2024 IEEE International Conference on Metaverse Computing, Networking, and Applications (MetaCom) | 979-8-3315-1599-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/MetaCom62920.2024.00031

DITTO: DIgital Twins for Testing and Optimizing Wireless Decisions

Richard Kumahia*, Utku Demir†, Suyash Pradhan‡, Batool Salehihikouei§,Kaushik Chowdhury¶ and Stratis Ioannidis‖
Institute for the Wireless Internet of Things
Northeastern University, Boston, Massachusetts, USA
*Kumahia.r@northeastern.edu, †udemir@ece.neu.edu, ‡pradhan.suy@northeastern.edu,
§salehihikouei.b@northeastern.edu, ¶krc@ece.neu.edu, ‖ioannidis@ece.neu.edu

***Abstract*—Digital Twins (DTs) are powerful tools for decision making that mirror real-world systems and continuous interactions between them. We study on the application and capabilities of DTs in the realm of wireless communications, using two leading wireless communication tools: Wireless InSite and Sionna. Specifically, we compare the fidelity of the two wireless communication tools by measuring several metrics with a real-life dataset. Our comprehensive analysis aims to determine the capabilities of Digital Twins in the context of wireless communications, offering valuable insights for future researchers in the field.**

***Index Terms*—digital twin, automation, tool, performance**

I. Introduction

A *digital twin* (DT) [1] is a replica of a physical system in a virtual environment. DTs are employed in applications such as modeling human-space-device interaction, the product life cycle, and detecting abnormal activity, in areas such aviation and healthcare [2]. Their popularity has led their way into the wireless applications as well [3], including 5G automation [4], beam selection [5], generative artificial intelligence (AI) [6], and reflective intelligent surfaces (RIS) [7]. Despite this, the analysis of DT tools for wireless applications is still in its infancy. There are several wireless tools available to create simulated replicas of the real world, but their capabilities not only vary but are also largely unknown prior to testing, requiring valuable time and effort.

We explore the capabilities of the two most recent and prominent tools to create a DT in the wireless field, Wireless InSite (WI) [8] and Sionna [9]. WI and Sionna are wireless simulation tools to construct a DT of a wireless system deployment, as they can imitate indoor and outdoor plans of buildings, objects (e.g., vehicles and pedestrians), and reflective material properties. As ray-tracing tools, they can emulate a wireless DT precisely, because they simulate wireless channels, taking various variables into account, such as electromagnetic wave propagation properties, communication signal types, and antenna patterns. WI and Sionna also offer different options: e.g., as shown in Table I, WI provides different wireless communication standards (LTE, WiMAX, 802.11n/ac), while Sionna follows the 5G-NR standard.

The authors gratefully acknowledge support from the National Science Foundation through grant 2112471.

Simulation parameters within the DTs need to be entered and modified so that the DT simulates its real world counterpart as close as possible. Therefore, the ability to realize object sizes, material properties, environment features carry utmost importance. Similarly, the ability to capture interactions among entities (e.g., obstructions, object movements, etc.) is important in ensuring DT fidelity [10]. Creating detailed maps or other representations of the physical world in a DT is a tedious and computationally intensive task, and the lack of automation in virtual world creation limits DT scalability.

We explore these issues by creating multiple DTs (i.e., virtual representations of the real world with different fidelity levels) and rigorously evaluating WI and Sionna. First, we create virtual worlds, by importing maps that include buildings and reflective material properties. We facilitate the virtual world creation in Blender [11], assisting researchers in creating DTs by entering the geographical coordinates of the area of interest. This reduces tedious map creation efforts and allows researchers to direct their focus to wireless experimentation and development. Second, we set the wireless components, such as antennas, signal waveforms, signal carrier frequency, electromagnetic properties of the reflective materials, channel characteristics, transmitter (Tx) and receiver (Rx) locations, ray reflective properties, by mimicking their real-life values. Third, we collect a large dataset using the available output options in each simulation tool, such as received power, ray tracing, and SNR. Finally, in the evaluation step, we not only measure the performance of the DTs created by the two simulators, but also test the DT results using the FLASH dataset [12], a real life testbed that we followed to create its DTs in WI and Sionna. This way, we demonstrate the similarity between the wireless DT and its real-world counterpart. Our contributions are as follows:

- We create multiple DTs in Sionna for wireless experiments in order to allow researchers to bypass the tedious virtual world creation by simply entering the map coordinates of interest.
- We explore different levels of fidelity, establishing tradeoffs among map detail, accuracy and consistency in wireless outputs, and required computation.
- We evaluate our results using the FLASH dataset [12]

979-8-3315-1599-7/24/$31.00 ©2024 IEEE
DOI 10.1109/MetaCom62920.2024.00031

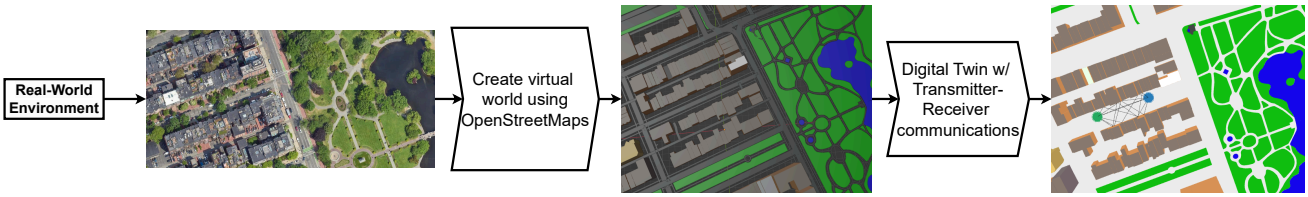


Fig. 1: We compare digital twins and how closely they perform to compared to the real world. Our approach to Digital Twins for Testing and Optimizing Decisions (DITTO) involves taking a real-life deployment of wireless transmitters, embedding them onto a 3D representation of the environment, using a raytracing framework, and simulating the transmissions.

