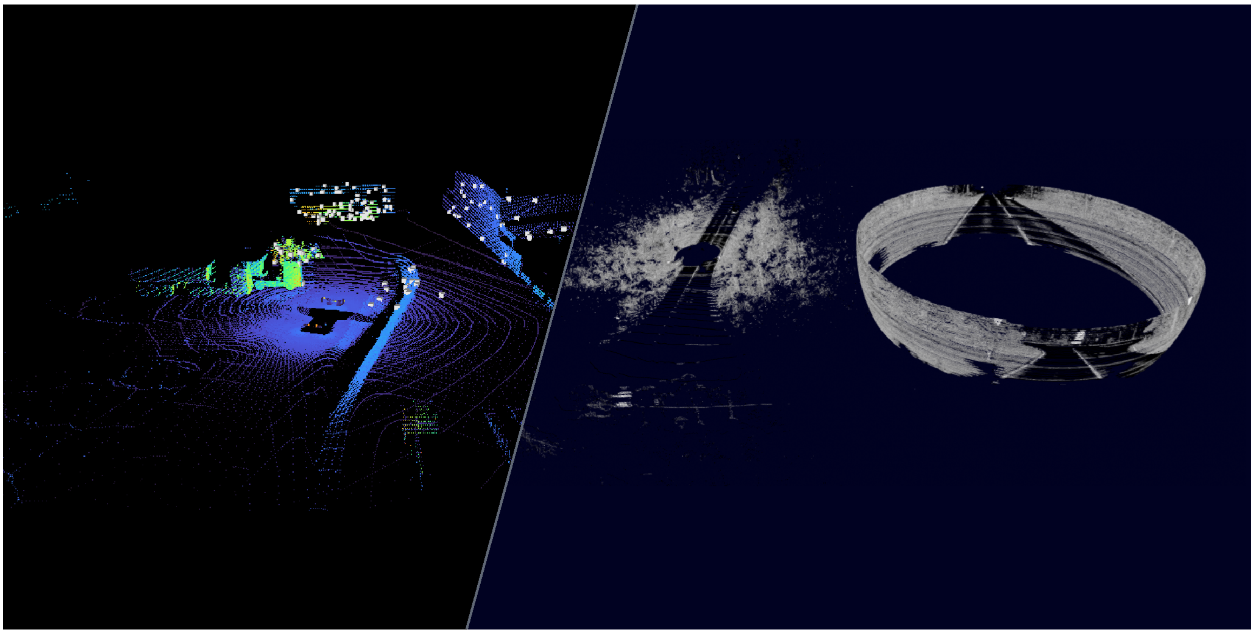


Real-Time LiDAR Sensor Modeling

Intensity Modeling and Evaluation for Autonomous Vehicle Simulation

Master's thesis in MPCAS and MPSYS

JINGBO ZHOU
HAILAN GARAU CHEN

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Master's thesis 2026

Real-Time LiDAR Sensor Modeling

Intensity Modeling and Evaluation for Autonomous Vehicle Simulation

JINGBO ZHOU
HAILAN GARAU CHEN

Department of Mechanics and Maritime Sciences
Chalmers University of Technology
Gothenburg, Sweden 2026

Real-Time LiDAR Sensor Modeling: Intensity Modeling and Evaluation for Autonomous Vehicle Simulation
JINGBO ZHOU
HAILAN GARAU CHEN

Supervisor: Yufei Zhang, Volvo Autonomous Solutions
Examiner: Peter Forsberg, Department of Mechanics and Maritime Sciences

Master's Thesis 2026
Department of Mechanics and Maritime Sciences
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: The left side shows a simulated LiDAR point cloud generated in the CARLA-based digital twin environment. The right side shows a visualization of the evaluation method.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Real-Time LiDAR Sensor Modeling: Intensity Modeling and Evaluation for Autonomous Vehicle Simulation
JINGBO ZHOU
HAILAN GARAU CHEN
Department of Mechanics and Maritime Sciences
Chalmers University of Technology

Abstract

Realistic LiDAR simulation is important for the development and validation of autonomous driving systems, but accurately reproducing LiDAR intensity remains challenging. Unlike point geometry, intensity depends on range, incidence angle, surface reflectivity, sensor-specific processing, and environmental effects. In addition, evaluating simulated intensity against real-world data is difficult because exact pointwise alignment between real and simulated point clouds is rarely achievable in a digital twin environment.

This thesis investigates LiDAR intensity simulation in a CARLA-based digital twin of the AstaZero proving ground, developed in connection with Volvo Autonomous Solutions. Real-world LiDAR reference data are reconstructed from MCAP recordings and used to evaluate the simulated intensity output. A physically motivated intensity model is introduced for the simulated LiDAR, incorporating the main factors that affect return strength, including range, incidence angle, and material reflectivity. However, because the target LiDAR sensor outputs a vendor-specific value affected by an inaccessible, proprietary internal processing pipeline, a direct analytical sensor model is unattainable. Hence, the framework complements this physical formulation to a final calibrated reflectivity simulation model through empirical distribution mapping. The resulting model serves as a practical, real-time approximation of calibrated reflectivity behavior rather than a complete reproduction of the internal sensor-processing pipeline.

To evaluate simulated LiDAR intensity, this thesis combines conventional histogram-based metrics with a novel geometry tolerant evaluation method proposed in this work. Wasserstein distance and Jensen–Shannon distance are used as baseline measures of global intensity distribution agreement. The proposed spherical harmonic based method represents each LiDAR frame as an angular intensity function on the sphere and compares frames using a weighted distance between their degree-wise spherical harmonic energy descriptors. This method captures coarse angular intensity structure in a rotation invariant manner without requiring exact pointwise correspondence.

The results show that the proposed intensity model improves the similarity between simulated and real-world reference intensity distributions. The proposed evaluation method also provides a more informative comparison than traditional distribution-based metrics by preserving directional intensity structure when local geometric mismatch is present.

Keywords: LiDAR simulation, intensity modeling, calibrated reflectivity, autonomous driving, digital twin, spherical harmonics, sim-to-real evaluation

Acknowledgements

We would like to express our sincere gratitude to our examiner, Peter Forsberg at the Department of Mechanics and Maritime Sciences, Chalmers University of Technology, for his academic guidance, constructive comments, and continuous support throughout the thesis process. We especially appreciate that he organized regular weekly meetings with us, which helped us a lot in keeping the work on track, discussing problems in time, and improving the thesis step by step.

We would also like to thank our supervisor, Yufei Zhang at Volvo Autonomous Solutions, for his guidance, support, and valuable feedback throughout this thesis work. His technical insights and practical suggestions have been very helpful in shaping the direction of this project and improving the quality of the work.

Furthermore, we would like to thank Volvo Autonomous Solutions for providing the opportunity to work on this thesis topic and for giving us access to the simulation environment, data, and technical resources needed for the project. We are also grateful to the colleagues and engineers who supported us with discussions, technical advice, and practical assistance during the development and evaluation of the LiDAR simulation model.

Personal Acknowledgements

[Jingbo Zhou]:

I would like to express my deepest gratitude to my parents, Shaolan Li and Yong Zhou, for their unconditional love, trust, and support. Their encouragement has given me the opportunity and confidence to continue my master's studies abroad, and their support has always been the foundation that allowed me to keep moving forward.

I would also like to thank all my friends back in China, especially Han Qiu and Kaiyuan Xue, for their encouragement and support throughout this journey. I would like to especially thank Cunkun Lin, an amazing indie game developer and a close friend, for the support and inspiration he has given me over the years. I am also grateful to all my friends at Chalmers, especially Eric Blohm and my friends from the Complex Adaptive Systems programme, as well as the friends I have met in Sweden during the past two years. Living and studying in a different country has not always been easy, and without their help, company, and kindness, it would have been much harder for me to adapt to life here.

Finally, I would like to thank my thesis partner, Hailan Garau Chen, for his great effort, patience, and support throughout this thesis work. He has contributed a lot to this project, and I have truly enjoyed working together with him during this process.

[Hailan Garau Chen]:

To my parents, Yueh Jiuan Chen Yang and José Alberto Garau Serra, and my sister, Sinty Garau Chen, from the bottom of my heart, thank you. Thank you for your constant support, for sharing with me your wisdom and thoughtful advice, and for loving me for who I am. Thank you for being here with me despite the kilometers keeping us apart, and for making me the person I am today. I love you all deeply.

I would also like to thank Murilo Cossovan Marques, Juan Manuel Rubio Vanegas, Toshith Manglani and Kanishk Nama, for being a constant throughout this master's, for being there in the good, and especially in the bad times, and for being my home in Gothenburg.

In addition, thanks to my neighbor Alejandro Vivas-Viaña for the late-night talks and his friendship. I am incredibly grateful to have shared these past two years with the rest of my friends from the MPSYS program, my volleyball teammates at Gothia, and the wonderful people at Spanska Köket.

To the rest of my family and friends back in Spain, thank you for your support from afar. Even though our contact has been less frequent during these years, knowing that our bond remains unchanged and that you will be there when I return has been a constant source of strength.

Last but not least, I want to express my deepest thanks to my thesis partner, Jingbo Zhou. Thank you for your strong and constant support throughout this thesis, your clever insights, and for being such an amazing person to collaborate with.

Jingbo Zhou and Hailan Garau Chen, Gothenburg, 2026

Thesis advisor: Yufei Zhang, Volvo Autonomous Solutions
Thesis examiner: Peter Forsberg, Mechanics and Maritime Sciences

Acknowledgement of AI Use

During the preparation of this report, generative AI tools were utilized for supporting tasks. Google Gemini and OpenAI ChatGPT assisted with language refinement, manuscript editing, and LaTeX table formatting, while GitHub Copilot was used within VS Code to support data processing script development. The technical framework, data analysis, and final conclusions represent the original work of the authors. All AI-assisted suggestions and code outputs were thoroughly reviewed, verified, and manually adapted. The authors bear full responsibility for all content presented.

