

DeepMIMO

The Community Database and Toolchain for Ray-tracing Datasets

[4.0.0beta is now available](#)



Wireless
Intelligence Lab

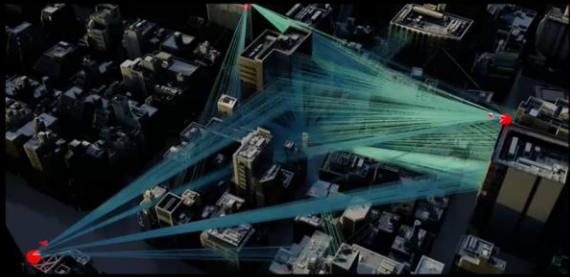
AI-RANTM
ALLIANCE

Seamless, Scalable, Reproducible

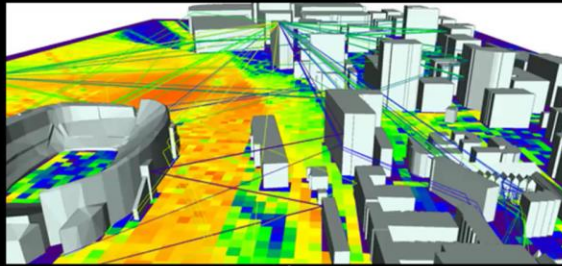
DeepMIMO's mission is to accelerate AI wireless research by making ray tracing data easy to access and share at scale.

DeepMIMO Bridges Ray Tracer Datasets and Simulation Tools

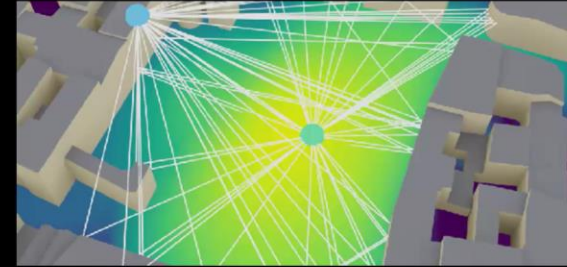
 **nvidia**. AODT



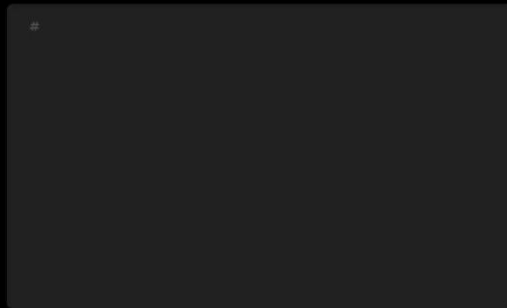
REMCOM InSite



 **nvidia**. Sionna RT



DeepMIMO Parameters



Unified DeepMIMO Scenario

DeepMIMO Database

DeepMIMO Generator



DeepMIMO Dataset



 **nvidia**.

Sionna

 **interdigital**.

NeoRadium

DeepMIMO

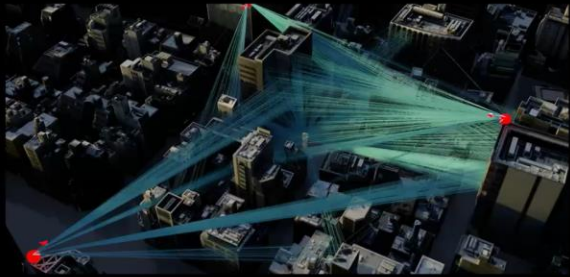
Processing & Plotting

 **MathWorks**

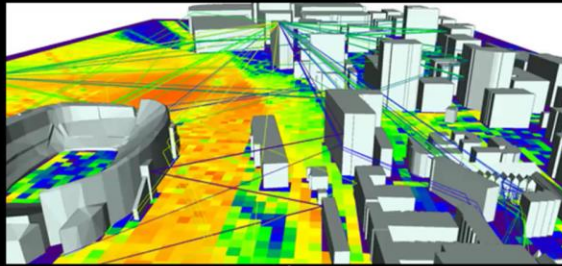
5G Toolbox

Convert Ray Tracing Simulations to DeepMIMO

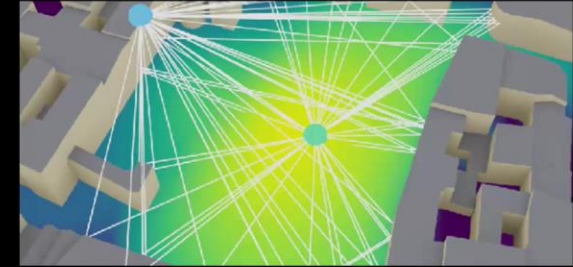
 NVIDIA AODT



 REMCOM InSite



 NVIDIA Sionna RT



One standard format for all data

DeepMIMO Scenario

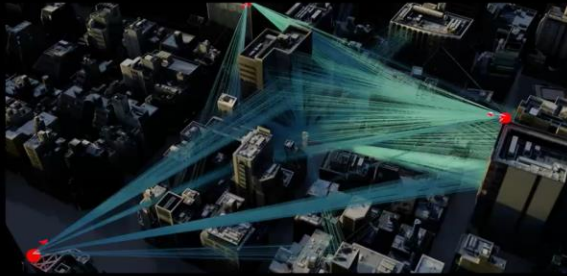


```
import deepmimo as dm  
dm.convert(raytracing_folder)
```