Feature	Wireless InSite	Sionna
Reflections	✓	✓
Diffractions	✓	✓
Transmissions	✓	✓
Custom antenna	✓	✓
Material choices	Concrete, asphalt, metal, glass	Concrete, metal, glass
Available frequencies	$0.1 - 100 \times 10^9 Hz$	No limit on frequency range
Available standards	LTE, WiMAX, 802.11n, 802.11ac	5G-NR
Tx and Rx options	Locations, alignment, Tx Power	Locations, alignment
Map	Terrain, coordinates	Terrain, coordinates
Trainable coefficients		✓
Antenna options	Antenna gain, MIMO, built-in, and custom antennas	Antenna gain, dipole, HW dipole, Custom patterns, Polarization Models, Slant angles

TABLE I: Comparing features of Wireless InSite and Sionna.

and quantitatively evaluate the parallels between WI and Sionna outputs.

- Our work provides guidelines as to which level of detail and type of twin to select based on potential needs.

The remainder of the paper is organized as follows. In Sec. II we provide a summary of prior works in DT. Sec. III details the functionality of the DT tools, WI and Sionna. Sec. IV delves further into how we developed our experiment pipeline and evaluation metrics, which we rigorously test and discuss in Sec. V. Sec. VI concludes the paper.

II. Related Work

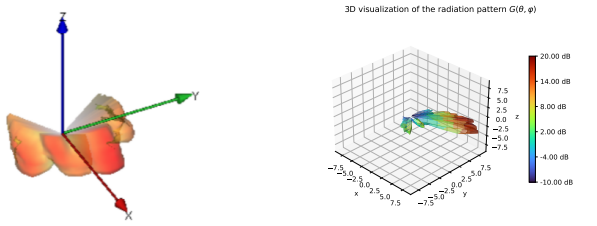
Digital Twins for Wireless Beamforming. Salehi *et al.* [5] proposed the "Multiverse-at-the-Edge" paradigm to mirror beam-forming via DTs with varying degrees of fidelity. Deep learning (DL) and convolutional neural networks (CNNs) were utilized to predict the most accurate beam in the communications between an autonomous vehicle and roadside basestation (BS). The time to select the best beam at the BS was shortened significantly compared to the preexisting exhaustive-sweep approach in the IEEE 802.11ad and 5G-New Radio (NR). In environments not previously encountered, the accuracy of these models significantly decreases, leading to the need for DTs to emulate radio frequency (RF) propagation patterns. The novelty of these authors' work is by offering DTs of varying number of beam reflections to mitigate the loss of accuracy in DL predictions while still maintaining a performance increase over exhaustive sweeps. Li *et al.* [13] propose a DT-aided learning framework that leverages the QUAsi Deterministic RadIo channel GenerAtor (QuaDRiGa) platform and 5G NR standards to simulate real-world frequency division duplex (FDD) systems, generating significant data to training a deep generative model. This approach addresses the challenge of acquiring channel probability distribution information, especially in 5G NR FDD systems with limited feedback.

Optimization of Wireless Access Point Deployment. Abdulwahid *et al.* [14] present an investigation and optimization method for the deployment of wireless access points (APs) in indoor networks. They aim to enhance coverage and network performance by strategically placing APs using propagation modeling and optimization algorithms. By simulating wireless signal behavior in indoor environments and considering obstacles like walls and furniture, they identify optimal AP locations to improve coverage and reduce interference. This work resembles our research by highlighting the importance of optimization in wireless network performance. While Abdulwahid *et al.* focus on indoor environments, we leverage digital twins to test and optimize wireless decisions in outdoor environments, specifically for autonomous vehicle use cases.

Digital Twins for Vehicle-to-Cloud Communication. Liao *et al.* [15] exhibit a practical application of DTs in wireless communications is ramp merging in highways. The use of vehicle-to-everything (V2X) communications allows vehicles, and the drivers operating them, to efficiently plan their next actions on the highway without endangering others. Liao *et al.* propose an approach to map virtual copies of real vehicles to merging lanes, helping connected vehicles adjust their position and speed in advance. Their contributions were the implementation and real world experimentation of DTs in connected and autonomous vehicle systems. The authors found that their implementation improved on the safety and environmental sustainability of current ramp merging systems. Wang *et al.* propose a DT framework for connected vehicles, utilizing V2C communication to synchronize real-world activities with virtual counterparts. This framework operates within a two-layer structure, with the physical layer representing the real-world entities and interactions, and the cyber layer handling computational tasks and data processing [16]. Using a case study of a cooperative ramp merging scenario, the DT benefits the transportation systems with sufficient performance.

III. Implementation

Our pipeline is shown in Figure 3. We enter the minimum and maximum latitudes and longitudes of the real world environments coordinates into OpenStreetMap (OSM). The 3D map is then imported to Blender, where we add obstacles and export the map as a Mitsuba XML file. We then read the XML file using Wireless Insite (WI) and Sionna, and enter the



(a) WI Antenna Pattern Example (b) Sionna Antenna Pattern Example

Fig. 2: Visualizations of one of the antenna patterns simulated by WI (left) and Sionna (right). The patterns show which direction the antenna's gain increases and decreases.

coordinates of the transmitter (Tx) and receiver (Rx) as well as their antenna gain patterns. We then simulate the Tx-Rx communications and collect the simulated received powers.