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

ADS	Autonomous Driving System
BRDF	Bidirectional Reflectance Distribution Function
BVH	Bounding Volume Hierarchy
FMU	Functional Mock-up Unit
FOV	Field of View
GNSS	Global Navigation Satellite System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
JSD	Jensen–Shannon Distance
LiDAR	Light Detection and Ranging
MSE	Mean Squared Error
NIR	Near-Infrared
RFL	Reflectivity
RGB	Red, Green, and Blue
RMSE	Root Mean Squared Error
SH	Spherical Harmonics
UDP	User Datagram Protocol
VAS	Volvo Autonomous Solutions
WD	Wasserstein Distance
XYZI	X, Y, Z, and Intensity

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

k	Index for a LiDAR point or angular intensity sample
i	Measurement index of interpolation region
j	Index for a basis function in the flattened spherical harmonic basis matrix
ℓ	Degree of a spherical harmonic basis function
m	Order of a spherical harmonic basis function

Sets

$\mathcal{M}$	Available materials within the spectral library database
H	Candidate parametric models for sensor-specific empirical distribution mapping
$\mathcal{P}$	LiDAR point cloud frame
$\mathcal{P}^{\mathrm{real}}$	Real-world LiDAR point cloud frame
$\mathcal{P}^{\mathrm{sim}}$	Simulated LiDAR point cloud frame
$\mathcal{S}$	Set of angular intensity samples on the sphere
S^2	Unit sphere
$L^2(S^2)$	Space of square-integrable functions on the unit sphere

Parameters

N	Number of valid LiDAR points or angular samples in a frame

K	Number of spherical harmonic basis functions
L	Maximum spherical harmonic degree used in the approximation
α	Range-dependent attenuation coefficient
a	Gain parameter in the empirical intensity mapping
b	Offset parameter in the empirical intensity mapping
c	Scale parameter in the logarithmic empirical mapping
γ	Exponent parameter in the power-law empirical mapping
w_k	Weight assigned to sample k in coefficient estimation
w_ℓ	Weight assigned to spherical harmonic degree ℓ in the energy distance

Variables

p_k	The k-th LiDAR point in XYZI format
x_k, y_k, z_k	Cartesian coordinates of the k-th LiDAR point
i_k	Intensity value of the k-th LiDAR point
r_k	Range of the k-th LiDAR point from the sensor origin
θ_k	Polar angle of the k-th LiDAR point
ϕ_k	Azimuth angle of the k-th LiDAR point
$I(\theta, \phi)$	Angular intensity function on the sphere
I_{sliver}	Normalized intensity value extracted from a PointCloudSliver message
R_{RFL8}	Reconstructed Ouster RFL8 calibrated reflectivity representation
I_{phys}	Physically motivated simulated return intensity
ρ_m	Reflectivity of semantic material m
$\rho_m(\theta)$	Incidence-angle-dependent reflectivity of semantic material m
$A(r)$	Range-dependent attenuation term
F_{Ouster}	Fitting curve from physical return intensity to Ouster-like calibrated reflectivity
(x_q, y_q)	Quantile pairs for sensor-specific empirical distribution mapping
β	Interpolation weights for material reflectivity computation
μ_h	Parameter vector for model h
$\hat{\mu}_h$	Optimal parameter vector for model h
$h^\star$	Optimal mapping model
BL	Baseline model

PM-BF	Proposed model before fitting
$F_{A,\text{full}}$	Full Field-of-View (FOV) fitting curve for scenario A
$F_{B,\text{full}}$	Full Field-of-View (FOV) fitting curve for scenario B
$F_{A,\text{red}}$	Reduced-FOV fitting curve for scenario A
$F_{B,\text{red}}$	Reduced-FOV fitting curve for scenario B
$\bar{F}^{\text{full}}_{\text{Ouster}}$	Master Full-FOV fitting curve
$\bar{F}^{\text{red}}_{\text{Ouster}}$	Master Reduced-FOV fitting curve
R_{sim}	Final simulated calibrated reflectivity after sensor-specific mapping
$Y_\ell^m(\theta, \phi)$	Complex spherical harmonic basis function
$Y_{\ell m}^{\text{real}}(\theta, \phi)$	Real-valued spherical harmonic basis function
$c_{\ell m}$	Spherical harmonic coefficient of degree ℓ and order m
$\hat{\mathbf{c}}$	Estimated spherical harmonic coefficient vector
$\mathbf{A}$	Spherical harmonic basis matrix evaluated at LiDAR sample directions
$\mathbf{W}$	Diagonal sample-weight matrix
$\mathbf{i}$	Vector of LiDAR intensity samples
E_ℓ	Degree-wise spherical harmonic energy at degree ℓ
$\mathbf{E}$	Spherical harmonic energy descriptor
D_{w}	Weighted spherical harmonic energy distance

Contents

List of Acronyms **xi**

Nomenclature **xiii**

List of Figures **xix**

List of Tables **xxiii**

1 Introduction **1**
- 1.1 Background 2
- 1.2 Purpose 3
  - 1.2.1 Research Questions 3
- 1.3 Limitations 4

2 Theory **5**
- 2.1 Simulation for ADS 5
- 2.2 Unreal Engine 5
- 2.3 CARLA 6
- 2.4 LiDAR 6
  - 2.4.1 LiDAR Sensing 7
  - 2.4.2 LiDAR Intensity 7
- 2.5 Ray Casting 7
- 2.6 Spherical Projection 8
- 2.7 Spherical Harmonics 9
- 2.8 Sim-to-Real Gap 10

3 Methods **13**
- 3.1 Reference LiDAR Data 13
  - 3.1.1 Real-World Data Collection and MCAP Recording 13
  - 3.1.2 Vendor-Specific LiDAR Data Structure 15
  - 3.1.3 Point Cloud Frame Reconstruction 18
- 3.2 Simulated LiDAR Data 19
  - 3.2.1 Simulation Environment and Virtual Sensor Setup 19
  - 3.2.2 Ray-Casting-Based LiDAR Generation 21
  - 3.2.3 Physically Motivated Intensity Model 22

3.2.3.1 Range-Dependent Attenuation 22
3.2.3.2 Semantic Material and Angle-dependent Material Reflectance 23
3.2.4 Sensor-Specific Empirical Distribution Mapping 24
3.2.5 Cross-Scenario Generalization and Validation Framework . . . 25
3.3 Evaluation Method 28
3.3.1 Spherical Intensity Representation 29
3.3.1.1 Function Space and Inner Product 30
3.3.1.2 Spherical Harmonic Basis Functions 31
3.3.1.3 Discrete Coefficient Estimation from LiDAR Samples 32
3.3.2 Rotationally Invariant Energy Descriptor 34

4 Results 39
4.1 Intensity Distributions Alignment and Fitting Effects 39
4.2 Traditional Distribution-Based Frame Comparison 46
4.2.1 Evaluation in Generalization Scenarios (Scenarios A and B) . 49
4.2.2 Evaluation in Validation Scenarios (Scenarios C and D) 50
4.2.3 Jensen–Shannon Distance Analysis 50
4.3 Spherical Harmonic Energy Distance Evaluation 52
4.3.1 Validation of the Spherical Harmonic Evaluation Metric: Frame-Shift Sensitivity Analysis 53
4.3.2 Continuous Distance Evolution in Reference Configurations (Scenarios A and B) 54
4.3.3 Generalization Capabilities Across Unseen Trajectories (Scenarios C and D) 57
4.3.4 Quantitative Statistical Summary 57

5 Discussion 61
5.1 Interpretation of the Intensity Model Results 61
5.1.1 Overall Improvement of the Proposed Model 61
5.1.2 Choice of Fitting Data 62
5.1.3 Scene and Material Mismatch 63
5.1.4 Reflectivity Variation Under Different Conditions 63
5.1.5 Sensor-Specific Processing Gap 64
5.2 Comparison Between Traditional Metrics and the Proposed Evaluation Method 65
5.2.1 Role of Traditional Distribution-Based Metrics 65
5.2.2 Need for Directional Intensity Comparison 66
5.2.3 Limitations of the Spherical Harmonic Energy Method 66
5.2.4 Complementary Interpretation of the Metrics 67
5.3 Future Work 67

6 Conclusion 69

Bibliography 71

List of Figures

2.1 Illustration of LiDAR simulation by ray casting. Virtual rays are emitted into the scene to approximate the sensing process of a LiDAR, and intersections with scene geometry are used to generate simulated returns. 8

2.2 Illustration of spherical projection for LiDAR data 9

2.3 Visualization of real spherical harmonic basis functions of increasing degree and order. The figure is adapted from Wikipedia, *Spherical harmonics*: `https://en.wikipedia.org/wiki/Spherical_harmonics`. 10

3.1 Compact overview of the proposed sim-to-real evaluation pipeline. A real LiDAR frame is reconstructed from recorded MCAP data, while a corresponding simulated frame is generated in the digital twin using the proposed intensity model. The two frames are then loaded into the evaluation program and compared using the selected metrics. . . 14

3.2 Simplified organization of the MCAP recording format used for LiDAR data extraction. 15

3.3 Example Ouster channel data structure for the `RNG19_RFL8_SIG16_NIR16` single-return profile. The profile contains range, calibrated reflectivity, signal, and near-infrared fields for each channel data block. 17

3.4 Mapping between the 8-bit reflectivity representation and calibrated reflectivity values for the Ouster `RFL8` field, shown on a logarithmic scale. [15] 18

3.5 Example CARLA simulation scene and corresponding virtual LiDAR output. The overlaid point cloud view shows the LiDAR returns generated from the simulated environment. 20

3.6 Illustration of spherical projection for LiDAR data. The left side shows the original LiDAR point cloud in Cartesian space, while the right side shows the same data mapped onto the spherical domain according to the angular coordinates of each point. This representation preserves the directional structure of the LiDAR frame and provides the basis for subsequent spherical harmonic analysis. 30

3.7 Rotational invariance of the spherical harmonic energy descriptor. The two LiDAR frames are rotated relative to each other, but their spherical harmonic energy descriptors remain the same or very similar because the descriptor depends on degree-wise energy rather than the absolute orientation of the frame. 37

4.1 Single-frame intensity and calibrated reflectivity distribution matrices for Scenario A (left column) and Scenario B (right column). The top row (a, b) illustrates the raw, uncalibrated gap between simulated physical intensity (I_{phys}) and real-world reference data (R_{RFL8}). The middle row (c, d) shows the resulting distributions after applying empirically fitted master Full-FOV function ($\bar{F}^{\text{full}}_{\text{Ouster}}$). The bottom row (e, f) shows the alignment achieved using the master Reduced-FOV function ($\bar{F}^{\text{red}}_{\text{Ouster}}$). 41

4.2 Empirical quantile matching pairs with corresponding optimized cubic regression curves for Scenario A (Dry Conditions). 42

4.3 Empirical quantile matching pairs with corresponding optimized cubic regression curves for Scenario B (Wet Conditions). 43

