Multi-cell Outdoor Channel State Information Dataset (MOCSID)

Mohamed El Mehdi Makhlouf, Maxime Guillaud, Yamil Vindas Yassine

Multi-cell Outdoor Channel State Information Dataset (MOCSID)

Mohamed El Mehdi Makhlouf, Maxime Guillaud, Yamil Vindas
Inria, INSA Lyon, CITI, UR3720, 69621 Villeurbanne, France
Email: {mohamed-el-mehdi.makhlouf,maxime.guillaud,yamil.vindas-yassine}@inria.fr

***Abstract*—We introduce MOCSID, a multi-cell outdoor channel state information dataset of synthetic channel state information (CSI) samples mimicking an outdoor campus scenario, including multiple base stations with partially overlapping coverage areas and pedestrian user mobility. The scenario is characterized by a high density of base stations (10 base stations within a 625 m x 535 m area) and includes a mixture of non-line-of-sight and line-of-sight propagation. MOCSID includes user locations, timestamps, velocity and multipath component information (delays and path coefficients), following realistic pedestrian user mobility patterns generated using the probabilistic roadmap algorithm, and captures key signal propagation characteristics including path loss, shadowing, and multipath effects. Since MOCSID is intended as a reference for the development and validation of channel charting algorithms, we put particular emphasis on the spatial consistency of the synthetic data. With this dataset, we aim to foster progress in channel charting research by facilitating entry into the field and encouraging reproducibility, collaboration, and benchmarking within the community. MOCSID was generated using the NVIDIA Sionna ray tracing tool; the codebase used to generate the dataset as well as the scene description data and user movement patterns are also publicly available, allowing for easy replication, reproduction, or extension.**

I. Introduction

Channel charting [1], [2] is an application of self-supervised learning that maps high-dimensional channel state information (CSI) to a low-dimensional spatial representation, referred to as a "channel chart". This chart captures the relative positioning of user equipment (UE) within an environment, where points close to each other on the chart represent physically nearby locations. A key advantage of channel charting is its self-supervised nature: it relies solely on passively collected CSI, eliminating the need for external location data such as from Global Navigation Satellite Systems (GNSS) information or costly measurement campaigns. This approach enables location-based applications to operate with significantly reduced overhead.

Obtaining real-world data for channel charting is often challenging and expensive. The validation and evaluation of current channel charting approaches requires large datasets including mobility with spatial and temporal consistency, as well as inter-user and inter-base station (for the overlapping areas) consistency which are rarely available in existing datasets. Furthermore, existing real-world datasets may not meet and satisfy some of the new constraints that define novel scenarios. For instance, the DICHASUS [3] and MaMIMO-UAV [4] datasets do not include wide coverage areas spanning multiple cells. Synthetic datasets, generated through simulations such as ray tracing, offer a practical alternative by providing the flexibility to create controlled and customizable environments. Ray tracing models are preferred over stochastic models for wireless communication simulations due to their ability to accurately capture the geometric details of the environment, ensuring precise modeling of signal interactions and propagation effects in complex and dynamic settings. The DeepMIMO dataset [5] offers a wide range of pre-defined scenarios including multi-cell set-ups. However, support for user mobility is limited to a set of pre-computed locations; furthermore, it relies on a proprietary ray-tracer which limits reproducibility and options to extend the work. The use of these datasets offers a viable and cost-effective solution, facilitating the exploration of a broad spectrum of conditions that might otherwise prove challenging to replicate in real-world settings.

This paper presents a new synthetic dataset generated using the NVIDIA Sionna ray tracer [6] that simulates mobile users in an outdoor multi-cell environment representative of a campus setting. We have followed the best practices [7] for documenting and making the data reusable and reproducible. We expect it to serve as a valuable resource for researchers to develop and benchmark channel charting techniques and other data-driven methods relying on CSI. Our main contributions can be summarized as follows:

- We define an outdoor MIMO (multiple-input multiple-output) scenario situated in a large, campus-like environment featuring a dense distribution of base stations and encompassing hundreds of user trajectories.
- This extensive pre-computed dataset, along with its source code, is made publicly available to ensure compliance with the FAIR (Findable, Accessible, Interoperable, and Reusable) data principles. The source code for our implementation is available for download at [8] and the dataset is available from [9].