A. Generation of Map & 3D Environment

A digital twin world creation starts with 3D representations of the environment. To represent a given geographical location, we use OpenStreetMaps [17] (OSM) to provide a detailed and customizable map. We enter the minimum and maximum latitude and longitude in a Geographic Information System (GIS) of the desired geographic location (geo-location). OSM returns 3D objects representing the buildings, streets, sidewalks, railways and parks in the specified geo-location. We apply these 3D objects in Blender [11], a 3D computer graphics software used for animations. We use Blender-OSM [18], an extension of Blender, that integrates Blender with OSM so that we can further manipulate the environment. Blender offers the flexibility to modify existing structures and add custom elements as needed. We alter the materials of the buildings to be compatible with the wireless communication simulators. We assign the building materials as *concrete*, which more accurately represents the reflective and absorption properties of the building materials. The material of an object will determine its permittivity, conductivity, frequency range and scattering coefficient during the simulation. Moreover specific scenarios like non-line-of-sight (NLOS) conditions are simulated by manually placing obstacles within the digital twin environment. We then export Blender OSM's virtual map to Sionna as an XML file using Mitsuba 3, a rendering tool.

B. Sionna

1) Inputs: With the environment as an XML file, Sionna is run in a Python or Jupyter Notebook program and use the XML file as an input. Jupyter notebook allows the user to preview the simulation environment. The XML file contains the virtual environment's material and location information. The transmitter (Tx) and noise power differ from WI as they are not directly set by the user, but calculated by the gains of the Tx and receiver (Rx) antenna patterns for a range of zenith and azimuth angles. Sionna has four predefined antenna patterns: isotropic, dipole, half-wavelength, and a pattern based on the 3GPP TR 38.901 channel model, a stochastic channel model for designing 5G systems. We have implemented the Talon legacy antenna patterns using the measurements saved in Seemoo Lab's repository [19]. The Tx and Rx locations and orientations can be set by the user. For our purposes, we set these parameters to match the FLASH environment.

2) Processing: Once the environment and Tx/Rx locations are set, Sionna conducts a series of processing steps, primarily utilizing ray tracing techniques. First, Sionna uses the antenna patterns set by the user to calculate the possible rays between the Rx and Tx. It calculates the propagations through the environment, incorporating reflections, diffraction, and scattering effects based on the environment materials and antenna patterns. It then returns the path loss, which we add the noise and Tx power to calculate the full received power.

3) Outputs: Sionna's outputs are the ray propagation paths and angles of arrival and departure. Sionna does not save the reflection points of the rays. The *power related* outputs of Sionna are the passband coefficient a_i and the channel impulse response. We use the passband coefficients and channel impulse response to calculate the received power and the other power related outputs. The *time related outputs* of Sionna is solely the propagation delay of the ray. To compare with WI, we calculate the Signal to Noise Ratio (SNR). We do this by calculating the path loss of the beam from the passband coefficient and channel impulse response. We then add the path loss to the noise power of the environment and the Tx power. The SNR is the sum of the three variables.

C. Wireless InSite

1) Inputs: WI has an extensive list of wireless properties that gives users flexibility and freedom to conduct a wide range of realistic and detailed wireless experiments. Efficient wireless communication initiates from precise antenna patterns, which WI provides through built-in and customizable antenna options. Users are able to set antenna gain, directivity (E and H-plane half powers) in single or MIMO antenna configurations. High granularity of ray spacing, i.e. no limit on how small the angle is between two adjacent emitted rays from transmitters, allows for precise ray tracing results.Users are free to choose any Transmitter (Tx) and Receiver (Rx) location on the map, where they can realize any antenna alignment.

One of the distinguishing properties of WI is that the tool provides the selection of wireless standards, which includes, 802.11ac, ax, WiMax and LTE as well as propagating signal waveform, frequency, and phase. Setting the Tx power and levels of interference and noise allows for imitating real life power measurements. Lastly, electromagnetic (EM) features of rays and reflective surfaces could be set for proper emulation. These EM features include the number of allowed ray reflection, diffraction, and transmission (penetration through obstacles) before a ray reaches to Rx from Tx as well as channel propagation characteristics, e.g. open air and canyon.

2) Processing: The computation power of the computer on which WI is running does not affect the wireless related outputs, e.g. received power, strongest rays, reflection points.

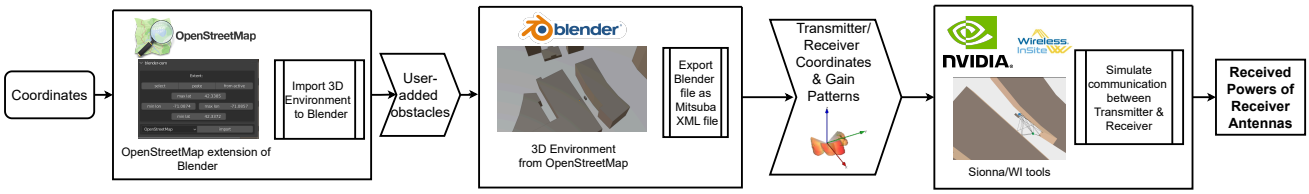


Fig. 3: Pipeline of our experiment process. We enter the minimum and maximum of the real-world environment's coordinates into (OSM). We import the map to Blender, and add obstacles in Blender. We export the map as a Mitsuba XML file. We read the XML file using WI and Sionna, and enter the Tx and Rx and their antenna gain patterns. We then simulate the Tx-Rx communications and record the simulated received powers.

The computation power only affects the time it takes to obtain ray tracing results.