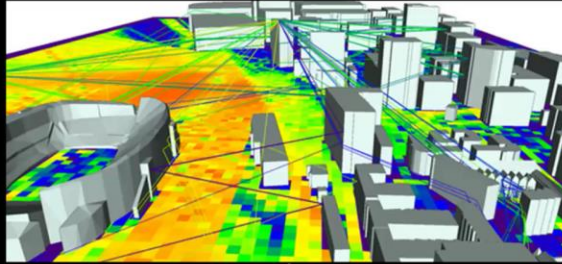
Unlock Interoperability

Convert Ray Tracing Simulations to DeepMIMO

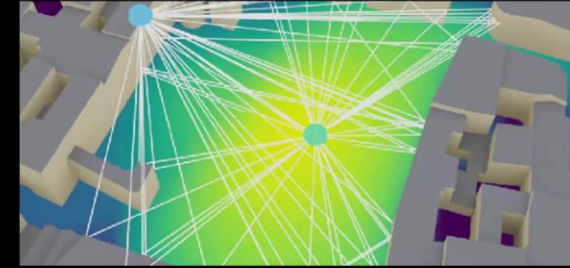
 **NVIDIA** AODT



 **REMCOM** InSite



 **NVIDIA** Sionna RT



One standard format for all data

DeepMIMO Scenario

DeepMIMO Database



```
import deepmimo as dm  
dm.convert(raytracing_folder)
```



Growing Database of Multiple Ray Tracers

Unlock Interoperability

Enable Machine Learning at Scale

Enable Ray Tracer Comparisons

Generate Wireless Channel Datasets

```
import deepmimo as dm

# Load a scenario
dataset = dm.load('asu_campus_3p5')

# Instantiate channel parameters
params = dm.ChannelParameters()

# Configure BS antenna array
params.bs_antenna.shape = [8, 4] # 8x4 array

# Configure UE antenna array
params.ue_antenna.shape = [1, 1] # Single antenna

# Configure OFDM parameters
params.ofdm.bandwidth = 10e6 # 10 MHz

# Generate frequency-domain channels
channels = dataset.compute_channels(params)
```

```
import deepmimo as dm

dm.download('asu_campus_3p5')
```

DeepMIMO Scenario

DeepMIMO Database

DeepMIMO Generator

Flexibility

Reproducibility

Antenna Array

Accurately defined

Doppler & Mobility

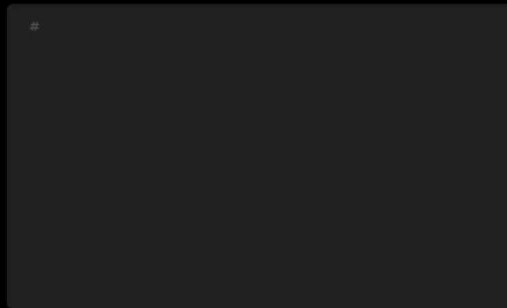
Easy to share

Integrate DeepMIMO with Link/System-level Simulators



```
from deepmimo.integration import SionnaAdapter
from deepmimo.integration import NeoRadiumAdapter
from deepmimo.integration import Matlab5GAdapter
```

DeepMIMO Parameters



DeepMIMO Scenario

DeepMIMO Generator



DeepMIMO Dataset

Portable
All you need is Python

Interoperable
Not locked to one full chain



Sionna



NeoRadium


DeepMIMO

Processing & Plotting






5G Toolbox

Search 100s of Scenarios based on Ray Tracing Parameters

Search by scenario title... 15 per page < << Page 1 of 11 > >>  Advanced Settings