4.4 Comprehensive comparison of all local scenario-specific calibration curves and their corresponding averaged master functions ($\bar{F}^{\text{full}}_{\text{Ouster}}$ and $\bar{F}^{\text{red}}_{\text{Ouster}}$). 44

4.5 Comparison of real-world and simulated LiDAR scan intensity frames under dry conditions (Scenario A). The real-world reference data captures the material heterogeneity and structural variations between gravel and soil on the track. On the other hand, the digital-twin simulation renders the drivable area as a completely homogeneous material surface. 45

4.6 Comparison of real-world and simulated LiDAR scan intensity frames under wet conditions (Scenario B). The real-world reference data shows highly localized low-reflectivity regions caused by water puddles that scatter the laser beams away from the receiver. In contrast, the simulation treats the entire surface with a uniform, unvaried wet material profile, failing to replicate the irregular spatial distribution of surface water. 46

4.7 Continuous frame-level temporal evolution of the Wasserstein distance (top) and Jensen–Shannon distance (bottom) across the Scenario A (Dry) trajectory, focusing on the scenario specific and master configurations of the proposed calibration model. 49

4.8 Continuous frame-level temporal evolution of the Wasserstein distance (top) and Jensen–Shannon distance (bottom) across the Scenario B (Wet) trajectory, focusing on the scenario specific and master configurations of the proposed calibration model. 50

4.9 Continuous frame-level temporal evolution of the Wasserstein distance (top) and Jensen–Shannon distance (bottom) across the validation Scenario C (Dry) using master fitting curves. 51

4.10 Continuous frame-level temporal evolution of the Wasserstein distance (top) and Jensen–Shannon distance (bottom) across the validation Scenario D (Wet) using master fitting curves. 51
4.11 Workflow of the spherical harmonic energy distance evaluation. Raw point clouds are mapped into spherical intensity projections, followed by degree-wise spherical harmonic coefficient expansions to derive a rotation-invariant distance metric. 52
4.12 Autocorrelation validation of the spherical harmonic energy distance metric using short temporally shifted sequences ($\Delta t \in \{0, 5, 10, 15\}$ frames) of the real-world Scenario A log. 53
4.13 Sensitivity profile of the spherical harmonic energy distance as a function of discrete frame shift displacement from 0 to 20 frames. Central point markers denote the energy distance mean, while the vertical error bars represent the ±1 standard deviation (σ). 54
4.14 Continuous spherical harmonic energy distance evolution along the full recording of Scenario A (Dry Environment). Panel (a) contrasts the uncalibrated models against the fitted curves, while Panel (b) presents the differences between scene-specific and master curves. . . 55
4.15 Continuous spherical harmonic energy distance evolution along the full recording of Scenario B (Wet Environment). Panel (a) contrasts the uncalibrated models against the fitted curves, while Panel (b) presents the variances between scene-specific and master curves. . . . 56
4.16 Continuous spherical harmonic energy distance tracking across unseen Scenarios C and D, demonstrating the generalization capability of the two master calibration curves. 58

List of Tables

3.1 Matrix showing the Vendor-Specific Empirical Mapping Framework . 28

4.1 Optimal cubic regression parameters and fitting errors for the calibration configurations 44
4.2 Statistics of frame-level Wasserstein distance (WD) across all scenarios and calibration configurations. Lower values indicate closer frequency distribution agreement with the real-world sensor logs. . . . 47
4.3 Statistics of frame-level Jensen–Shannon distance (JSD) across all scenarios and calibration configurations. Lower values indicate closer frequency distribution agreement with the real-world sensor logs. . . . 48
4.4 Comprehensive descriptive statistics of the continuous frame-level spherical harmonic energy distance logs across all scenarios and model configurations. Lower values indicate closer agreement with the real-world sensor logs. 59

1

Introduction

Autonomous driving systems (ADS) are becoming increasingly important in both research and industrial applications. They are expected to improve transportation safety, increase operational efficiency, and enable new forms of mobility and logistics in domains such as mining, construction, and transport. To achieve these goals, ADS must operate robustly in complex and changing environments, which places high demands on both the algorithms and the sensor systems on which these algorithms rely.

Simulation plays a critical role in the development and validation of ADS. Compared with purely real-world testing, simulation provides a safe, scalable, and cost-efficient environment for testing perception, localization, planning, and control algorithms, while also enabling systematic scenario generation, repeatable experiments, and large-scale data generation for training and evaluation. At VAS, an in-house simulation platform based on Unreal Engine has been developed as an evaluation environment for autonomous driving algorithms. The platform can generate synthetic sensor data for the autonomous driving stack, including camera, LiDAR, radar, and GNSS/INS-related measurements. Among these modalities, realistic LiDAR simulation remains particularly challenging, especially with respect to intensity modeling.

LiDAR intensity is governed by a complex sensing process involving laser emission, atmospheric propagation, surface interaction, and signal reception. As a result, the returned intensity is affected by multiple factors, including range, surface reflectivity, incidence angle, material properties, sensor-specific characteristics, and environmental conditions. Capturing these effects in a physically meaningful yet computationally efficient way is therefore a major challenge in real-time simulation.

Evaluating the quality of simulated LiDAR intensity is also a challenging task in an industrial setting. Reliable assessment often requires considerable time, computational effort, and access to representative real-world data. Furthermore, LiDAR intensity is influenced by a range of factors beyond surface properties alone, which makes it difficult to model and evaluate consistently. Consequently, both realistic intensity generation and robust quality assessment remain important challenges in real-time LiDAR simulation.

Against this background, this thesis aims to improve the current LiDAR simulation pipeline in two directions: first, by introducing a more physically motivated

LiDAR intensity model suitable for real-time simulation; and second, by proposing a geometry-tolerant evaluation method that can assess simulated LiDAR intensity quality without requiring exact one-to-one alignment with real-world data.

1.1 Background

Autonomous driving development relies heavily on simulation. Simulation is used for system testing, scenario-based validation, and synthetic data generation, allowing developers to evaluate algorithms under rare, dangerous, or difficult-to-reproduce conditions. In these applications, the fidelity of the simulation model is important. If the simulated sensor data do not represent the behavior of the real sensor sufficiently well, the resulting data distributions and evaluation results may not transfer reliably to real-world operation.

LiDAR simulation is an important part of autonomous driving simulation because LiDAR sensors provide dense three-dimensional information for perception, localization, and scene understanding. In simulation, a LiDAR point cloud can be considered from two related perspectives: geometry and intensity. The geometric part describes where the simulated rays intersect the environment, while the intensity part describes the strength or reflectivity-related value assigned to each valid return. Compared with geometric ray casting, realistic intensity simulation is more difficult because the measured intensity is affected by many physical and sensor-specific factors.

Previous work on LiDAR radiometric processing has shown that measured intensity depends on factors such as range, incidence angle, target reflectance, atmospheric effects, and sensor-dependent processing [1, 2]. In automotive LiDAR simulation, material properties and incidence-angle-dependent reflectance have also been identified as important components for generating more realistic simulated returns [3]. Other high-fidelity LiDAR sensor models include detailed ray tracing, scan pattern modeling, and sensor signal-processing components [4]. However, a real-time simulation system cannot usually reproduce the full physical and internal sensor-processing pipeline in detail. A practical model must therefore balance physical motivation, computational efficiency, and the ability to approximate the intensity representation observed in real sensor data.

In this thesis, the simulated LiDAR intensity model is formulated as a simplified physically motivated model suitable for real-time simulation. The model focuses on the main factors that affect the returned intensity, including range-dependent attenuation, incidence angle, and material reflectivity. Since the reference intensity values are calibrated and vendor-specific, the simulated output is later adjusted to make it comparable with the real-world reference data.

Evaluating the quality of simulated LiDAR intensity is also challenging. Previous approaches have evaluated LiDAR realism through physical accuracy models, radiometric correction, or downstream perception performance [5, 6, 7]. These methods are useful for understanding different aspects of LiDAR data quality, but they either require detailed physical calibration and sensor information or assess realism indirectly through the performance of a perception algorithm. In this thesis, the goal is to compare simulated and real-world intensity behavior more directly, without relying on a complete physical sensor model or an end-to-end perception evaluation.

A further difficulty is that exact point-wise correspondence between real and simulated LiDAR data is difficult to achieve. The digital twin is only an approximation of the real test site, the ray casting geometry is simplified for computational efficiency, and the real-world sensor pose may not be reproduced exactly in simulation. Traditional distribution-based metrics, such as histogram comparison, Wasserstein distance, or Jensen–Shannon distance, avoid strict point-wise correspondence by comparing intensity distributions. However, they ignore where the intensity values occur around the sensor. Two observations may therefore have similar intensity histograms while still having different angular intensity structures. This motivates an evaluation method that remains tolerant to geometric mismatch while preserving directional intensity information.

Motivated by these limitations, this thesis proposes a geometry-tolerant frame-level evaluation method based on spherical harmonic energy descriptors. Each LiDAR frame is converted into an angular intensity representation on the sphere. The frame-level intensity function is then approximated using spherical harmonic basis functions, and the degree-wise coefficient energy is used as a rotationally invariant descriptor. This descriptor preserves information about the angular distribution of intensity while avoiding strict point-wise correspondence. The resulting energy distance provides a practical way to compare real and simulated LiDAR intensity when small pose differences and local geometric mismatch are unavoidable.

1.2 Purpose

The purpose of this thesis is to develop a high-fidelity LiDAR intensity model suitable for real-time simulation, and to propose a practical method for evaluating the quality of simulated LiDAR intensity data in an industrial simulation environment.

More specifically, the work aims to improve the physical realism of LiDAR intensity generation while also introducing an evaluation framework that is robust to imperfect geometric alignment between real and simulated observations. Together, these two components are intended to support more reliable development and validation of LiDAR simulation at VAS.

1.2.1 Research Questions

This thesis is guided by the following research questions:

1. How can a high-quality LiDAR intensity model be designed for real-time simulation?
2. How can the quality of simulated LiDAR intensity be evaluated in a robust

and practical manner?

1.3 Limitations

Since the simulator is required to operate in real time, the proposed intensity model cannot include all physical effects in full detail. Some factors must therefore be simplified or approximated in order to satisfy computational constraints. The work in this thesis focuses on identifying a balance between physical realism and runtime efficiency, rather than constructing a fully exhaustive physical sensor model.