3) Outputs: WI has an extensive wireless output options, which we group under three categories, namely *ray*, *power*, and *time related*. All the outputs can be observed both in the WI GUI and can be downloaded as .txt files. *Ray related* outputs include ray propagation paths, angle of arrivals and departures, and ray reflection points. *Power related* outputs include received power, rssi, bit-error-rate, throughput, pathloss, and SNR, SIR, and SINR given that the user has an option to select noise and interference levels. *Time related* outputs include ray time of arrival, delay spread, power delay profile, and experiment computation time.

IV. Experiment Setup

We import a 3D map from the FLASH experiment area, and generate DTs as in Section III. Following the environments created in [5], we also generate an open field, to act as our lowest fidelity DT. We use different metrics to evaluate the accuracy of our simulations: Magnitude Difference, Cosine Similarity, Concordance Index, Normalized Discounted Cumulative Gain, and a unique metric developed by Salehi *et al.* [5] (see Sec. IV-D). In this section, we elaborate on these metrics and delve deeper into our DT generation.

A. Real World Dataset

The real world dataset, FLASH, is drawn from [12], [20], which was collected in four experiment setups that the authors call *categories*. Each category consists of 10 episodes, i.e. experiments conducted, in which an autonomous car with camera, LiDAR, and GPS sensors communicates with Tx deployed at the edge of a road (i.e. the Rx is on the car) using the 802.11ad standard, i.e. $60GHz$ signals (mmWave), while it is passing by the Tx. The data is collected in an urban environment of concrete buildings with a partial glass exterior. Given that the system is a simple 802.11ad network, the Rx at the autonomous car sweeps for the best beam (antenna pattern) out of the 34 available beams so that the optimal communication takes place between the car and the Tx. *The portion of the dataset we utilized is the ranking of the candidate beams by the GPS location.*

The categories in the FLASH dataset are grouped based on the line-of-sight (LOS) availability and realistic obstacles in between the Tx and the Rx on the autonomous car. Accordingly, the data environment setup for the categories are as follows: *Category-1*: LOS, *Category-2*: pedestrian standing between the Tx and Rx, hence non-line-of-sight (NLOS), *Category-3*: NLOS, with a stationary car obstacle between Tx and Rx, and finally *Category-4*: NLOS, with a stationary car obstacle between Tx and Rx. For all the experiments different vehicle and obstacle speeds, ranging from $10-20\,mph$, and obstacle movement directions are used, totalling 210 episodes with $\sim$32K sample points. Further details on the real dataset could be found in [12]. In our DT evaluations we used 1 episode from the categories 1 and 3.

B. Synthetic Datasets

For our experiments, we generated synthetic datasets to emulate real-world scenarios encountered in wireless communication environments, inspired by a similar dataset collected in the study [5], which was created using WI. We replicated these scenarios in Sionna, eventually creating the categories 1 and 3 in [12]. In the FLASH dataset, certain points in the experiment do not have received power information for every beam emitted by the transmitter. This could be due to certain beams not reaching the receiver within a certain timeframe, or not reaching the receiver at all. Because of this, we adopt two conventions/approaches to address this issue when comparing FLASH to simulated datasets. In our first approach, which we call a *restricted*, we remove synthetic beams that were missing in the real world dataset from the simulated data. In the second, *unrestricted* approach, we fill the missing beam patterns with the value 0. By doing this, we simply are assuming the received power is the lowest possible value in FLASH. When comparing FLASH to DTs, we report metrics for both settings.

For the WI dataset, the same propagating signal types (carrier frequency, $60\,GHz$) as the FLASH dataset was used. Also the same reflective surface materials are used, i.e. asphalt on the road and concrete on the buildings. NLOS scenarios in WI was created through a metallic box that represents the car obstacle in category 3 in the FLASH dataset. In the WI dataset, three twins are defined: *Baseline*, *1-Reflection*, and *3-Reflection*. The wireless settings are the same between scenarios, only the number of reflections change. The Baseline twin was created in an open area, where the reflectors are distanced far away in order to assess the twin performance for the simplest case. In the Baseline twin, only 1 sample

point used where the 34 beams were swept, parallel to the FLASH dataset. 1-Reflection and 3-Reflection twins are in the same urban canyon environment that simulates the FLASH dataset environment, where the only difference is the number of reflection allowed. In these twins there are 200 equidistant sample points on the $\sim 40\,m$ trajectory of the simulated autonomous car, where 34 beams were swept at each sample point. In short, there are total of 6 scenarios (3 twins and their LOS/NLOS cases). Each of these scenarios corresponds to a single episode in the real world dataset. The purpose of introducing multiple twins was to evaluate whether the computation burden worth the accuracy of beam selection.

To have an unbiased assessment, we replicated the scenarios given in [5] in Sionna, selecting the same signal properties of the WI dataset and materials, metal, concrete and glass. The vehicle traveled down an alley situated between a building and a parking garage with a Rx. The datasets encompassed two distinct scenarios, each representing category 1 and 3 in [12]. We collected an 6 scenarios for each category to represent the selected scenarios in Sionna, to match WI's three twins and their LOS/NLOS scenarios. Due to having access to Sionna for a longer time, we were also able to collect every episode from categories 1 and 3 from FLASH and simulate them in Sionna. For this extra data, we solely compare Sionna to FLASH.

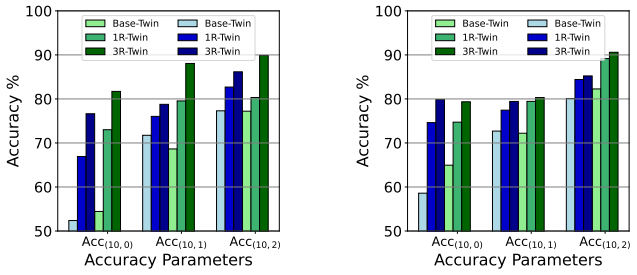
C. Automating DT creation

The DT generated using WI was automated using a Python script interfacing with the software's graphical user interface (GUI). The simulation in WI involved modeling the transmission of data from the base station to the vehicle using a set of 34 different antenna patterns. The received power of each beam transmission was recorded, and the optimal beam pattern for communication was determined based on the received power. The DT generated with Sionna was achieved through scripting a Python program to interact directly with Sionna's API. We utilized the antenna beam capabilities provided by Talon Tools for the DTs.