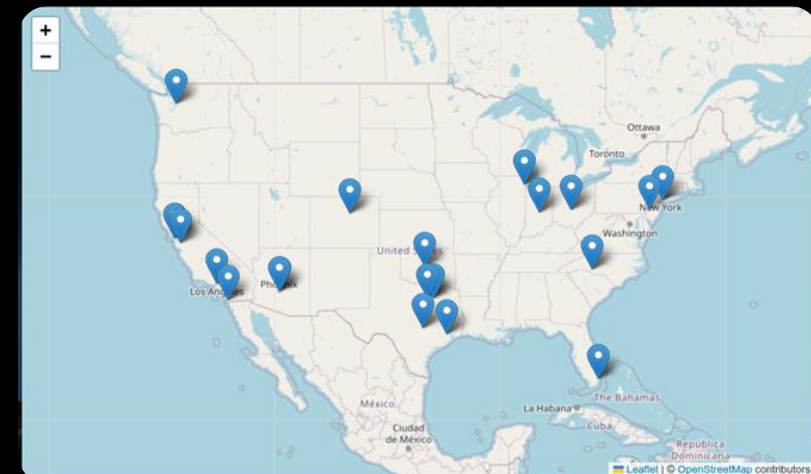
152 TOTAL RESULTS [Copy](#) **SELECT** ▼ Mir Ma Mir Ma All ▼ All ▼

SCENARIO	IMAGE	BAND	NUM RX	MAX REFLECTIONS	RAYTRACER	ENVIRONMENT
asu_campus_3p5		sub6	85158	6	Insite	outdoor
city_0_newyork_3p5		sub6	16306	3	Insite	outdoor
city_1_losangeles_3p5		sub6	19914	3	Insite	outdoor

DeepMIMO Database



World Map Available



+150 Scenarios Indexed by:

Ray-tracing parameters

Location, Environment, etc.

Use APIs to Interface with DeepMIMO Database Directly in Code



Download

```
import deepmimo as dm

scenario_name = 'asu_campus_3p5'
dm.download(scenario_name)
```



Search

```
query = {'maxReflections': {'min': 5}}
search_results = dm.search(query)
```



Upload

```
import deepmimo as dm

dm.upload('my_converted_dm_scenario')
```

Online Dashboard to Manage Your Submissions

DeepMIMO

Home Get Started Documentation Visualizer Scenarios Publications Legacy Help About

Admin Dashboard

- Pending Submissions
- Submission Logs
- User Scenario
- Scenarios
- Applications
- Mailing List
- API Key
- Download Stats
- Docs Upload

Pending (1) Approved (149) Rejected (0)