In addition, the proposed evaluation method does not aim to recover true physical reflectance in a strict radiometric sense. Instead, it relies on intensity calibration to transform raw LiDAR intensity into a more stable proxy related to surface properties. For LiDAR systems in which such calibration is already handled, or can be approximated reliably from the available measurements, the evaluation framework is expected to be relatively robust. However, for sensors that provide only raw intensity values in a less normalized form, additional preprocessing may be required before the method can be applied consistently.

A further limitation is that the evaluation is designed to compare the overall distributional behavior of the intensity signal rather than exact pointwise correspondence. This makes the method more robust to geometric mismatch, but it also means that it is not intended to diagnose fine-grained local errors at the individual point level. Detailed pointwise discrepancies and downstream task effects are therefore outside the primary scope of the proposed evaluation framework.

2

Theory

2.1 Simulation for ADS

In engineering, simulation generally refers to the use of computational models to reproduce selected aspects of a real system, such that its behavior can be studied under controlled conditions. Instead of interacting with the physical system directly, simulation provides a virtual environment in which inputs, states, and external conditions can be systematically adjusted and observed [9].

In the context of autonomous driving systems (ADS), simulation denotes the virtual reproduction of the driving task, including the road environment, surrounding traffic participants, vehicle dynamics, and sensor observations. Its purpose is not merely to visualize traffic scenes, but to emulate the interaction between the autonomous vehicle and its environment, so that modules such as perception, localization, planning, and control can be developed and analyzed in a digital setting [8].

ADS simulation can therefore be understood as a combination of environment modeling, actor modeling, vehicle modeling, and sensor simulation. Since autonomous driving operates in open, dynamic, and safety-critical environments, simulation has become an important component of the broader development and validation pipeline, where it complements real-world testing by enabling controlled and repeatable experimentation [9, 10].

2.2 Unreal Engine

Unreal Engine is a real-time 3D engine for constructing and executing interactive virtual environments. More generally, it provides a computational framework in which scenes are represented through geometry, materials, lighting, dynamic objects, and spatial relationships, while also supporting rendering, physics, and interaction. In this sense, Unreal Engine is not merely a graphics tool, but a digital world engine that can represent and update complex virtual environments in real time.

Because of this, Unreal Engine can also serve as a foundation for simulation. It enables roads, buildings, vehicles, pedestrians, and other scene elements to be modeled within a coherent virtual world, which can then be used not only for visualization but also for synthetic sensor generation through rendering or geometric queries.

In autonomous driving simulation, this makes Unreal Engine a suitable basis for constructing driving environments and supporting perception-oriented simulation platforms such as CARLA.

2.3 CARLA

CARLA [11] is an open-source simulator developed for autonomous driving research, designed to support the development, training, and validation of autonomous driving systems [8]. Built on top of Unreal Engine, it provides a virtual driving environment together with the infrastructure needed to simulate vehicles, traffic participants, and sensor modules within a unified framework [8]. In this sense, CARLA can be understood as a higher-level autonomous driving simulator that extends Unreal Engine toward autonomous driving applications.

For this thesis, CARLA is relevant as the platform in which the LiDAR simulation is implemented. Its simulator architecture provides a convenient basis for sensor development, since virtual sensors can interact with the environment and access the geometric information required for measurement generation. Based on this framework, the LiDAR simulation developed in this thesis can generate simulated point cloud data, which can then be exported and compared with corresponding real-world test data in the later evaluation stage [12].

2.4 LiDAR

LiDAR, short for Light Detection and Ranging, is an active sensor that measures the environment by emitting laser light and analyzing the returned signal after reflection from surrounding objects. In automotive systems, distance is commonly estimated using the time-of-flight principle, and together with the beam direction this allows the sensor to generate a three-dimensional point cloud of the scene [18].

LiDAR is crucial in autonomous driving systems (ADS) because the vehicle must reliably perceive three-dimensional structure in order to detect obstacles, estimate free space, localize itself, and understand the spatial arrangement of surrounding objects. Compared with cameras, which do not directly measure depth, and radar, which usually provides lower spatial resolution, LiDAR offers detailed geometric information and therefore plays an important role in perception, mapping, localization, and multi-sensor fusion.

In simulation, LiDAR is equally important because it is one of the main ways a virtual environment is converted into machine-readable sensor data. In platforms such as CARLA, LiDAR is commonly implemented through ray-casting, where virtual rays intersect scene geometry to generate simulated point clouds. This makes LiDAR simulation an efficient basis for further modeling of intensity and other sensor effects.

2.4.1 LiDAR Sensing

The basic sensing principle of LiDAR is to emit laser pulses into the environment and measure the returned signal after reflection from surrounding surfaces. In most automotive LiDAR systems, the travel time of the emitted light is used to estimate the distance to a target, and together with the beam direction this allows the sensor to determine the three-dimensional position of reflecting surfaces. By repeating this process across many directions, the sensor generates a point cloud that describes the geometric structure of the environment.

In practice, LiDAR sensing is influenced by several factors beyond range measurement alone. The returned signal depends on propagation distance, incidence angle, surface properties, and sensor characteristics. In particular, increasing distance generally weakens the return due to attenuation, while large incidence angles often reduce the received energy. These effects are important in simulation, since a realistic LiDAR model should not only reproduce point positions, but also capture how sensing behavior varies with geometry and surface interaction. In the simulation developed in this thesis, such factors are therefore considered as important components for improving the realism of the LiDAR sensing process.

2.4.2 LiDAR Intensity

In addition to geometric range information, many LiDAR systems provide an intensity value for each return. This value describes the strength of the received signal and can therefore provide information beyond geometry alone. In practice, LiDAR intensity can help distinguish surfaces with similar shape but different optical properties, which makes it useful for tasks such as perception, localization, and scene interpretation. However, intensity is not a direct material property, since the recorded value depends not only on the surface itself but also on sensing conditions [1].

LiDAR intensity is influenced by distance, incidence angle, material properties, and sensor-specific characteristics. The returned signal generally weakens with increasing distance due to propagation losses, while larger incidence angles often reduce the energy reflected back toward the receiver. Material properties such as reflectance, roughness, and wavelength-dependent behavior also affect the response. In addition, recorded intensity is often sensor- and vendor-dependent, which makes realistic simulation more challenging.

In this thesis, these effects are treated as the main targets of the LiDAR intensity model in order to improve the realism of the simulated sensor response.

2.5 Ray Casting

Ray casting is a fundamental technique for simulating LiDAR measurements in virtual environments. The basic idea is to mimic the sensing process of a real LiDAR by emitting virtual rays into the scene and computing their intersections

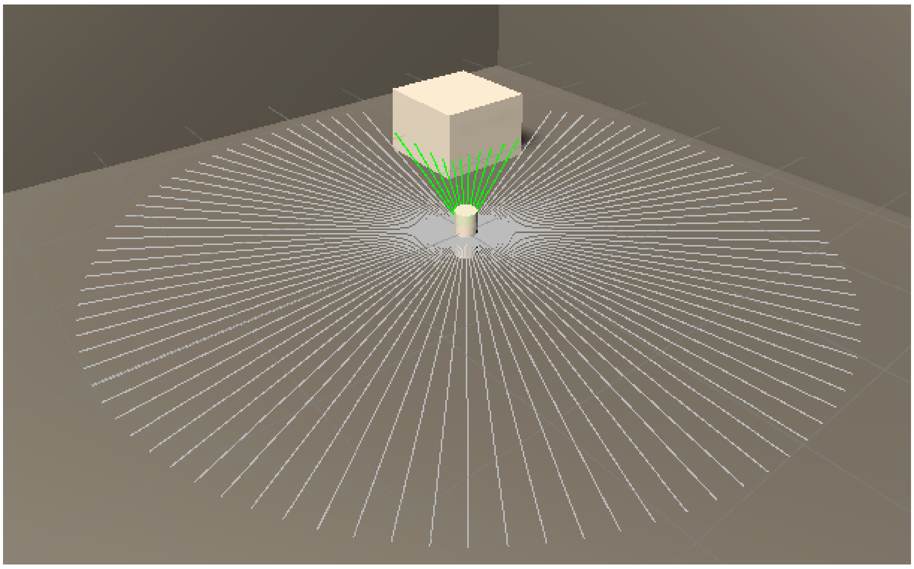


Figure 2.1: Illustration of LiDAR simulation by ray casting. Virtual rays are emitted into the scene to approximate the sensing process of a LiDAR, and intersections with scene geometry are used to generate simulated returns.

with surrounding geometry, as observed in Figure 2.1. For each emitted ray, the simulator determines whether it hits an object, where the intersection occurs, and how far that point is from the sensor. By repeating this process over many directions, a point cloud representation of the environment can be generated.

In LiDAR simulation, ray casting provides an efficient way to approximate the geometric measurement process of the sensor. It captures the basic behavior that LiDAR beams travel through space, interact with surfaces, and return spatial information about the scene. Because of its simplicity and computational efficiency, ray casting is widely used as the foundation of real-time LiDAR simulation. In this thesis, it serves as the basis for generating simulated LiDAR returns before additional effects such as intensity response are considered.

2.6 Spherical Projection

Spherical projection is a way of representing LiDAR measurements in a structured form by mapping each point according to its direction relative to the sensor. Instead of treating the point cloud only as an unordered set of 3D points, the measurements are projected onto a sphere centered at the LiDAR. In this representation, each return is described by angular coordinates together with an associated value, such as range or intensity.

This transformation is useful because raw point clouds are irregular and unstructured, which makes direct global comparison difficult. By projecting the data onto a sphere, the measurements can be organized into a consistent directional representation that preserves how the environment appears from the sensor viewpoint. This makes it easier to compare observations at the distribution level rather than relying on exact point-to-point correspondence.

In this thesis, spherical projection is used to convert LiDAR observations into a structured representation suitable for global comparison and later spectral analysis. In this way, the method becomes less sensitive to local geometric mismatch while still preserving the overall directional structure of the measured signal.