D. Performance Comparison Metrics

To evaluate the performance of our DTs, we quantitatively evaluate received signal power and how it affects the ranking of the 34 beam candidates to choose for the optimal communication between the Tx and Rx. We have utilized several different metrics towards that end, including Accuracy (Acc), Magnitude Difference (MD), Cosine Similarity, Concordance Index (CI) and Normalized Discounted Cumulative Gain (NDCG), described in detail below. We also visually inspect the 3D antenna similarity between the DT as well as the real world Talon antennas Fig. 2).

Accuracy. Salehi *et al.* [5] quantify the accuracy of a simulation relative to the real world ground truth via notion of relaxed accuracy. For a DT and the corresponding real-life counterpart in a scenario, let x, y represent the vectors of received powers of each receiver antenna in dB in the two settings. Then, given a set of scenarios, the relaxed accuracy $\text{Acc}(K,T)$ contains the fraction of scenarios in which the top-K beams in simulated



(a) LOS scenario (b) NLOS scenario

Fig. 4: Relaxed accuracy $\text{ACC}(K,T)$, comparing received powers from Sionna (shades of green) and WI (shades of blue) map to the real world's data. We use three DTs of increasing fidelity for WI and Sionna and determine how accurately the top beams map to the real world. There are three different thresholds used in our experiments (0, 1, and 2 dB) to determine if the top beams fall within them.

vector x contain a beam within T dB from the optimal beam in the ground truth vector y. We experiment with this metric for different values of K and T.

Magnitude Difference. MD is the Euclidian distance of the received powers of the antennas between the two simulations, i.e., $\text{MD} = \|x-y\|$, where x, y represent the received powers of each receiver antenna in dB in two settings.

Cosine Similarity. CS measures the cosine of the angle between the two vectors, i.e., $\text{CS} = \frac{x\cdot y}{\|x\|\|y\|}$, ranging from 1 (completely aligned) to -1 (aligned but the vectors are in opposite directions): This measures the similarity of the received powers' orientations ignoring differences in scaling.

Concordance Index. The CI measures how well a model's predicted outcomes align with the actual outcomes in pairwise rankings. Formally, $\text{CI} = \frac{\sum_{i=1}^{d}\sum_{j=i+1}^{d}\mathbb{1}_{x_j<x_i}\cdot\mathbb{1}_{y_i<y_j}}{d(d-1)/2}$, where $d=34$ is the vector dimension (number of beams). Intuitively, CI is the ratio of the number of concordant pairs and divided by the total number of pairs. A score of 1 indicates perfect concordance per pair, while 0 indicates a full order reversal.

Normalized Discounted Cumulative Gain. NDCG captures both ranking importance and gain magnitudes, focusing more on agreement on larger beams. Formally, the discounted cumulative gain (DCG) of x w.r.t. y is given by: $f(x,y) = \sum_{i=1}^{n}\frac{x_{(i)}}{\log_2(i+1)}$ where $x_{(1)}$, $x_{(2)}$, ..., $x_{(n)}$ is an indexing of the coordinates of x w.r.t. the order induced by y (that is, (1) is the index of the highest gain in y, (2) is the index of the second highest gain, etc.). The NDCG is defined as $\text{NDCG} = \frac{f(x,y)}{f(x,x)}$. Perfect alignment yields a score of 1. Missaligning more powerful beams lowers scores more, due to the log term.

V. Results

We present here the results of the experiments described in Section IV. We first compare between pairs, i.e., Sionna and WI, Sionna and FLASH, and WI and FLASH. We also discuss the differences between the restricted and unrestricted approaches that we employed to combat missing beam data from FLASH, determining the strengths and weaknesses of each approach. We then compare the metrics of the Twins

Metric	Scenario	Sionna-WI			Sionna-FLASH				WI-FLASH			
					Restricted		Unrestricted		Restricted		Unrestricted	
		Base	1R	3R	1R	3R	1R	3R	1R	3R	1R	3R
MD	LOS	1.47	2.85	3.77	3.23	3.14	3.55	3.46	9.71	7.29	11.21	8.53
	NLOS	3.38	2.90	4.04	4.43	3.78	5.38	6.02	9.12	6.89	12.60	9.3
CS	LOS	0.99	0.99	0.99	0.99	0.99	0.74	0.74	0.99	0.99	0.99	0.99
	NLOS	0.99	0.96	0.99	0.76	0.99	0.48	0.50	0.99	0.99	0.99	0.99
CI	LOS	0.53	0.99	0.99	0.32	0.32	0.52	0.51	0.32	0.32	0.47	0.47
	NLOS	0.54	0.99	0.99	0.29	0.29	0.48	0.48	0.29	0.29	0.51	0.51
NDCG	LOS	0.74	0.85	0.88	0.94	0.93	0.57	0.54	0.92	0.95	0.90	0.91
	NLOS	0.83	0.88	0.86	0.95	0.94	0.53	0.55	0.96	0.97	0.87	0.90

TABLE II: Comparison of score metrics of Sionna and WI against an episode of FLASH using parameters such as LOS/NLOS scenarios, base, 1 and 3 reflection simulation fidelities and the use of restricted and unrestricted methods.