Select All

New Scenario

SCENARIO


asu_campus_3p5_90861711

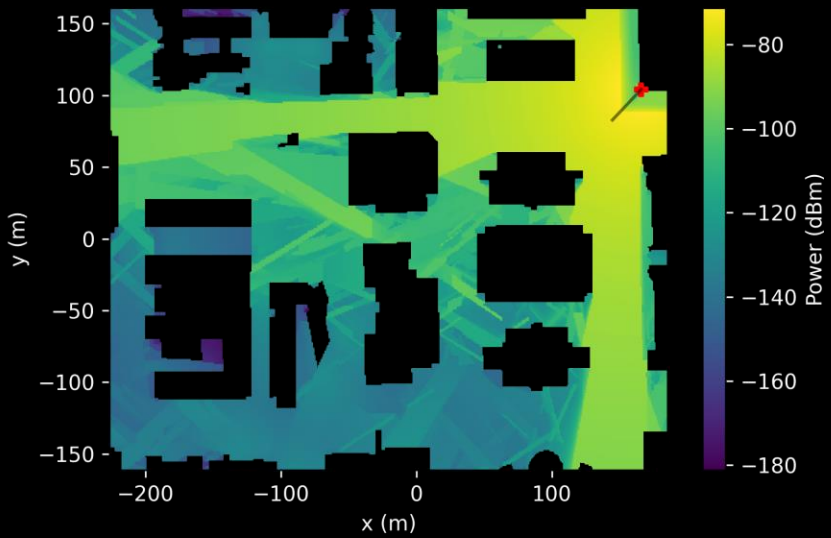
Submitted by: jmoraispk@gmail.com


RT Source: Present Delete Source

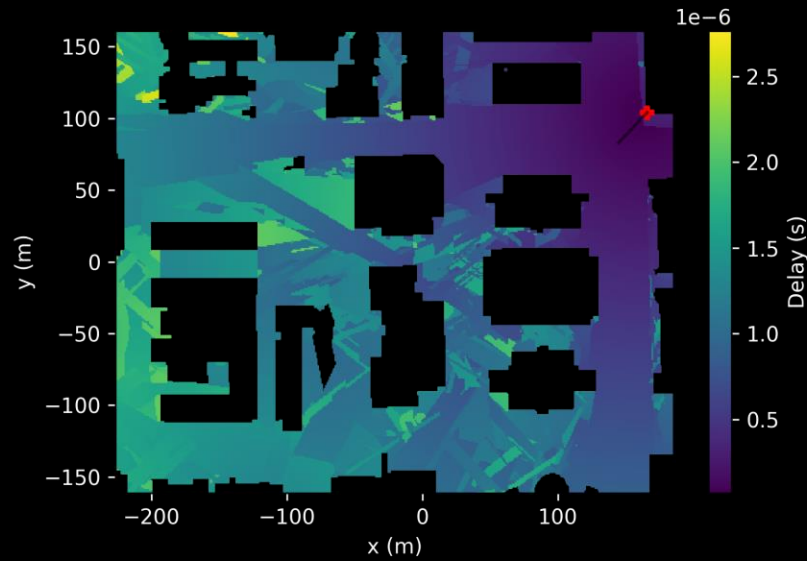
Primary Parameters Advanced Parameters


Leverage Many Built-in Processing and Visualization Tools

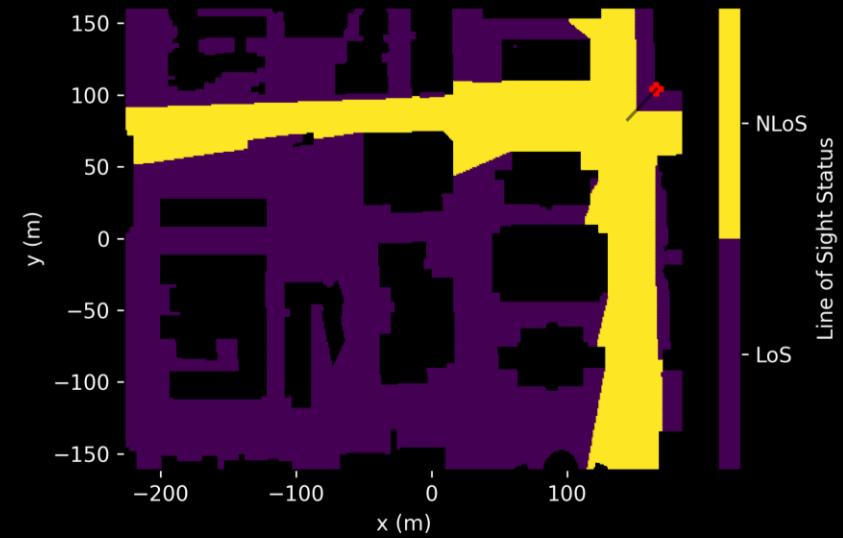

`dataset.power.plot()`




`dataset.delay.plot()`




`dataset.los.plot()`



Manipulate Data with Matrix-based Objects



```
dataset.power # [n_rx, n_paths]
```



```
dataset.phase # [n_rx, n_paths]
```



```
dataset.delay # [n_rx, n_paths]
```



```
dataset.aoa_az # [n_rx, n_paths]
```



```
dataset.aoa_el # [n_rx, n_paths]
```



```
dataset.aod_az # [n_rx, n_paths]
```



```
dataset.aod_el # [n_rx, n_paths]
```



```
dataset.inter # [n_rx, n_paths]
```



```
dataset.inter_pos # [n_rx, n_paths, n_inter, 3]
```



```
dataset.rx_pos # [n_rx, 3]
```