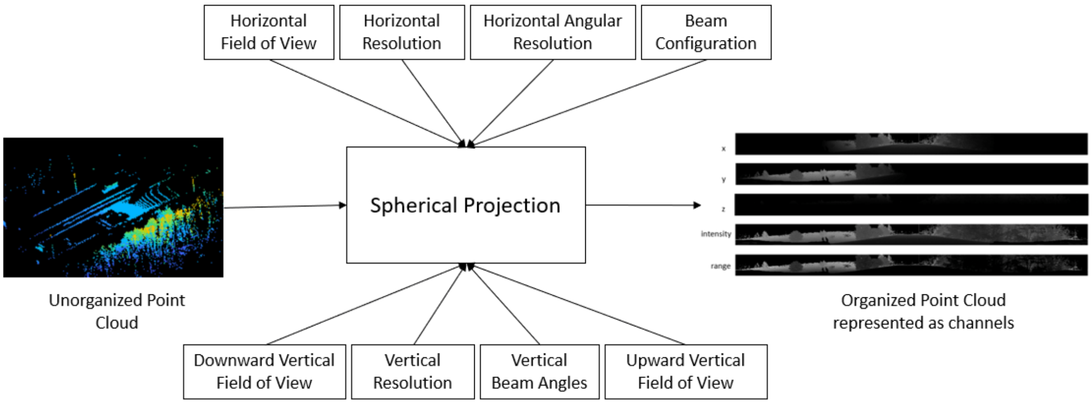


Figure 2.2: Illustration of spherical projection for LiDAR data, adapted from MathWorks MATLAB Lidar Toolbox documentation: *Unorganized to Organized Conversion of Point Clouds Using Spherical Projection*, `https://www.mathworks.com/help/lidar/ug/unorganized-to-organized-pointcloud-conversion.html`. Each 3D point is mapped onto a 2D image according to its angular coordinates (θ, ϕ) relative to the sensor, while the pixel value represents a quantity such as intensity or range.

2.7 Spherical Harmonics

Spherical harmonics (SH) are a set of orthogonal basis functions defined on the unit sphere. They are commonly written as

$$Y_l^m(\theta, \phi) = N_l^m \, P_l^m(\cos\theta) \, e^{im\phi}, \tag{2.1}$$

where $l \geq 0$ is the degree, $-l \leq m \leq l$ is the order, P_l^m denotes the associated Legendre polynomial, and N_l^m is a normalization constant.

A signal defined on the sphere, denoted by $f(\theta, \phi)$, can be expanded in the SH basis as

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} c_l^m \, Y_l^m(\theta, \phi), \tag{2.2}$$

where c_l^m are the spherical harmonic coefficients.

In this thesis, spherical harmonics are used to represent LiDAR signals after spherical projection in a compact spectral form. Instead of comparing individual points

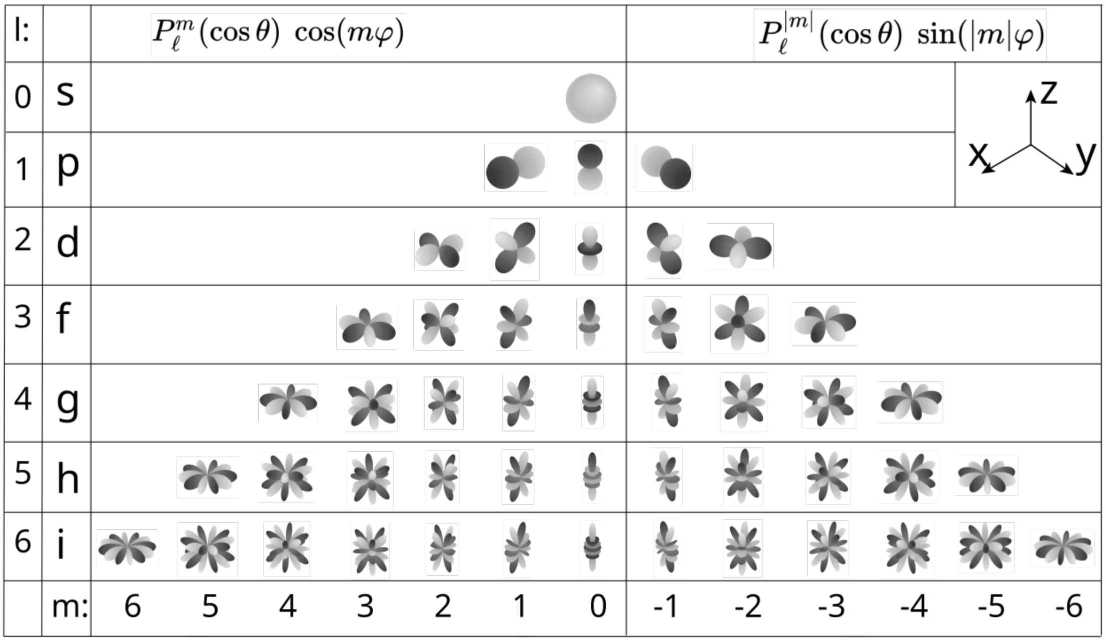


Figure 2.3: Visualization of real spherical harmonic basis functions of increasing degree and order. The figure is adapted from Wikipedia, *Spherical harmonics*: `https://en.wikipedia.org/wiki/Spherical_harmonics`.

directly, each LiDAR frame is described by its SH coefficients. This is useful because descriptors derived from these coefficients can be made rotation-invariant, which enables more robust global comparison between LiDAR frames.

2.8 Sim-to-Real Gap

The sim-to-real gap refers to the discrepancy between simulated data and real-world observations. In sensor simulation, this gap arises because a virtual environment cannot perfectly reproduce all aspects of the real sensing process, even when the scene geometry and sensor configuration are similar. As a result, simulated outputs may differ from real measurements in ways that affect both the sensor signal itself and the behavior of downstream algorithms [19].

In the context of LiDAR simulation, the sim-to-real gap is influenced by multiple factors, including geometric fidelity, material modeling, scanning effects, and other sensor-dependent phenomena. Recent work has shown that reducing this gap requires more than reproducing approximate scene structure alone; realistic modeling of sensor behavior and scene properties is also important for obtaining simulated LiDAR data that better matches real observations.

For this reason, the sim-to-real gap is a central issue in simulation-based development and evaluation. It motivates both the improvement of sensor models and the design of comparison methods that remain meaningful even when exact agreement between simulation and reality cannot be achieved.

3

Methods

The methodology in this thesis follows a sim-to-real workflow in which LiDAR data collected at the AstaZero proving ground are compared with LiDAR data generated in a CARLA-based digital twin of the same test site. The real-world dataset is stored in MCAP format and contains timestamped sensor and localization data, including RGB camera images, GNSS measurements, and LiDAR point clouds. These real LiDAR measurements are used as reference data for evaluating the realism of the simulated LiDAR intensity.

In the simulation environment, a standalone LiDAR sensor is spawned in the AstaZero digital twin. The simulated LiDAR point cloud is generated through ray casting, where rays are emitted from the sensor according to the configured scan pattern and intersections with the virtual environment are recorded. The proposed intensity model is integrated into this generation process and assigns an intensity value to each valid LiDAR return based on physically motivated factors such as range, incidence angle, and surface reflectance.

The simulated and real LiDAR data are then compared by extracting frames from corresponding locations or scenarios. Since exact point-wise alignment between the real and simulated point clouds is difficult to achieve, the evaluation focuses on geometry-tolerant comparison of intensity characteristics rather than strict one-to-one matching. This workflow allows the proposed intensity model to be assessed in terms of how closely the simulated LiDAR intensity resembles the real-world measurements.

3.1 Reference LiDAR Data

3.1.1 Real-World Data Collection and MCAP Recording

The real-world reference data used in this thesis were collected at the AstaZero proving ground using an instrumented vehicle equipped with multiple perception and localization sensors. The recorded data are stored in MCAP files, which provide a structured container format for storing timestamped messages from different sensors and software modules. In this work, the MCAP recordings are used as the source of the real-world LiDAR measurements against which the simulated LiDAR intensity is evaluated.

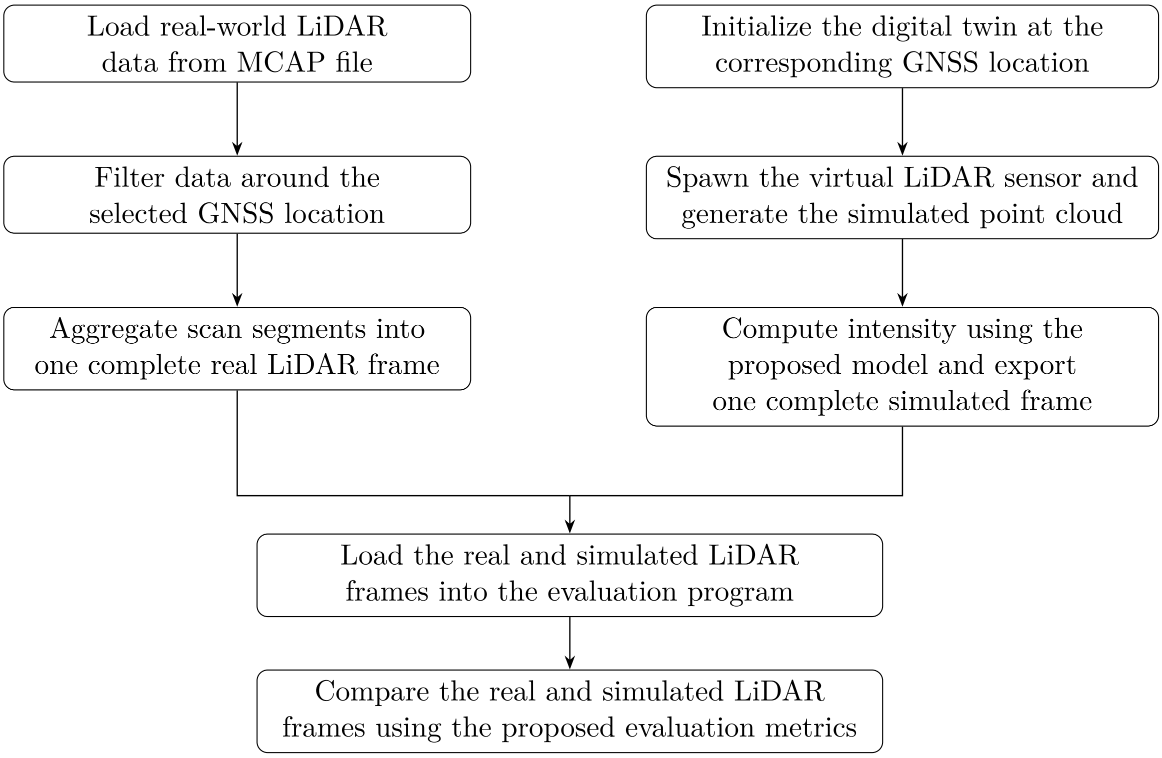


Figure 3.1: Compact overview of the proposed sim-to-real evaluation pipeline. A real LiDAR frame is reconstructed from recorded MCAP data, while a corresponding simulated frame is generated in the digital twin using the proposed intensity model. The two frames are then loaded into the evaluation program and compared using the selected metrics.