The rest of the paper is structured as follows. First, in Section II, we describe the simulation scenario and parameters, emphasizing the scattering environment, the base stations (BSs), and the user mobility. In Section III, we detail the dataset's storage format and structure, providing guidance on its usage. In Section IV, we demonstrate the value of the dataset by showcasing initial channel charting results obtained using it. Finally, in Section V we present the main conclusions of this work.

II. Scenario description

The scenario includes 10 base stations positioned at a height of $25\ \mathrm{m}$, in an outdoor campus-like setup covering an area of dimensions $625\ \mathrm{m} \times 535\ \mathrm{m}$, serving mobile users with pedestrian-like mobility patterns (see Fig. 1).

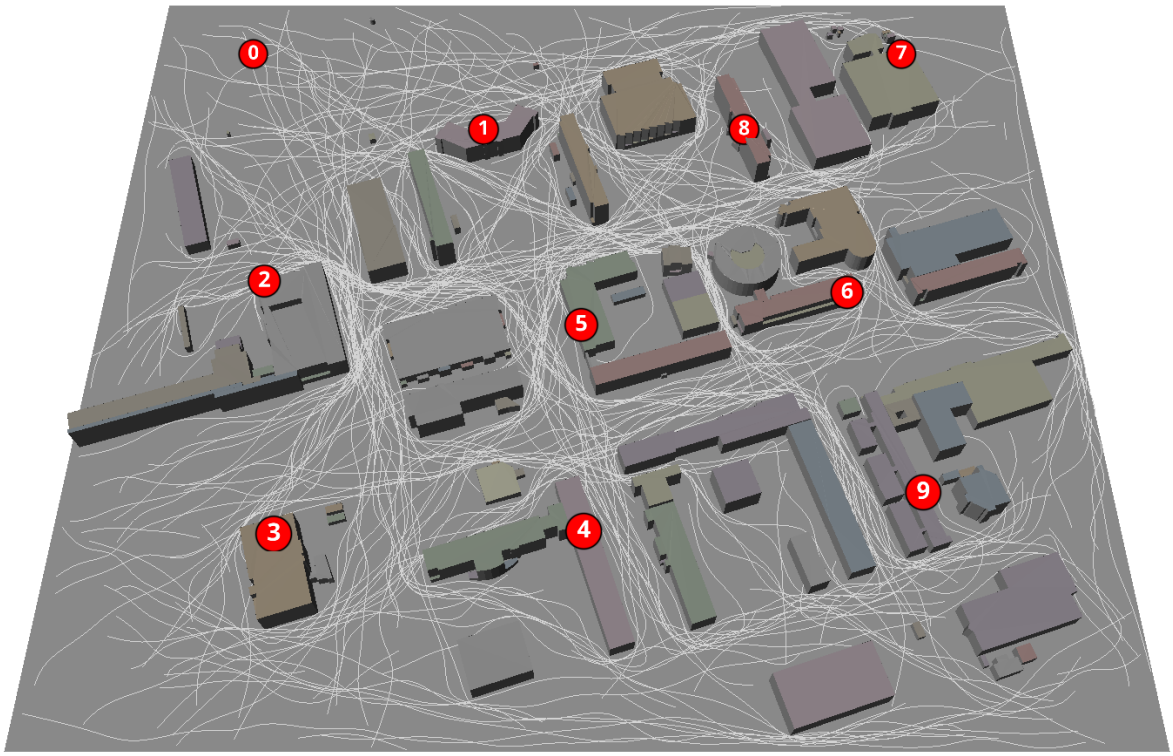


Fig. 1: Aerial view of the scene, showing the BSs (dots) annotated with their indices, and 241 user trajectories (white lines).