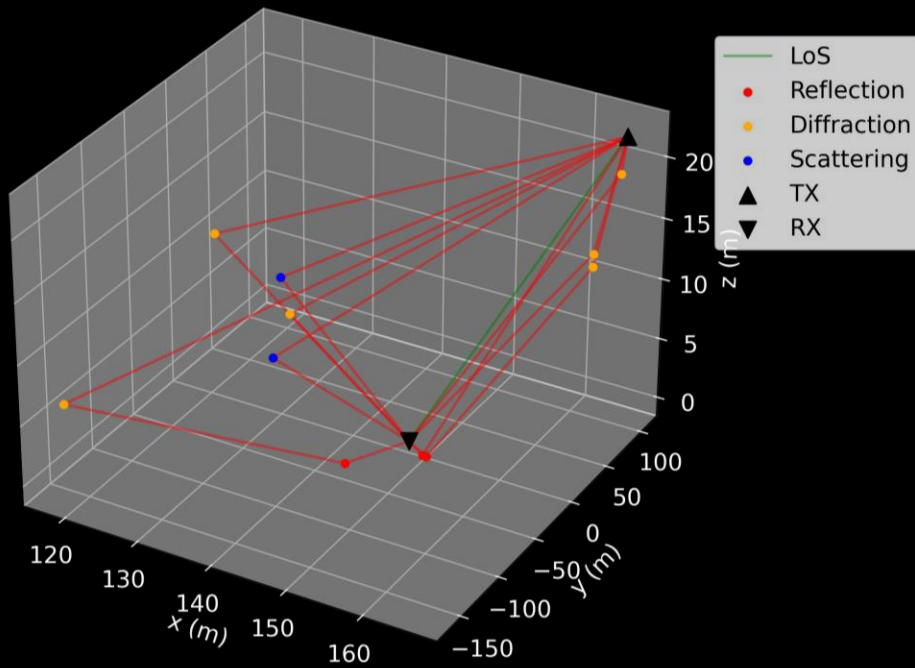
```
dataset.tx_pos # [n_tx, 3]
```

Plot Rays and Scene Information

Rays with Matplotlib



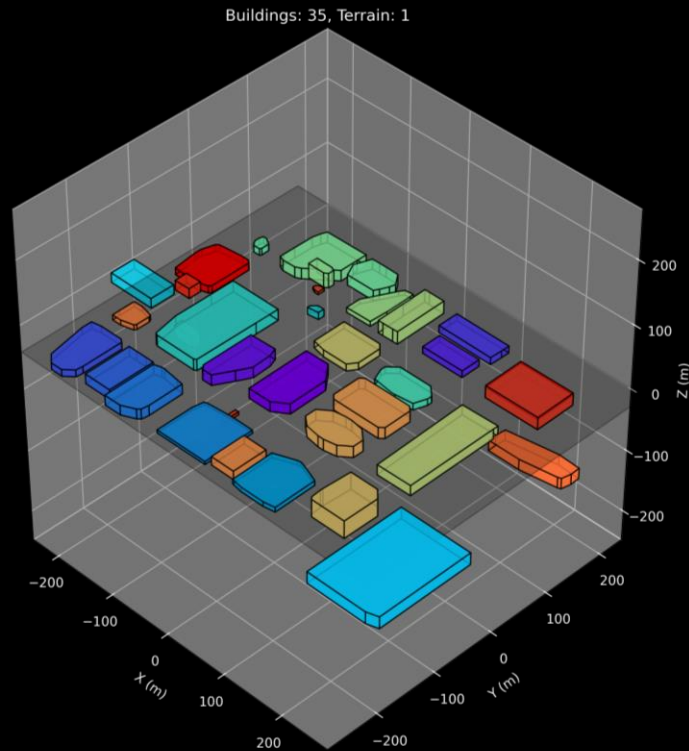
```
dataset.plot_rays()
```



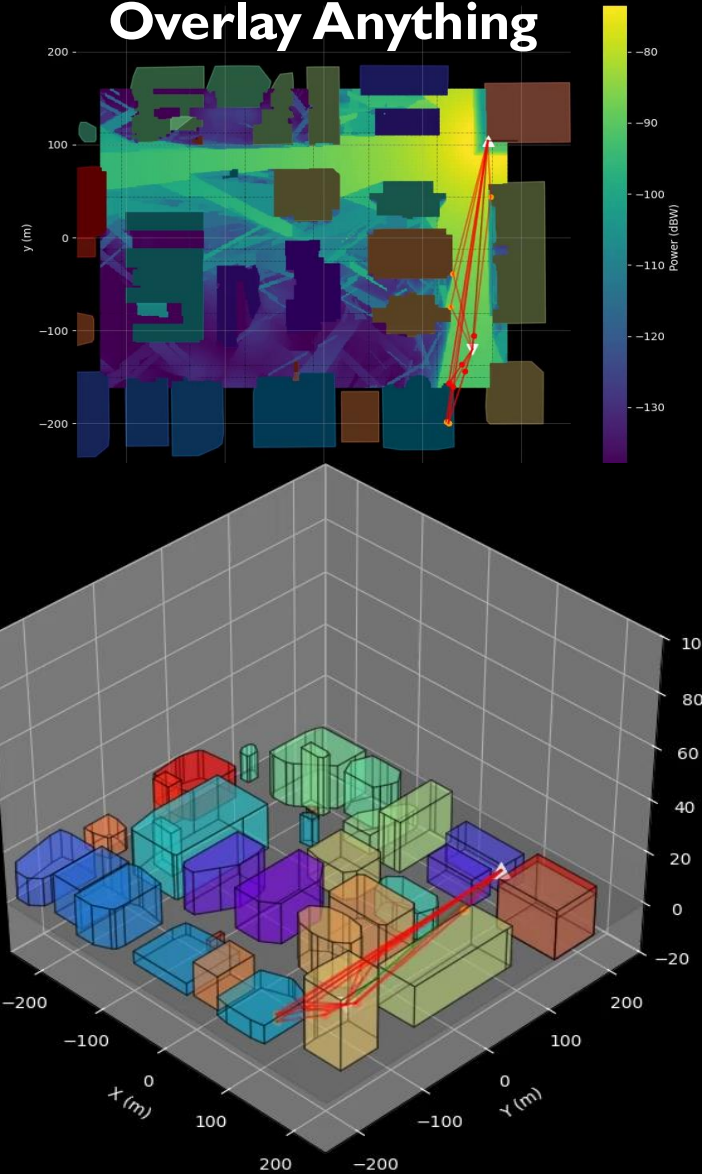
Objects by Material in the Scene



```
dataset.scene.plot()
```



