastian Cammerer, Fayçal Ait Aoudia, et al. *Sionna*. Version 1.0.2. https://nvlabs.github.io/sionna/. 2022.
- [4] Jérome Eertmans, Nicola Di Cicco, Claude Oestges, et al. "Towards Generative Ray Path Sampling for Faster Point-to-Point Ray Tracing". In: 2025 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN) (2025 ICMLCN). Barcelona, Spain, May 2025, pp. 1–6.
- Jérome Eertmans, Enrico M. Vitucci, Vittorio Degli-Esposti, et al. "Comparing Differentiable and Dynamic Ray Tracing: Introducing the Multipath Lifetime Map". In: 2025 19th European Conference on Antennas and Propagation (EuCAP) (Eu-CAP 2025). Stockholm, Sweden, Mar. 2025, pp. 1–5.

Multipath Lifetime Map