A. Scattering Environment

The simulated scattering environment replicates a part of the *La Doua* campus [10] located in Lyon, France. It was created by importing the relevant data from OpenStreetMap [11] into Blender using the Blosm [12] add-on and converting it to a format compatible with Mitsuba [13], the ray-tracing engine used in Sionna [6]; the process follows the steps detailed in the video tutorial [14]. Objects in the scene were assigned materials with specific electromagnetic properties as defined in [15]. Building walls were modeled using `concrete` material properties, roofs were assigned `metal` properties, and the ground, assumed to be a planar surface, was modeled with `medium_dry_ground` properties.

B. Base Stations

1) Antenna arrays: Each BS is equipped with a 2D planar antenna array consisting of 8 antenna elements, arranged in 4 rows and 2 columns (see Figure 2), with half-wavelength (4.3 cm at 3.5 GHz) inter-element spacing in both the vertical and horizontal dimensions. Note that in Sionna, all BSs are constrained to have the same array configuration. Each antenna element supports two polarizations ($\pm 45°$) and has an isotropic radiation pattern. The two orthogonally polarized antennas are assumed to share the same physical location. The positions of the antenna elements are marked in the figure using $\times$ symbols, where each $\times$ represents the location of a dual-polarized antenna.

2) Path-Loss Maps and Coverage Areas: Path loss maps for the whole scene have been pre-computed for each BS using the corresponding functionality of Sionna; the method

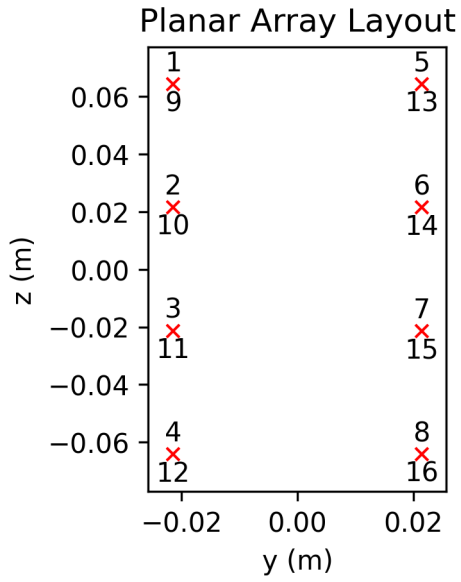
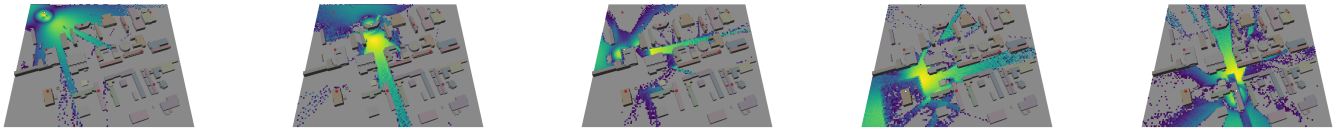
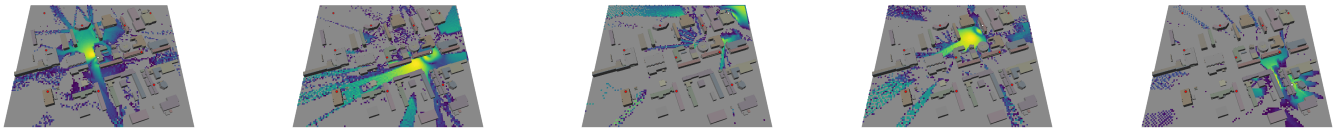


Fig. 2: Array configuration of base stations showing the position of the dual-polarized antenna elements, and the corresponding antenna indices (1-8 for the $-45°$ polarization and 9-16 for the $+45°$ polarization).



(a) Base Station 0 (b) Base Station 1 (c) Base Station 2 (d) Base Station 3 (e) Base Station 4



(f) Base Station 5 (g) Base Station 6 (h) Base Station 7 (i) Base Station 8 (j) Base Station 9

Fig. 3: Path loss maps for different base stations with a threshold value of -100 dB.

consists in dividing the scene into square elements, with an approximate path gain value calculated for each element. By identifying the location of each CSI sample within one of the elements of the path loss map, we can easily access its (approximate) corresponding path gain. This approach allows the a posteriori application of a global path loss threshold that controls the amount of overlap between the respective coverage areas of the BSs: based on the chosen threshold, the multipath components (MPCs) for a BS can be set to zero if its path gain falls below the threshold. This is important because getting the right amount of overlap is critical for the validation of multi-site channel charting methods. As an example of this thresholding operation, Figure 3 depicts the coverage areas obtained by setting the path gain threshold to $-100\,\mathrm{dB}$.