Overlay Anything



Train and Benchmark Machine Learning Models on Hundreds of Scenarios in a Few Lines of Code

```
1 import deepmimo as dm
2
3 scenarios = dm.search({
4     'bands': ['sub6'], 'environment': 'outdoor'
5 })
6
7 for scenario in scenarios:
8     dataset = dm.load(scenario)
9
10    # Channel Compression
11    model.train(dataset.channels)
12
13    # Coverage Prediction
14    model.train(dataset.rx_pos, dataset.power)
15
16    # CSI Localization
17    model.train(dataset.rx_pos, dataset.channels)
18
19    # LoS Identification
20    model.train(dataset.los, dataset.power)
21    model.train(dataset.los, dataset.channels)
22
23    # Position-aided Beam Prediction
24    beams = get_codebook(dataset.channels, n_beams=32)
25    model.train(dataset.rx_pos, beams.max(axis=1))
```

Find and Reproduce Research with the Database of Publications

TASKS

Beam Management

Channel Estimation

Channel Prediction

Handover

ISAC

Localization

Power Control

Scheduling

APPLICATIONS

AR/VR/XR

Cell-Free

Energy Efficiency

Indoor

IoT

Mobility

RIS

Security

UAV

V2X

mmWave

Require ALL selected tags (AND)
 Clear All

Search by title, author, or journal

15 per page ▾

<<
<
Page 1 of 53
>
>>

⚙️ Advanced Settings

789 TOTAL RESULTS

Copy

Min Y

Max Y

Min

Max

All Types ▾

FILTER TAGS... ▾

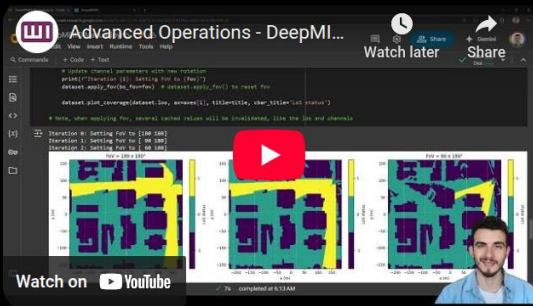
TITLE	AUTHORS	YEAR ↓	CITATIONS	JOURNAL/VENUE	TAGS
Redes Neurais com Aprendizagem Profunda Aplicadas à Estimaco de Canais Sem Fio MIMO	WR Alves	2025	0	WR Alves	-
Y-Twin, 3D RT 및 시나리오 자동화 플랫폼	김건하 , 박정훈	2025	0	한국통신학회 학술대회논문집	-
IJCNC 01	MS Aljumaily , H Li	2025	0	ijcnc.com	MmWave

And More ...