Execution Time (Seconds)				
# Beam Reflections	WI		Sionna	
	LOS	NLOS	LOS	NLOS
1	12.510	14.360	0.254	0.251
3	20.474	23.920	0.414	0.446
5	28.358	33.634	0.602	0.688
7	36.408	41.466	0.827	0.965
10	43.854	51.495	1.192	1.387

TABLE III: Comparison of WI and Sionna's execution time performance for calculating the received powers for one point in one episode of our experimentation. Measurements are provided for both LOS (Line of Sight) and NLOS (Non-Line of Sight) scenarios.

on a more granular level, by comparing their metrics in LOS vs NLOS scenarios, at speeds of 10 vs 15 vs 20 mph, and between 1 and 3 reflections for the fidelity of the Twin.

A. Digital Twin's Similarity to Real World

To assess the performance of WI and Sionna digital twins, we conduct an evaluation of the mapping accuracies, implementing the accuracy metric $\text{Acc}(K, T)$ with K set to 10 and gradually relaxed the SNR threshold T ($T = \{0 \text{ dB}, 1 \text{ dB}, 2 \text{ dB}\}$). Figure 4 illustrates the comparison of different DTs w.r.t. the ground-truth measurements from the FLASH dataset to validate the fidelity of emulation outputs (Base Twin, 1R, 3R). The power threshold $T = 0$ represents the most extreme case, considering only the top-K beams in FLASH. As shown in Figure 4, the absolute accuracies between WI and Sionna were similar overall, with Sionna's trends slightly outperforming WI within certain parameters. An exception to this is Acc(10, 1) in an LOS scenario, where Sionna's accuracy was lower in the Base twin experiment.

The trend for both simulators is that the mapping accuracy increases as the fidelity of the DT increases. For example, Sionna's accuracy in a LOS scenario for Acc(10, 0) goes from 0.54 to 0.73 to 0.82 for Base, 1 reflection, and 3 reflection twins, respectively. This is to be expected, as the number of reflections and similarity of the environment to the real world increases, the accuracy of the DT should increase.

B. Comparison between Digital Twins

In Table II, we compare WI to Sionna, WI to FLASH, and Sionna to FLASH using the metrics MD, CS, CI and NDCG. We also provide execution times in Table III. The MD scores between Sionna and WI trend upwards as the simulation fidelities increase from 1R to 3R. This could mean that the simulators' received powers diverge at lower fidelities and increase with fidelity. This could be due to deviations in the simulations becoming more pronounced with each computation. The CS of WI and Sionna are very high, meaning that the simulators' received powers are pointing in a very similar direction. The CI scores of the Base-Twins are low, but the NDCG scores are higher. The CI and NDCG scores were high for the 1R and 3R Twins. This leads us to believe that when accounting for every pair of received powers, the Base simulators do not have strong similarities between the order of the strongest and weakest beams. However, biasing towards the most powerful beams, the simulators do have stronger similarities; this is indeed the most interesting regime, as we care about the most powerful beams. Overall, the similarities between WI and Sionna were satisfactory.

When comparing the simulators to FLASH, other differences between them come to light. The MD is higher when comparing WI to FLASH versus comparing Sionna to FLASH. The CS is lower when comparing Sionna to FLASH in an unrestricted manner. The CI is similar between the two, but NDCG is lower for Sionna when using the unrestricted approach. This leads us to believe that Sionna performs similarly to FLASH and WI when accounting only for the present beams in FLASH, but WI is more robust to substituting missing data.

C. Enhanced Comparison between Sionna and FLASH

Since we were able to generate more episodes of Sionna to match the number of LOS and NLOS episodes in FLASH, we performed a third evaluation with an increased focus on the differences between Sionna and FLASH. Figures **??** and 8 delve into the differences between the two datasets. We use our restricted and unrestricted approaches for our comparisons. We split the datasets three times, one for LOS versus NLOS scenarios, one for 1 versus 3 reflection simulation fidelities, and one for 10 versus 15 versus 20 mph speeds.

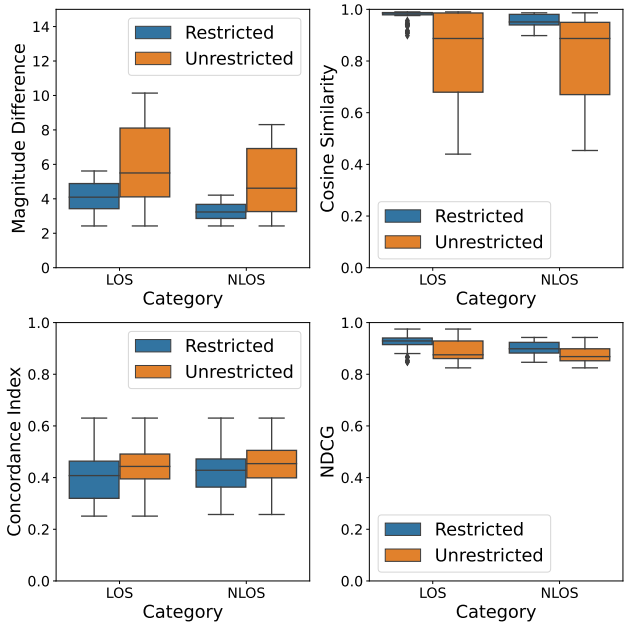


Fig. 5: Comparisons between the Sionna and FLASH datasets, grouped by LOS vs. NLOS, using the restricted (blue) and unrestricted (orange) methods.