C. Mobile Users

The scenario models the dynamic nature of user mobility through multiple trajectories, designed to reflect typical pedestrian movement. We assume that the mobile user equipment is positioned at a height of $1.5\,\mathrm{m}$ and is equipped with a planar array consisting of a single antenna featuring an isotropic radiation pattern with vertical polarization.

To simulate realistic pedestrian movement, our approach combines probabilistic pathfinding and trajectory smoothing. The Probabilistic Roadmap (PRM) algorithm [16] was employed to generate random yet plausible user trajectories within the region of interest. This method constructs a graph by randomly sampling points within the environment (excluding obstacles such as buildings) and connecting them based on a predefined number of nearest neighbors 15 for our case. These edges represent feasible elementary movements, ensuring that the resulting roadmap reflects the physical constraints of the scene. A trajectory is subsequently derived by finding the shortest path between randomly selected start and end points, as illustrated in Figure 4, which depicts the the constructed graph and a single trajectory using the PRM. A new random graph is generated independently for each trajectory.

The piecewise linear paths generated by the PRM algorithm are not sufficiently realistic models of pedestrian movement. To overcome this limitation, we applied smoothing using third-order cubic spline interpolation, ensuring that the trajectories

Fig. 4: Illustration PRM-based trajectory generation. Random graph, PRM path on the graph and smoothed trajectory .

are continuously differentiable and free of sharp turns. The interpolated trajectories were resampled at $10\,\mathrm{Hz}$ to produce a sequence of time-indexed user positions, approximating continuous movement. Spline interpolation introduces slight variations in instantaneous velocity around a nominal speed of $1\,\mathrm{m\,s^{-1}}$. The histogram in Figure 5 shows the distribution of instantaneous velocities (defined based on the discrete differences between successive samples) in the dataset.

For each interpolated user position along the trajectory, the channel MPCs were simulated using the ray tracing methods. The successive MPCs samples capture the dynamic nature of wireless channels as the user navigates the environment. The dataset consists of 241 simulated trajectories (shown in Figure 1), collectively representing approximately 24 hours of movement. The trajectories vary in length, with travel times ranging from 13 seconds to 13 minutes (average travel time of 6 minutes), providing a diverse set of mobility patterns.

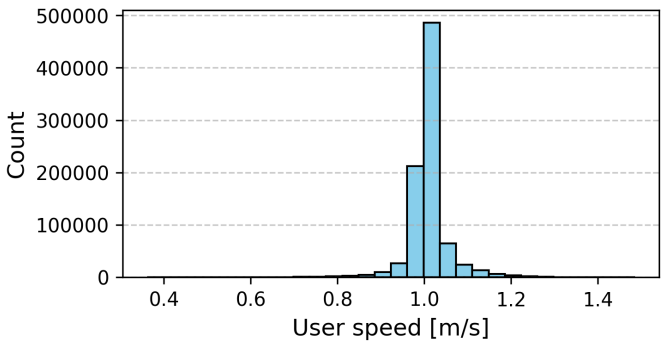


Fig. 5: Histogram of the user speed distribution across the smoothed trajectories (241 trajectory)

D. Radio and Ray Tracing Parameters

The main parameters of the ray-tracing simulation are summarized in Table I. To simulate user mobility in Sionna,

TABLE I: Main Simulation Parameters

Parameter	Value
Central frequency	3.5 GHz
CSI sampling frequency	10 Hz
Total bandwidth	15.36 MHz
Synthetic Array	True
Ray tracer ray candidates (`num_samples`)	10^6
Ray tracer max. no. of bounces (`max_depth`)	5
Ray tracer no. of paths (`num_paths`)	40
Ray tracer scattering	Disabled
Normalize delays	False

we chose to manually move the user between different positions rather than relying on the Doppler interpolation method proposed by Sionna to avoid border effects at the batch boundaries. We also disabled path scattering. Although this choice may reduce the realism of the simulated data, the stochastic nature of scattering, as implemented in Sionna, can introduce unwarranted spatial discontinuities in the model. In addition, accurate simulation of scattering may require precise configuration of the environment's material properties, such as the scattering pattern, to ensure consistent results.