Tutorials

Online Visualizer


Forum, FAQ, Roadmap

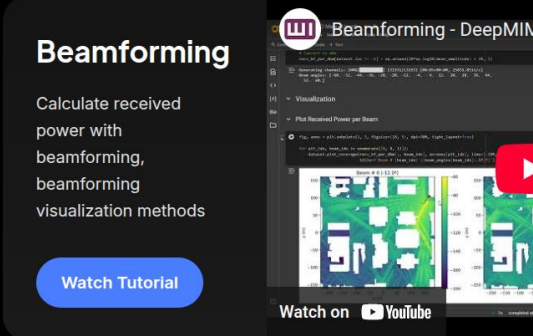


Advanced Operations

FoV analysis for receivers

Watch Tutorial


Watch on  YouTube

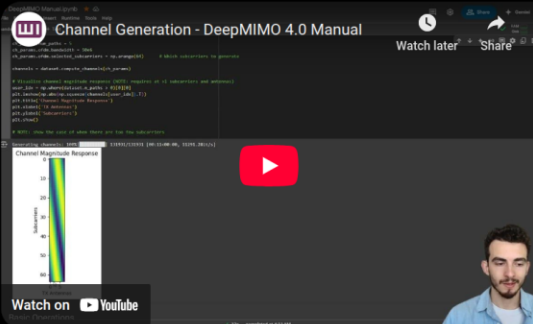


Beamforming

Calculate received power with beamforming, beamforming visualization methods

Watch Tutorial


Watch on  YouTube

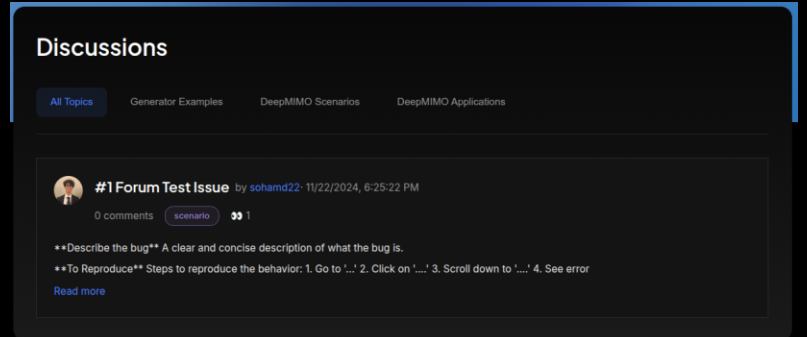
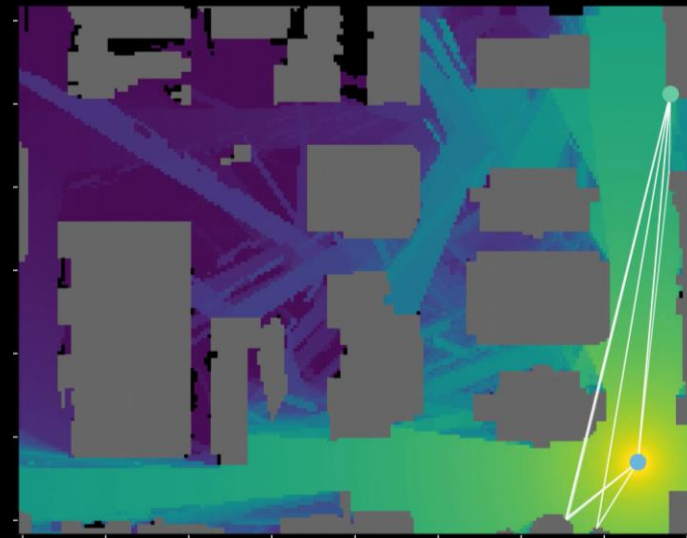
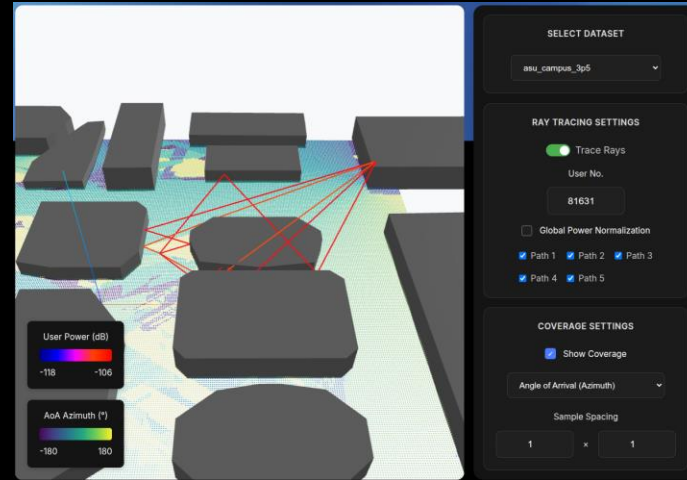


Channel Generation

Configuring channel generation and generating time-domain and OFDM channel responses

Watch Tutorial

Watch on  YouTube



Discussions

All Topics Generator Examples DeepMIMO Scenarios DeepMIMO Applications

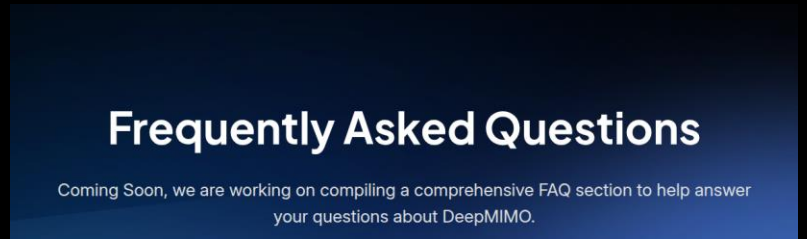
#1 Forum Test Issue by sohamd22 · 11/22/2024, 6:25:22 PM

0 comments scenario 1

Describe the bug A clear and concise description of what the bug is.

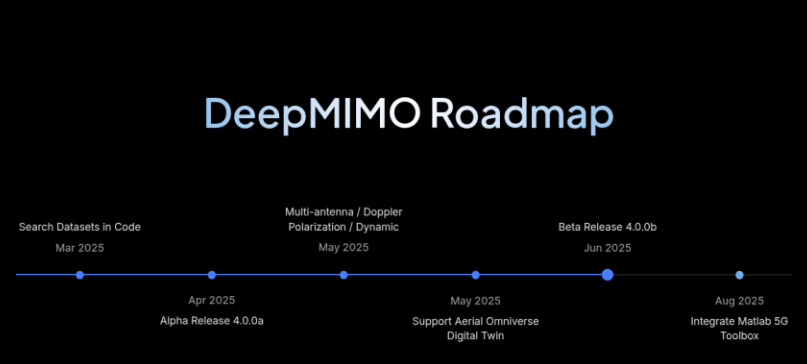
To Reproduce Steps to reproduce the behavior: 1. Go to '!'. 2. Click on '!'. 3. Scroll down to '!'. 4. See error

[Read more](#)



Frequently Asked Questions

Coming Soon, we are working on compiling a comprehensive FAQ section to help answer your questions about DeepMIMO.