An MCAP file [13] organizes recorded data using three main concepts: schemas, channels, and messages. A schema describes the structure of a certain type of message, for example which fields are present in the data and how they should be interpreted. A channel connects such a message type to a specific recorded topic, such as a LiDAR, camera, or localization stream, and specifies how messages on that topic are encoded. The messages themselves contain the actual recorded data, each associated with timing information and a channel. This structure allows heterogeneous sensor data to be stored in the same file while preserving the relationship between topic, data type, encoding, and timestamp.

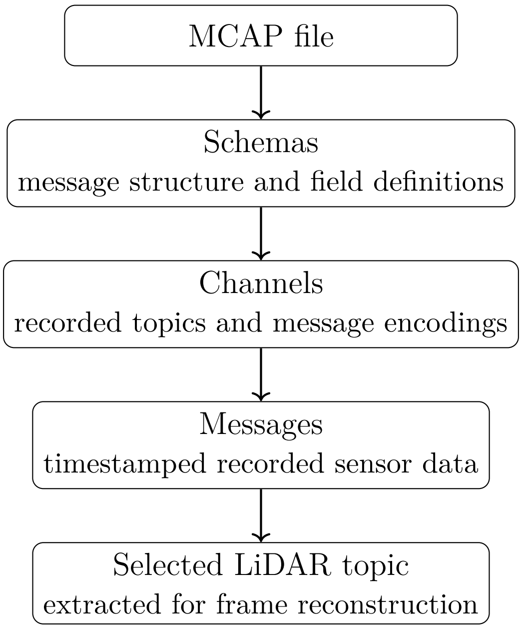


Figure 3.2: Simplified organization of the MCAP recording format used for LiDAR data extraction.

Although the MCAP recordings contain several sensor modalities, including camera, GNSS, IMU, radar, and LiDAR data, the current intensity evaluation pipeline only extracts the LiDAR stream. The LiDAR measurements are stored as PointCloudSliver messages, which represent partial point cloud data produced by the sensor during a scan. These sliver messages are vendor-specific and must therefore be decoded and reconstructed according to the corresponding LiDAR data layout. The reconstruction of complete LiDAR frames from these messages is described in the following subsection.

3.1.2 Vendor-Specific LiDAR Data Structure

Although the LiDAR data stored in the MCAP recordings are already provided as PointCloudSliver messages rather than raw Ouster UDP packets, the meaning of the intensity field is still determined by the original Ouster sensor data format. In the extracted point cloud representation, each point is available in an XYZI format, where the spatial coordinates describe the measured 3D position and the intensity value originates from the corresponding Ouster return measurement. Therefore, before using the recorded LiDAR data as reference data, it is necessary to clarify which Ouster quantity is represented by the intensity field.

Ouster sensors support different LiDAR return profiles. The relevant intensity interpretation in this work is described using the single-return `RNG19_RFL8_SIG16_NIR16` profile [14] as a representative example. As illustrated in Figure 3.3, this profile contains several measurement fields for each channel data block, including range, calibrated reflectivity, signal photons, and near-infrared photons. These fields represent different physical quantities. Range describes the measured distance to the detected target, signal photons represent the strength of the laser return, and near-infrared photons are related to ambient near-infrared illumination. The calibrated reflectivity field provides a sensor-processed estimate of target reflectivity based on the received signal, measured range, and sensor sensitivity.

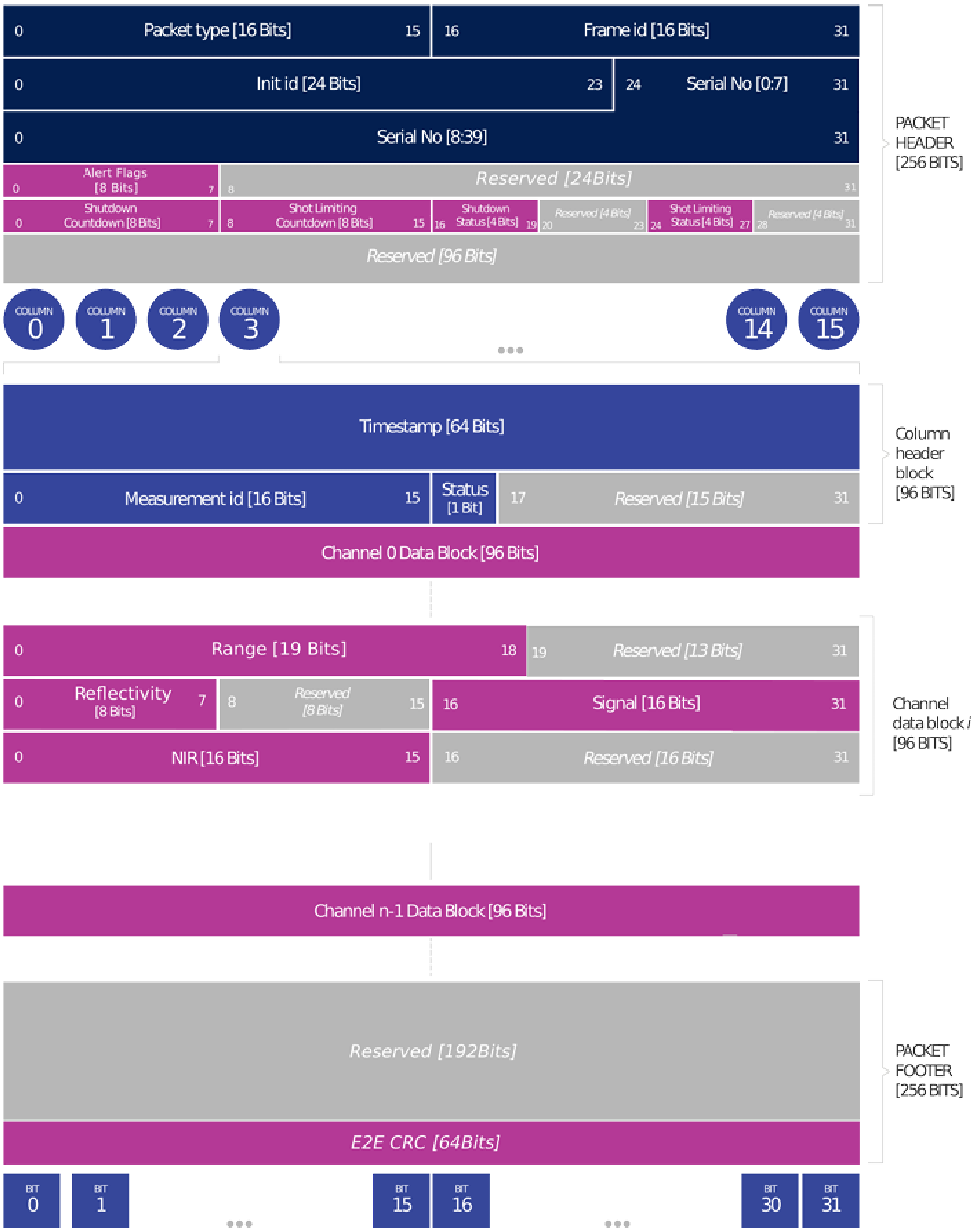


Figure 3.3: Example Ouster channel data structure for the `RNG19_RFL8_SIG16_NIR16` single-return profile. The profile contains range, calibrated reflectivity, signal, and near-infrared fields for each channel data block.

In the reconstructed point clouds used in this thesis, the intensity component is interpreted as the Ouster calibrated reflectivity field, denoted by `RFL8`. This field is stored as an 8-bit unsigned integer with values from 0 to 255. However, the integer value is an encoded representation rather than the calibrated reflectivity value itself. According to the Ouster reflectivity representation, each integer maps to a corresponding calibrated reflectivity level. The mapping is approximately linear for lower values, while higher integer values represent increasingly larger reflectivity levels.

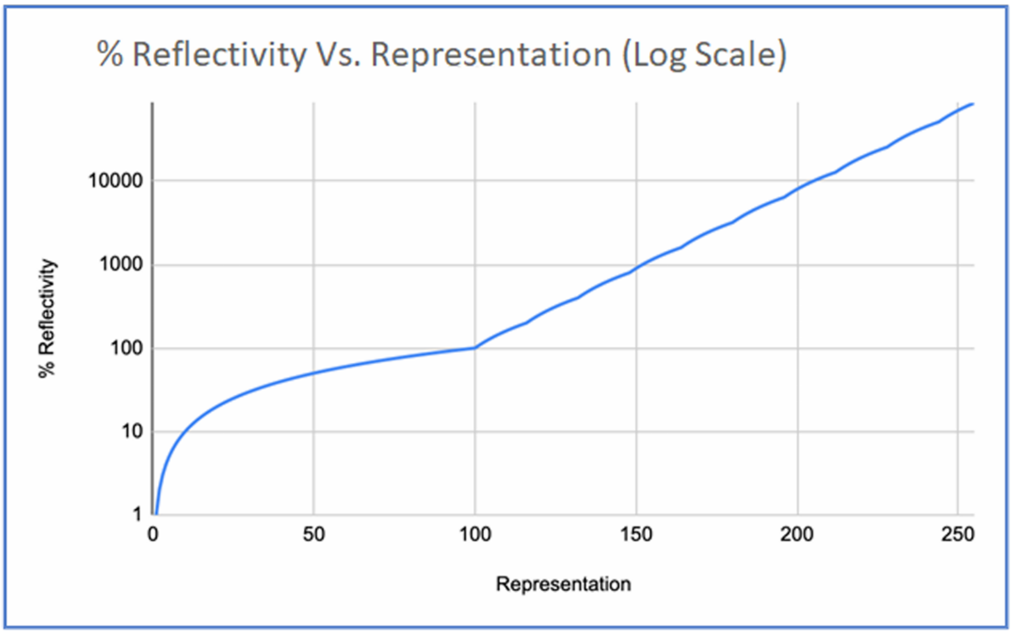


Figure 3.4: Mapping between the 8-bit reflectivity representation and calibrated reflectivity values for the Ouster `RFL8` field, shown on a logarithmic scale. [15]

The calibrated reflectivity should not be interpreted as raw received signal strength. Instead, it is a sensor-processed quantity derived from the signal photon measurement and compensated based on factors such as measured range and sensor sensitivity. Therefore, the calibrated reflectivity represented by `RFL8` is used as the real-world reference quantity in this thesis. The objective of the proposed simulation model is to approximate this calibrated reflectivity behavior in the simulated LiDAR point cloud, while the subsequent processing operates on the XYZI point cloud representation extracted from the MCAP recordings.