To efficiently handle large antenna arrays, we use the synthetic array option, which significantly reduces memory usage and computational complexity. This method simplifies the ray tracing step by considering only the center of the array, with phase shifts computed to obtain channel coefficients for the other antennas under the assumption of a planar wave. We also limit the maximum number of interactions per path to 5 in the scene. We retain 40 paths in the MPCs, selecting the strongest if there are more than 40, and padding with special values if there are fewer.

III. Dataset Structure and Usage

The dataset is stored as a single file using the Hierarchical Data Format version 5 (HDF5) format [17]. It contains different groups, each group containing all samples from a given user trajectory. A trajectory (see Section II-C) consists of a sequence of geolocated CSI samples. To reduce storage space, we store the MPC representation of the channel state; the frequency-domain representation can be dynamically generated from the Fourier transform of the MPC using the `cir_to_ofdm_channel` Sionna primitive.

For each trajectory (group in the HDF5 terminology), the following keys (datasets in the HDF5 terminology) are available in the HDF5 file :

- `timestamps`: simulated time (in seconds) since the beginning of the trajectory
- `positions`: the user's 3D position along the trajectory (local coordinates, in meters)
- `velocities`: the user's velocity (defined as the discrete derivative of the position) at each point along the trajectory
- `a`: MPC baseband channel coefficients between the different antennas of the BS and the user at each point along the trajectory.
- `tau`: MPC delays (in seconds) between the center of the antenna arrays at each point along the trajectory.

The dataset is distributed with the source code and key components required to reproduce the scenario, including BS locations and user trajectories. The Git repository [8] provides Python scripts for data generation and visualization. These scripts demonstrate how to generate, read, and visualize the data, including frequency-domain CSI, user positions, and path loss maps. Key scripts and their features include:

- `coverage_map_creation.py`: computes and visualizes path loss maps, as seen in Fig. 3.
- `data_generator.py`: generates and stores the data in a HDF5 file.
- `visualize_trajectories_on_scene.py`: 3D visualization of the simulated environment and user trajectories (used for Fig. 1).
- `plot_frequency_csi.py`: Demonstrates access to frequency-domain CSI, visualizes user positions on the path loss map and the CSI amplitudes for different BS and antennas with an interactive slider to explore trajectories (see Fig. 6).

IV. Channel Charting using the MOCSID dataset

A. Experimental setup

1) Data pre-processing: For our application of interest, Channel Charting, we utilized a subset of the MOCSID dataset to train a triplet-based neural network (NN) [18], with a coherence time $T_C = 2$ s. A total of 50 randomly selected trajectories were used to train and evaluate the NN model. Specifically, 45 trajectories were selected for training, and the remaining 5 trajectories were used for testing. The training trajectories included: 26, 9578, 5105, 9756, 15, 8501, 1610, 9554, 9709, 2135, 2922, 8138, 6074, 1223, 7669, 3470, 4480, 1788, 2494, 6944, 3848, 6016, 2617, 107, 4581, 5784, 4382, 2514, 7771, 6947, 7590, 5255, 9716, 5239, 2943, 7072, 7782, 5681, 3939, 8713, 102, 7163, 1668, 5497, and 7958. The test trajectories were: 1054, 1514, 7287, 5826, and 8252.