DeepMIMO Roadmap

Search Datasets in Code Mar 2025

Multi-antenna / Doppler Polarization / Dynamic May 2025

Beta Release 4.0.0b Jun 2025

Apr 2025 Alpha Release 4.0.0a

May 2025 Support Aerial Omniverse Digital Twin

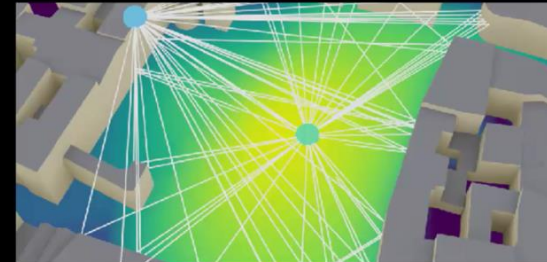
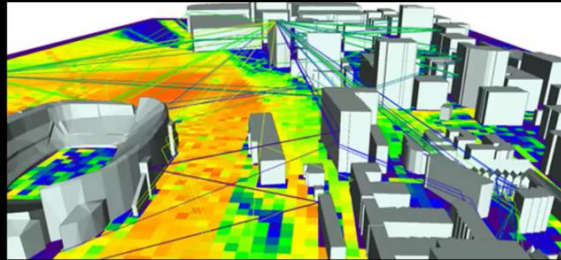
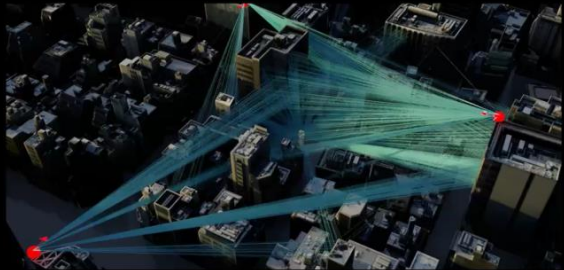
Aug 2025 Integrate Matlab 5G Toolbox

Enhancement I: OAI Integration

 **NVIDIA**. AODT

 **REMCOM**® InSite

 **NVIDIA**. Sionna RT




DeepMIMO Parameters

Unified DeepMIMO Scenario

DeepMIMO Generator

DeepMIMO Dataset

 **NVIDIA**.
Sionna

 **interdigital**.
NeoRadium

DeepMIMO
Processing & Plotting

 **MathWorks**.
5G Toolbox

OPEN AIR
INTERFACE

Enhancement 2: Raytracing Pipelines

Raytracer agnostic

1. Scene Extraction

e.g.: Open Street Maps, ...

2. Scene Processing

e.g.: Assing Materials → Surface,..

Raytracer dependent

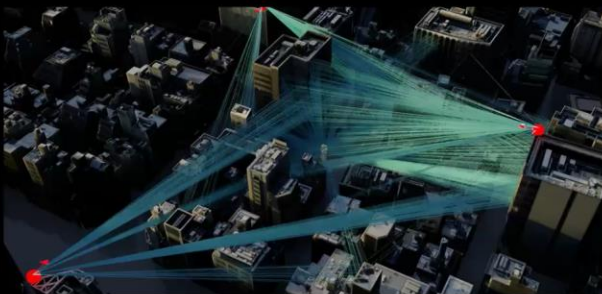
3. Scene Export

e.g.: Write scene in XML, USD, ...

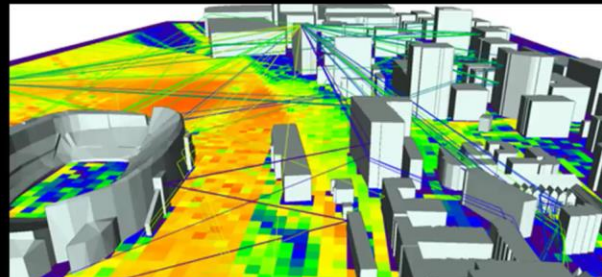
4. Raytracer Configuration

e.g.: Number of Diffractions, ...

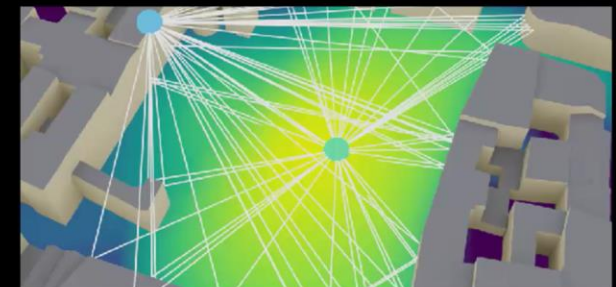
 **NVIDIA** AODT



 **REMCOM** InSite



 **NVIDIA** Sionna RT



Available Now

deepmimo.net

```
pip install --pre deepmimo
```