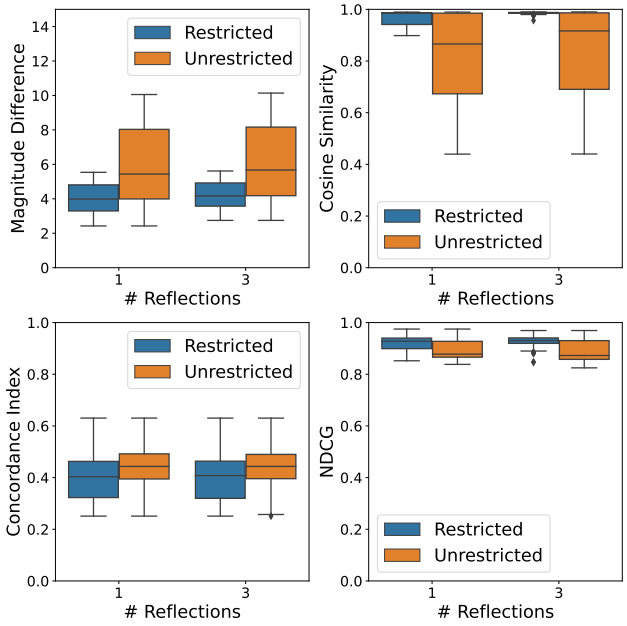


Fig. 6: Comparisons between the Sionna and FLASH datasets, grouped by 1 vs. 3 reflections, using the restricted (blue) and unrestricted (orange) methods.

When comparing the unrestricted and restricted approaches, the MDs are typically higher when using the unrestricted approaches. This is most likely due to the added zeros in the FLASH dataset not compensating for the non-zero elements in Sionna, causing the magnitude difference to be higher than when removing elements in the Sionna dataset. For the CS scores, the unrestricted approaches were also trending lower with higher standard deviations, once again most likely due to the added zeros not compensating adequately. The CS score being high in the restricted scenarios also implies that the direction of the Sionna datasets is still very accurate when accounting for beams with lost transmissions. The CI and NDCG scores had near identical medians, but significantly lower standard deviations in the unrestricted approaches. While the CI scores imply that there's little concordance between the datasets when taking all beams into account, the NDCG scores show that the added zeros increase the importance of the beams that are present, reducing the impact of missing beams.

When comparing across the LOS scenarios versus NLOS, the comparison scores of Figure 5 are similar, with the exception of MD, where the LOS scenario has a higher median score and standard deviation. This difference was unexpected; it may be due to how Sionna calculates the path loss of beams, leading to a lower path loss when there are fewer obstacles in the environment. When comparing across the 1 reflection scenarios versus 3 reflections (Figure 6), there is similar performance across all metrics, with a slight improvement using 3 reflections over 1. This supports that using higher fidelity DTs can improve the accuracy of real world simulations, but satisfactory performance can be achieved using lower fidelity

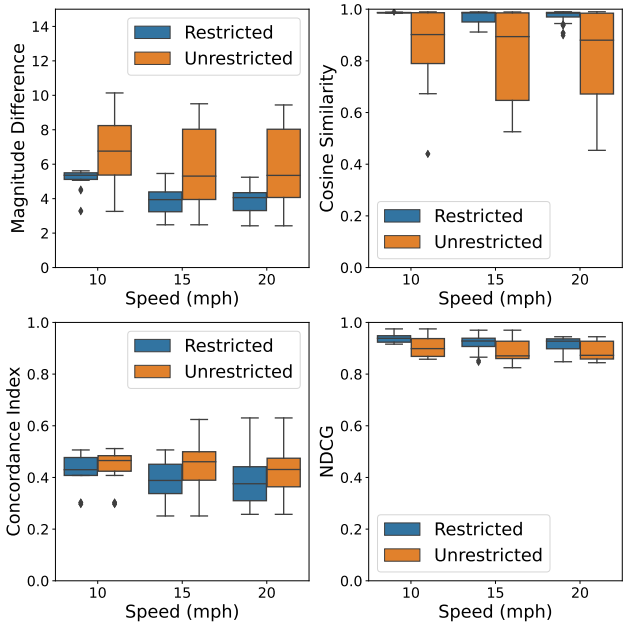


Fig. 7: Comparisons between the Sionna and FLASH datasets, grouped by 10 vs. 15 vs. 20 mph speeds, using the restricted (blue) and unrestricted (orange) methods.

DTs. When comparing 10 versus 15 versus 20 mph speed scenarios across the datasets (Figure 7), there is a significant increase in standard deviation of the MD and CS scores as the speed of the receiver increases. This is to be expected, as there

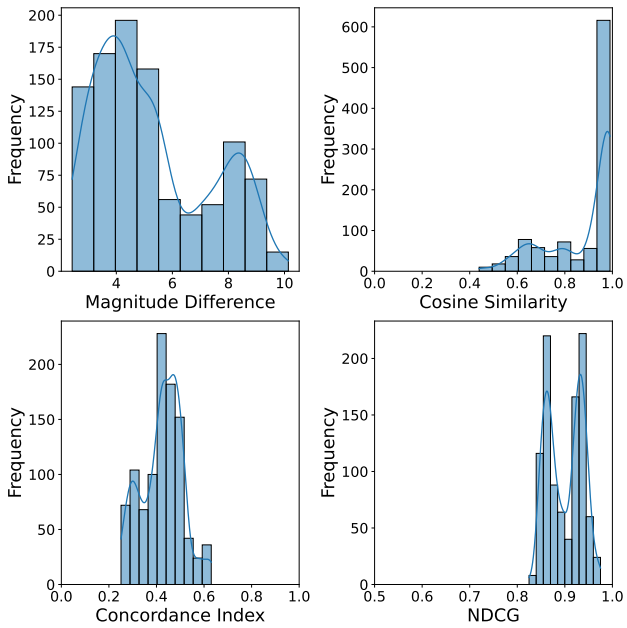


Fig. 8: Histogram plots detailing the distribution of the comparison metrics across the entirety of the Sionna and Flash datasets.

is a higher likelihood of inaccuracies between the estimated location of the receiver and the actual location. While the MD and CS scores had significant standard deviations, this was not the case for CI and NDCG, leading us to believe that these metrics are resilient to the variations between experiments.