3.1.3 Point Cloud Frame Reconstruction

The MCAP recordings provide the LiDAR measurements as PointCloudSliver messages. Each sliver contains a partial point cloud from a selected LiDAR topic, together with metadata required to decode the binary point buffer. Since one sliver only represents part of the scan, a complete LiDAR frame is reconstructed by decoding and concatenating multiple consecutive slivers from the same topic.

The reconstruction process starts by filtering the MCAP messages to the selected LiDAR topic. For each matching sliver, the message metadata are read to determine the point stride and the layout of the packed point fields. The field definitions specify the name, offset, and numeric type of each point attribute, allowing the raw data buffer to be interpreted as a structured point array. The decoded points are represented in XYZI format, where x, y, and z describe the Cartesian position of each point and i contains the recorded intensity value.

In the MCAP PointCloudSliver representation, the intensity value is stored as a normalized reflectivity value rather than directly as the original 8-bit `RFL8` representation. To recover the representation used by the Ouster reflectivity mapping, the normalized sliver intensity is converted back to the 0–255 scale as

$$R_{\mathrm{RFL8}} = I_{\mathrm{sliver}} \cdot \frac{255}{100} \tag{3.1}$$

Invalid or near-zero intensity values are removed before this reconstructed intensity is used for analysis. After decoding, filtering, and intensity conversion, the selected sequence of slivers is concatenated into a single LiDAR frame. The output of this step is a reconstructed real-world point cloud containing Cartesian coordinates and corresponding `RFL8`-based intensity values, which are used as the reference data for later sim-to-real evaluation.

3.2 Simulated LiDAR Data

This section describes how the simulated LiDAR data are generated in the CARLA-based digital twin. The simulation first defines the virtual environment and sensor configuration, then generates LiDAR returns through ray casting. For each valid ray-cast hit, an intensity value is computed using the proposed simulated intensity model. The resulting simulated point cloud is later compared with the reconstructed real-world reference data.

3.2.1 Simulation Environment and Virtual Sensor Setup

The simulated LiDAR data used in this thesis are generated in a CARLA-based simulation environment built on Unreal Engine. The environment contains a digital twin of the AstaZero proving ground, allowing the same test site to be represented in simulation as in the real-world data collection. In a standard approach, an ego vehicle would be spawned together with the surrounding static and dynamic objects required for the selected scenario, and the developed LiDAR sensor would be attached to the vehicle at a predefined mounting position corresponding to the physical sensor installation. This setup enables controlled generation of LiDAR measurements under repeatable conditions and provides the basis for later sim-to-real comparison.

However, because the Functional Mock-up Units (FMUs) of the vehicle models are still undergoing concurrent testing and validation during the development of the LiDAR model, a decoupled approach is implemented. Instead of mounting the virtual LiDAR directly to the vehicle model, the sensor is spawned as an independent entity within the virtual environment. Its spatial trajectory is defined using absolute positions extracted from MCAP log files, which are calculated by applying the coordinate transforms from the ego vehicle's localized pose to the relative physical position of the sensor. The sensor is configured to follow the intended scan pattern and measurement characteristics of the real LiDAR as closely as required for the evaluation. In practice, this includes parameters such as the number of channels, rotational behavior, angular sampling, minimum and maximum range, and frame generation rate. The purpose of this configuration is to ensure that the simulated point cloud has a comparable spatial sampling structure to the real-world reference data.

Figure 3.5 illustrates an example CARLA simulation scene with a virtual LiDAR output. The background shows a simulated driving environment rendered in CARLA, while the overlaid window visualizes the point cloud generated by the virtual LiDAR sensor. This illustrates the general ray-casting-based sensor simulation setup used to generate the simulated point cloud data.

Figure 3.5: Example CARLA simulation scene and corresponding virtual LiDAR output. The overlaid point cloud view shows the LiDAR returns generated from the simulated environment.

3.2.2 Ray-Casting-Based LiDAR Generation

The virtual LiDAR sensor generates point clouds by emitting rays into the simulated scene according to the configured scan pattern. At each simulation step, the number of scan columns to be generated is determined from the elapsed time, rotation frequency, horizontal resolution, and horizontal field of view. Each column defines one horizontal emission angle, while each LiDAR channel defines one vertical emission angle. Together, these angles determine the direction of each ray.

For every ray, a ray-cast query is performed in the Unreal Engine scene. If the ray intersects an object within the valid range, the hit is treated as a LiDAR return and its geometric information, such as hit position, range, surface normal, and semantic tag are used to output the calibrated reflectivity values according to the model. If no valid intersection is found, the return is marked as null calibrated reflectivity with relative position to the LiDAR sensor $XYZ = (0, 0, 1)$. The returns from all generated columns and channels are then assembled into a simulated LiDAR frame. This ray-casting procedure provides the geometric basis for the intensity model introduced in the following subsection.

Algorithm 1 Ray-casting-based LiDAR generation

```
Require: LiDAR configuration, sensor pose, simulation time step
Ensure: Simulated LiDAR frame
Determine the number of scan columns generated during the time step
for each scan column do
    Compute the horizontal emission angle
    for each vertical channel do
        Read the vertical emission angle
        Construct the ray direction from the horizontal and vertical angles
        Cast the ray into the simulated scene
        if the ray intersects an object within the valid range then
            Extract hit position, range, surface normal, and semantic tag
            Calculate calibrated reflectivity according to the model
            Store the return as a LiDAR point
        else
            Mark the return as XYZI = (0, 0, 1, 0)
        end if
    end for
end for
Assemble the returns into the simulated LiDAR frame
```

3.2.3 Physically Motivated Intensity Model

LiDAR intensity is affected by several physical and sensor-specific factors. Previous work on LiDAR radiometric processing and intensity correction commonly identifies range, incidence angle, target reflectance, atmospheric effects, and sensor-dependent processing as important contributors to the measured return strength [1, 2]. In automotive LiDAR simulation, material reflectance and incidence-angle-dependent response have also been used to model more realistic LiDAR returns [3]. Based on these observations, the model used in this thesis is formulated as a simplified real-time approximation rather than a full reconstruction of the complete radiometric and sensor-processing pipeline.

For each valid ray-cast hit, a physically motivated return term is first computed from three components: range-dependent attenuation, incidence-angle response, and material reflectivity. The base return model is written as

$$I_{\text{phys}} = A(r)\rho_m(\theta) \tag{3.2}$$

where r is the range from the LiDAR sensor to the hit point, $A(r)$ is the range-dependent attenuation term, θ is the incidence angle between the incoming ray direction and the surface normal, and ρ_m is the reflectivity assigned to the hit semantic material depending on the incidence angle.

The physical return term is not used directly as the final simulated intensity. The real-world reference data in this thesis are based on the Ouster `RFL8` calibrated reflectivity representation, which is a sensor-processed quantity rather than raw received signal strength. Since the exact internal Ouster processing is not available, an additional sensor-specific approximation function is applied:

$$R_{\text{sim}} = F_{\text{Ouster}}\left(I_{\text{phys}}\right) \tag{3.3}$$

Here, F_{Ouster} denotes an empirical scaling and normalization function that maps the physically motivated return term to an Ouster-like simulated calibrated reflectivity representation. The following subsections describe the individual terms $A(r)$ and $\rho_m(\theta)$, followed by the sensor-specific scaling and normalization used to obtain the final simulated calibrated reflectivity.

3.2.3.1 Range-Dependent Attenuation

The strength of a LiDAR return generally decreases with increasing distance between the sensor and the target due to propagation effects such as beam spreading and atmospheric attenuation. Mathematically, this behavior is represented by a range-dependent attenuation term $A(r)$:

$$A(r) = \frac{1}{r^n}\exp(-\alpha r) \tag{3.4}$$

where r is the distance from the LiDAR sensor to the ray-cast hit point and α is the atmospheric attenuation coefficient. The exponent n dictates the geometric power drop-off, which conventionally varies from $n = 2$ for extended Lambertian targets to $n = 4$ for distinct point targets, depending on the spatial fraction of the laser footprint intercepted by the object [3].

However, the studied Ouster sensor outputs a calibrated reflectivity metric. The sensor's internal firmware dynamically compensates for these distance-related energy losses ($1/r^n$) based on real-time range measurements, normalizing the output to represent a material property rather than a raw signal intensity. Consequently, to maintain parity during later sim-to-real comparisons, the explicit $1/r^n$ geometric attenuation term is omitted from the thesis simulation model. Instead, the simulation maps target returns directly to calibrated reflectivity values, preserving atmospheric attenuation $\exp(-\alpha r)$ for long-range environmental fidelity.

3.2.3.2 Semantic Material and Angle-dependent Material Reflectance

Different physical surfaces produce different LiDAR return intensities due to their material-specific reflectance properties and the orientation of the surface relative to the sensor. In the proposed model, each valid ray-cast hit within the simulation environment is linked to a unique semantic tag, which categorizes the surface into specific material classes such as asphalt, lane markings, vegetation, concrete, or metal.

Instead of assigning a single constant reflectivity value to each material, in order to map these semantic categories to realistic physical responses, an open-access angle dependent spectral library [16] based on a 945 nm Time-of-Flight Camera is used to incorporate empirical data into the model. This database provides discrete, angle-resolved material reflectance measurements done at 10° increments.

The material reflectivity term is modeled as a function of the incidence angle θ, defined as the angle between the incoming laser beam and the surface normal at the hitpoint, and the semantic material class m:

$$\rho_m = f(m, \theta) \tag{3.5}$$

Because the simulation environment samples arbitrary continuous incidence angles that can differ from the database's discrete 10° increments, the computed material reflectivity $\rho_m(\theta)$ for a given semantic label m and an incidence angle $\theta \in [0°, 90°]$ is determined according to the following formulation:

$$\rho_m(\theta) = \begin{cases} \cos(\theta), & \text{if } m \notin \mathcal{M} \\ \dfrac{1}{100}\left[(1-\beta)\rho_i + \beta\rho_{i+1}\right], & \text{if } m \in \mathcal{M} \text{ and } 0° \le \theta < 80° \\ \dfrac{1}{100}\left[(1-\beta)\rho_8\right], & \text{if } m \in \mathcal{M} \text{ and } 80° \le \theta \le 90° \end{cases} \tag{3.6}$$

where $\mathcal{M}$ represents the set of available materials within the spectral library database. The indexing and interpolation weights are calculated dynamically based on the region of the incidence angle:

- **For the standard interpolation region** ($0^\circ \leq \theta < 80^\circ$)**:** The bounding lower bin index is $i = \lfloor \theta/10 \rfloor$, the upper bin index is $i+1$, and the interpolation weight is defined as $\beta = (\theta/10) - i$. The values ρ_i and ρ_{i+1} correspond to the percentage reflectance values stored at these indices.
- **For the boundary extrapolation region** ($80^\circ \leq \theta \leq 90^\circ$)**:** The boundary index is fixed at the maximum available dataset entry ($i = 8$, representing 80°), and the linear fade weight to absolute zero at 90° is computed as $\beta = (\theta - 80)/10$.