For each sample, the MPCs were transformed into a frequency-domain representation with 256 subcarriers using the `cir_to_ofdm_channel` Sionna primitive. We do not normalize the delays or the energy per OFDM resource grid,

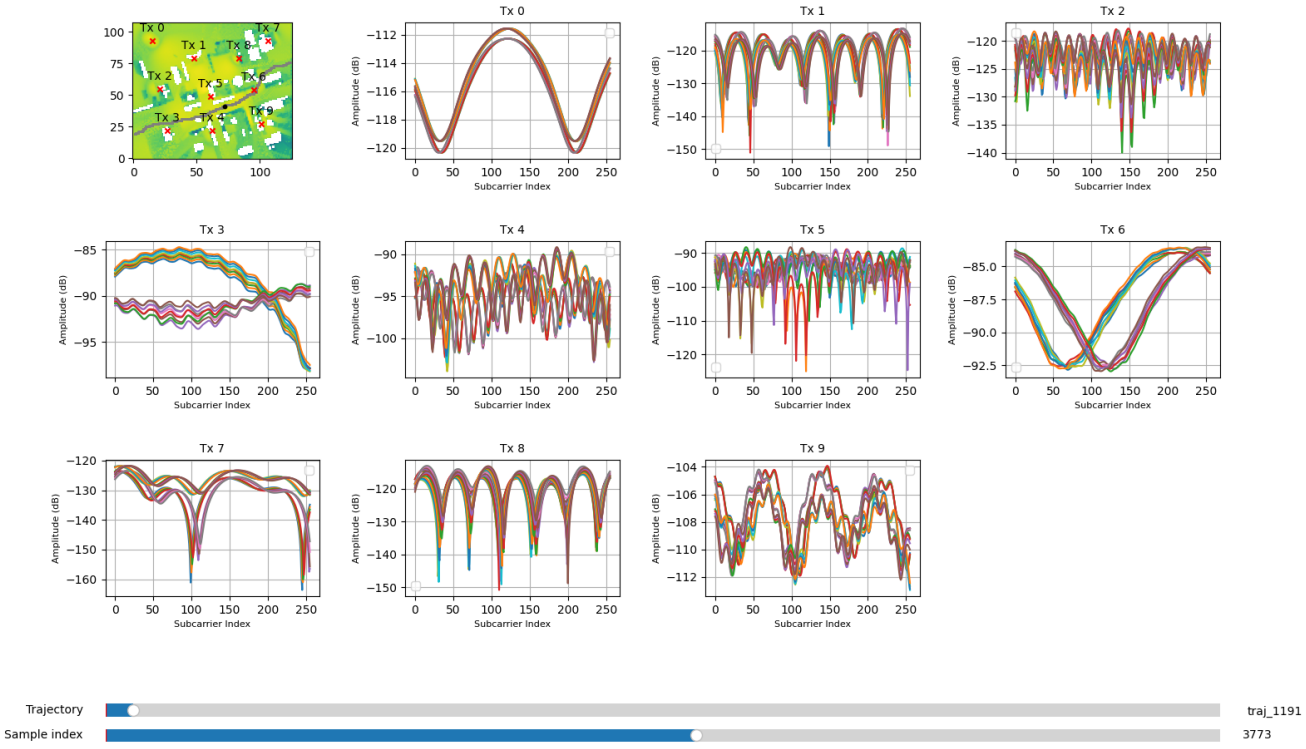


Fig. 6: Visualization interface showing user location, amplitude values (in dB) for antennas from different base stations.

preserving the natural variations in path delays and power levels as they occur in realistic channel conditions. The resulting frequency-domain CSI was then pre-processed using the approach proposed in [19]. Finally, the feature vectors from all BSs were concatenated to construct the final feature vector for each sample. However, not all sample positions are reachable by every BS, due to the respective BS coverage areas (see Figure 3), resulting in missing data. To address this issue, we employed mean imputation. Specifically, we calculated the mean feature vector for each BS based on the samples in the training set (using the non-missing data). Whenever the feature vector for a BS was missing for a given sample, it was completed using the pre-computed mean vector.