Lastly, we aggregate the metric scores of the Sionna dataset (Figure 8). The CS, CI and NDCG distributions show that there was a relatively low variance in those scores, with their modes: 0.99, 0.5 and 0.85, respectively, occurring significantly more frequently than other values. MD, however, has a larger variance in the dataset and has a bimodal distribution. Comparing the plots in Figures 5–7 to Figure 8, this bimodal distribution is caused by the distributions of the unrestricted and restricted scores having differing peaks.

VI. Conclusions

Our real-world to digital-world comparison confirms that WI and Sionna achieve a sufficient level of accuracy with real-world scenarios, with respect to several key metrics. Beyond the direct comparison between Sionna and WI, this paper's contributions also include the creation of a dataset covering different wireless scenarios and their digital twins. We make this and our code publicly available as a valuable resource for researchers working on wireless communications.[1] Automating the DT generation in real-time, including via varied obstacle modeling, and using it for performance prediction as in Salehi *et al.* [5] is an potential future direction that we could consider. Another direction we could follow is integrating our pipeline with LiDAR simulating softwares to create a more comprehensive autonomous vehicle environment via a sensor fusion framework. We can integrate other sensors as well, both real-time and non-real-time. Real-time sensors such as image and event cameras and LiDAR sensors provide continuous data streams, immediately reflecting changes in the environment. Non-real-time sensors, such as weather stations, offer periodic updates suitable for applications like environmental monitoring where immediate response is not critical. Integrating them will make our DTs more comprehensive.

[1]https://github.com/genesys-neu/DITTO/tree/automation

References

[1] H. X. Nguyen, R. Trestian, D. To, and M. Tatipamula, "Digital twin for 5g and beyond," *IEEE Communications Magazine*, vol. 59, no. 2, pp. 10–15, 2021.

[2] Y. Wu, K. Zhang, and Y. Zhang, "Digital twin networks: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13 789–13 804, 2021.

[3] L. U. Khan, Z. Han, W. Saad, E. Hossain, M. Guizani, and C. S. Hong, "Digital twin of wireless systems: Overview, taxonomy, challenges, and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2230–2254, 2022.

[4] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, "Deep learning for hybrid 5g services in mobile edge computing systems: Learn from a digital twin," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4692–4707, 2019.

[5] B. Salehi, U. Demir, D. Roy, S. Pradhan, J. Dy, S. Ioannidis, and K. Chowdhury, "Multiverse at the edge: Interacting real world and digital twins for wireless beamforming," *IEEE/ACM Transactions on Networking*, 2023.

[6] Z. Tao, W. Xu, Y. Huang, X. Wang, and X. You, "Wireless network digital twin for 6g: Generative ai as a key enabler," *IEEE Wireless Communications*, 2023.

[7] U. Demir, S. Pradhan, R. Kumahia, D. Roy, and K. Chowdhury, "Digital twins for maintaining qos in programmable vehicular networks," *IEEE Network*, 2023.

[8] "Wireless InSite, 3D Wireless Prediction Software," https://www.remcom.com/wireless-insite-em-propagation-software.

[9] J. Hoydis, F. A. Aoudia, S. Cammerer, M. Nimier-David, N. Binder, G. Marcus, and A. Keller, "Sionna rt: Differentiable ray tracing for radio propagation modeling," in *2023 IEEE Globecom Workshops (GC Wkshps)*, 2023, pp. 317–321.

[10] M. G. Juarez, V. J. Botti, and A. S. Giret, "Digital Twins: Review and Challenges," *Journal of Computing and Information Science in Engineering*, 2021.

[11] "Blender," https://www.blender.org.

[12] B. Salehi, J. Gu, D. Roy, and K. Chowdhury, "FLASH: Federated Learning for Automated Selection of High-band mmWave Sectors," in *INFOCOM*. IEEE, 2022.

[13] Y. Li, K. Li, L. Cheng, Q. Shi, and Z.-Q. Luo, "Digital twin-aided learning to enable robust beamforming: Limited feedback meets deep generative models," in *SPAWC*, 2021.

[14] M. M. Abdulwahid, M. S. Al-Hakeem, M. F. Mosleh, and R. A. Abd-alhmeed, "Investigation and optimization method for wireless ap deployment based indoor network," *IOP Conference Series: Materials Science and Engineering*, 2020.

[15] X. Liao, Z. Wang, X. Zhao, K. Han, P. Tiwari, M. J. Barth, and G. Wu, "Cooperative ramp merging design and field implementation: A digital twin approach based on vehicle-to-cloud communication," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[16] Z. Wang, X. Liao, X. Zhao, K. Han, P. Tiwari, M. J. Barth, and G. Wu, "A digital twin paradigm: Vehicle-to-cloud based advanced driver assistance systems," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–6.

[17] "OpenStreetMap," https://www.openstreetmap.org.

[18] Prochitecture, "Real world environments for blender." [Online]. Available: https://prochitecture.gumroad.com/

[19] D. Steinmetzer, D. Wegemer, M. Schulz, J. Widmer, and M. Hollick, "Compressive millimeter-wave sector selection in off-the-shelf ieee 802.11 ad devices," in *CoNEXT*, 2017, pp. 414–425.

[20] "FLASH Dataset," https://genesys-lab.org/multimodal-fusion-nextg-v2x-communications.