2) Training architecture and parameters: The NN architecture is based on the design presented in [18], with a slight modification: the first fully connected (FC) layer has 1280 input features and 512 output features. Each subsequent FC layer reduces the number of output features to half of its input features, except for the final layer, which outputs two features. The model was trained using ADAM optimizer for 20 epochs with a batch size of 1024, using a triplet loss function with a margin of 1. The optimization process employed a learning rate of 10^{-2} and a weight decay of 10^{-7}. Finally, due to the substantial computational resources required to train and evaluate the model, this experiment was conducted only once

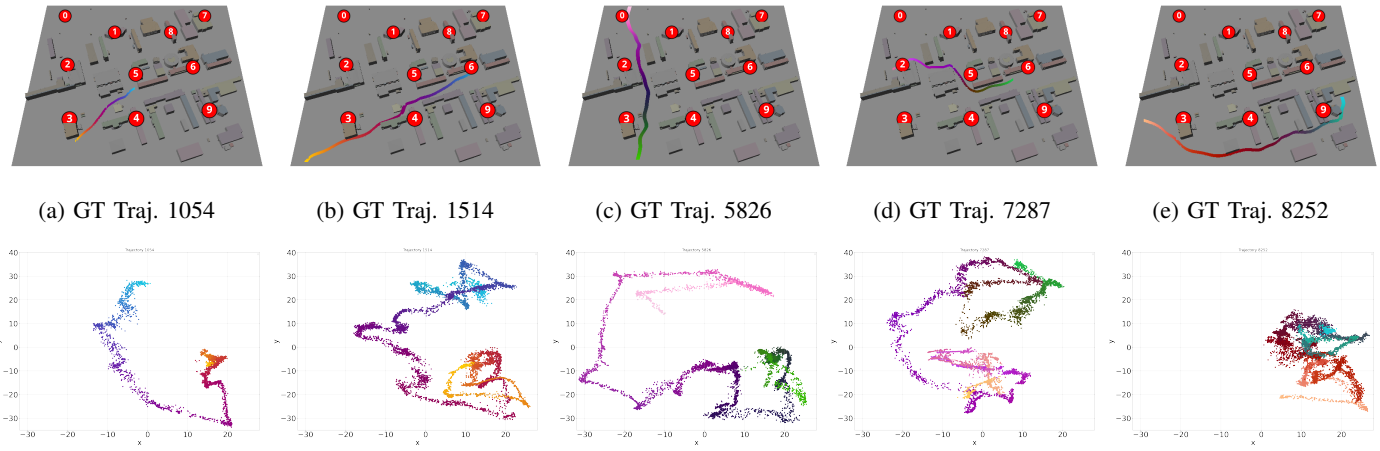
3) Evaluation metrics: In contrast with localization tasks which are supervised, aiming to estimate actual coordinates from CSI and evaluated using metrics like mean absolute localization error, channel charting is a self-supervised approach which can only be expected to generate a latent-space representation which respects the topology of the predominant parameters which explain the data (here, the mobile user position). Hence we need to evaluate the performance of the obtained CC using evaluation criteria specific to dimensionality reduction problems. Specifically, we used the trustworthiness (T), continuity (C), and Kruskal stress (KS) metrics, comparing the CC to the 2D positions where each CSI sample was acquired (though these positions are typically unavailable in practice). Trustworthiness and continuity quantify neighborhood preservation, while Kruskal stress measures the preservation of distances. For additional details, we refer the reader to [18].

B. Results

The evaluation results are presented in Table II, while the CC for each test trajectory is illustrated in Fig. 7. From Tab. II, we observe that overall, the triplet-based model achieves reasonable performance in terms of neighborhood and distance preservation, validating the use of the MOCSID dataset for channel charting. This observation is further supported by the visual representation of the channel charts in Fig. 7.

V. Conclusion

In this work, we introduced MOCSID, an outdoor synthetic dataset designed to represent a real-world campus scenario. We provided detailed simulation parameters and illustrative investigation examples to enable and facilitate the benchmarking of advanced channel charting techniques. The relevance of



(a) GT Traj. 1054 (b) GT Traj. 1514 (c) GT Traj. 5826 (d) GT Traj. 7287 (e) GT Traj. 8252

(f) CC Traj. 1054 (g) CC Traj. 1514 (h) CC Traj. 5826 (i) CC Traj. 7287 (j) CC Traj. 8252

Fig. 7: Ground truth locations (top) and corresponding channel charts (bottom) for the samples of the different test trajectories.

TABLE II: Model performance based on the test trajectory.

Trajectory ID	**T**(%) ↑	**C**(%) ↑	**KS** ↓
1054	98.00	98.60	0.47
1514	97.47	99.17	0.43
7287	96.94	99.28	0.47
5826	98.27	99.10	0.49
8252	95.32	98.46	0.42
All	92.00	98.96	0.46

MOCSID extends to applications such as wireless communication optimization, localization, signal quality prediction, and interference management. By making both the dataset and its generation source code publicly available, this work aims to support further advancements in channel charting and other data-driven approaches leveraging CSI data.

Acknowledgment

This work was supported by ANR (grant ANR-23-CHR4-0001-01) under the CHIST-ERA project CHASER (CHIST-ERA-22-WAI-01), and by Horizon Europe SNS project INSTINCT (grant 101139161). It was performed using HPC resources from GENCI-IDRIS (Grant 2023-AD010614930).

References

[1] C. Studer, S. Medjkouh, E. Gonultaş, T. Goldstein, and O. Tirkkonen, "Channel charting: Locating users within the radio environment using channel state information," *IEEE Access*, vol. 6, pp. 47 682–47 698, 2018.

[2] P. Ferrand, M. Guillaud, C. Studer, and O. Tirkkonen, "Wireless channel charting: Theory, practice, and applications," *IEEE Communications Magazine*, vol. 61, no. 6, pp. 124–130, 2023.

[3] F. Euchner, M. Gauger, S. Dörner, and S. ten Brink, "A Distributed Massive MIMO Channel Sounder for "Big CSI Data"-driven Machine Learning," in *International Workshop on Smart Antennas (WSA)*, 2021.

[4] A. Colpaert, C. Thys, Z. Cui, and S. Pollin. (2023) MaMIMO-UAV 3D Channel State Information Dataset. [Online]. Available: https://doi.org/10.48804/0IMQDF

[5] A. Alkhateeb, "DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications," in *Proc. of Information Theory and Applications Workshop (ITA)*, San Diego, CA, Feb 2019.

[6] NVIDIA Corporation. (2024) Sionna: An open-source library for next-generation physical layer research. Accessed: 2024-07-22. [Online]. Available: https://github.com/NVlabs/sionna

[7] G. G. Anagnostopoulos, P. Barsocchi, A. Crivello, C. Pendão, I. Silva, and J. Torres-Sospedra, "ORDIP: Principle, practice and guidelines for open research data in indoor positioning," *Internet of Things*, 2025.

[8] "MOCSID Gitlab repository," Online, https://gitlab.inria.fr/channelcharting/outdoor_dataset.

[9] M. E. M. Makhlouf, M. Guillaud, and Y. Vindas. (2024, Dec.) Multi-cell outdoors channel state information dataset (MOCSID). [Online]. Available: https://doi.org/10.5281/zenodo.14535165

[10] "La Doua campus on OpenStreetMap." [Online]. Available: https://www.openstreetmap.org/\#map=18/45.783286/4.871841

[11] OpenStreetMap contributors, "Openstreetmap data," https://www.openstreetmap.org.

[12] Prochitecture, "Blosm for blender." [Online]. Available: https://prochitecture.gumroad.com/l/blender-osm

[13] "Mitsuba 3." [Online]. Available: https://www.mitsuba-renderer.org/

[14] F. Ait Aoudia, "Sionna RT: Scene Creation with Blender using OpenStreetMap (YouTube)," 2023. [Online]. Available: https://www.youtube.com/watch?v=7xHLDxUaQ7c

[15] ITU-R, "Effects of building materials and structures on radiowave propagation above about 100 MHz," International Telecommunication Union, Tech. Rep. Recommendation ITU-R P.2040-2, 2020.

[16] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[17] The HDF Group. Hierarchical Data Format, version 5. [Online]. Available: https://github.com/HDFGroup/hdf5

[18] P. Ferrand, A. Decurninge, L. G. Ordoñez, and M. Guillaud, "Triplet-based wireless channel charting: Architecture and experiments," *IEEE Journ. Sel. Areas in Comm.*, vol. 39, no. 8, pp. 2361–2373, 2021.

[19] S. Taner, V. Palhares, and C. Studer, "Channel charting in real-world coordinates," in *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, 2023, pp. 3940–3946.