By combining semantic classification with angle-dependent database lookups, the simulation successfully reproduces variations in intensity across diverse materials while accounting for incidence-angle dependencies. This lookup and interpolation pipeline provides a practical, computationally lightweight approximation suitable for real-time LiDAR simulation without requiring the deployment of full bidirectional reflectance distribution functions (BRDFs).

3.2.4 Sensor-Specific Empirical Distribution Mapping

The physically based return term I_{phys} generated within the simulation environment represents a relative simulated return strength, but it is not directly equivalent to the Ouster `RFL8` calibrated reflectivity value. The `RFL8` value is a vendor-specific representation that is produced after proprietary internal processing, range compensation, sensitivity correction, and non-linear encoding. Since the exact parameters of this internal processing pipeline are inaccessible, the simulated physical return cannot be converted to R_{RFL8} using a known analytical sensor model.

For this reason, the final simulated calibrated reflectivity is obtained through an empirical distribution mapping rather than a purely physical formula. Let I_{phys} denote the physically motivated simulated return value and let R_{RFL8} denote the calibrated reflectivity representation used in the real-world reference data. The objective of the mapping is to transform the distribution of I_{phys} so that it becomes directly comparable to the distribution of R_{RFL8}:

$$R_{\text{sim}} = F_{\text{Ouster}}\left(I_{\text{phys}}\right) \tag{3.7}$$

where the mapping function F_{Ouster} is estimated empirically using regression over quantile correspondences across the simulated and real domains.

Let x denote the distribution of simulated physical intensity samples, and let y denote the distribution of real calibrated reflectivity samples. For each matched frame pair, a set of matching quantiles is extracted across a predefined range $[q_{\min}, q_{\max}]$. These matching points are pooled into a dataset of training pairs (x_q, y_q), establishing an empirical distribution profile between the two domains. A set of candidate parametric model families, H, is then evaluated to fit these quantile pairs:

$$y_q \approx h(x_q; \mu_h), \quad h \in H, \tag{3.8}$$

where μ_h represents the parameter vector for the candidate model h. The library of candidate models H contains six parametric families: linear, quadratic, cubic, logarithmic, power, and exponential-saturating functions.

The optimal parameter vector $\hat{\mu}_h$ for each candidate model is estimated through non-linear least-squares curve fitting using the `scipy.optimize.curve_fit` function provided by the SciPy library [17]. The optimal mapping model $h^\star$ is subsequently selected by identifying the candidate that minimizes the Mean Squared Error (MSE) over the aggregated quantile dataset:

$$h^\star = \arg\min_{h \in H} (\mathrm{MSE}_h) \tag{3.9}$$

$$F_{\mathrm{Ouster}}(I_{\mathrm{phys}}) = h^\star(I_{\mathrm{phys}}; \hat{\mu}_{h^\star}) \tag{3.10}$$

The final simulated calibrated reflectivity representation R_{sim} is obtained by passing the raw simulated intensity through the chosen function, where non-physical negative values are suppressed via a zero-clipping boundary condition:

$$R_{\mathrm{sim}} = \max\left(0, F_{\mathrm{Ouster}}(I_{\mathrm{phys}})\right). \tag{3.11}$$

The resulting value is therefore not a reconstruction of the internal Ouster processing pipeline. Instead, it is a statistically simulated calibrated reflectivity value whose distribution is aligned with the calibrated reflectivity representation observed in the real-world reference data.

3.2.5 Cross-Scenario Generalization and Validation Framework

To prevent the empirical distribution mapping from overfitting to a single localized scene layout or specific weather state, a cross-scenario generalization and validation framework was established. The main objective of this framework is to evaluate whether a physics-motivated mapping function can generalize across different environmental conditions by executing an average of separate empirically fitted curves. Specifically, this framework allows us to observe and compare whether a calibration model isolated to a restricted, controlled field of view yields a superior, more robust approximation of the sensor transfer function on unseen data compared to a full-field-of-view configuration.

The experimental design divides the available simulation and real-world log files into two distinct subsets: a calibration dataset and an unseen validation dataset. Both subsets feature opposite weather and surface states:

- **Calibration Dataset:** Comprises Scenario A, which represents dry environmental conditions, and Scenario B, which captures a wet/rainy environment.
- **Validation Dataset:** Comprises Scenario C (dry conditions) and Scenario D (wet/rainy conditions). This pair is not used during the initial optimization phase to serve as a baseline for testing the generalization capabilities of the obtained mapping.

Additionally, the chosen sequences are filtered so that elements such as vehicles, buildings, or dense tree vegetation do not significantly distort or skew the curve-fitting process and subsequent evaluations. These structural elements are intentionally excluded because they introduce severe sim-to-real discrepancies: simulated vehicles are restricted to a single, uniform semantic material tag, a simplification that fails to capture the complex multi-material composition like paint, glass, metal, rubber, of real-world vehicles, while assets like buildings and trees lack exact digital-twin geometric fidelity and their true real-world material specifications are fundamentally unknown.

Hence, focusing the calibration on the ground plane, mainly composed of gravel and soil within the drivable area, and low ground vegetation in the non-drivable areas, provides a highly stable dataset where material characteristics can be accurately assumed and modeled, ensuring that the physical reality gap between the simulation and the real logs remains minimized. This isolation allows the analysis to focus directly on reliable material reflectance properties. Within the simulation environment, the semantic material assignments are strictly controlled so that the exact same materials are present across all scenarios. The different environmental conditions are introduced by swapping the experimental lookup tables linked to these material labels: the dry scenarios (A and C) utilize the corresponding dry material measurements, while the wet scenarios (B and D) apply the wet material measurement values to those identical target classes.

For both calibration scenarios, the raw point clouds from the front-center LiDAR are processed under two different spatial fields of view (FOVs) to evaluate domain stability:

1. **Reduced FOV Configuration:** Restricts the spatial window to a centralized horizontal span of 70° and the lower 60 vertical channels. The primary objective of this configuration is to focus the vast majority of the transmitted laser beams directly onto the ground surface. By targeting a uniform surface region and removing peripheral data, this setup isolates a data subset that closely replicates the controlled real-world laboratory settings used for empirical material measurements, a working environment which was not accessible during the development of the thesis.
2. **Full FOV Configuration:** Utilizes the 180° horizontal span directly in front

of the sensor across all 64 vertical channels. While a broader FOV increases sample size, it introduces a variety of physical variables. Under full exposure, the recorded intensity is no longer purely a function of the known material type and local angle of incidence. Instead, it is heavily influenced by range-dependent distance attenuation and geometric structural differences across the scene.

By executing the empirical distribution mapping across these configurations, the cubic polynomial family was identified as the optimal minimum-MSE model $h^\star \in H$ across all calibration instances. This yields four independent sets of polynomial coefficients.

Because polynomial equations are linear with respect to their parameter vectors, a direct arithmetic mean of their corresponding coefficients was carried out to obtain the generalized/master mapping functions. Thus, the master full-FOV mapping function is expressed as a single cubic equation:

$$\bar{F}^{\text{full}}_{\text{Ouster}}(I_{\text{phys}}) = \bar{a}^{\text{full}} I^3_{\text{phys}} + \bar{b}^{\text{full}} I^2_{\text{phys}} + \bar{c}^{\text{full}} I_{\text{phys}} + \bar{d}^{\text{full}} \tag{3.12}$$

where each coefficient is defined as the average of the scenario-specific parameters:

$$\bar{a}^{\text{full}} = \frac{a^{\text{full}}_A + a^{\text{full}}_B}{2}, \quad \bar{b}^{\text{full}} = \frac{b^{\text{full}}_A + b^{\text{full}}_B}{2} \tag{3.13}$$

$$\bar{c}^{\text{full}} = \frac{c^{\text{full}}_A + c^{\text{full}}_B}{2}, \quad \bar{d}^{\text{full}} = \frac{d^{\text{full}}_A + d^{\text{full}}_B}{2} \tag{3.14}$$

The master reduced-FOV mapping function $\bar{F}^{\text{red}}_{\text{Ouster}}(I_{\text{phys}})$ is formulated symmetrically using the averaged coefficients derived from the isolated $70^\circ \times 60$ channel configurations.

These two master/averaged curves establish fixed, time-invariant calibration profiles. During the subsequent testing and validation phase, these curves are used directly in the developed model and deployed directly onto the unseen validation scenarios (Scenarios C and D) to convert simulated physical intensities into final simulated calibrated reflectivities, R_{sim}. The accuracy and structural alignment of the resulting point clouds are then quantified using the evaluation method detailed in the subsequent subsection.

Table 3.1 shows the matrix of scenarios and sensor configurations used during the generalization and validation framework.

Table 3.1: Matrix showing the Vendor-Specific Empirical Mapping Framework

Condition	Fitting Scenario	Validation Scenario
Dry	**Scenario A** • Full FOV • Reduced FOV	**Scenario C** • Evaluated with Master fitting curves
Wet	**Scenario B** • Full FOV • Reduced FOV	**Scenario D** • Evaluated with Master fitting curves

3.3 Evaluation Method

Direct point-wise comparison between real and simulated LiDAR point clouds is highly sensitive to localization errors, geometric mismatch, and sampling differences. In principle, the simulated LiDAR sensor can be placed at a pose corresponding to the real measurement in the digital twin. In practice, however, it is difficult to reproduce the exact sensor pose, vehicle state, scan timing, and surrounding geometry with sufficient accuracy for strict one-to-one point matching. Small pose errors can shift many points, causing large apparent differences even when the simulated intensity behavior is qualitatively similar to the real measurement.

Geometric mismatch is also unavoidable in this thesis. First, the AstaZero digital twin was created from scanned data collected several years before the real-world LiDAR recordings used in this work. Although the main road layout and static infrastructure remain similar, smaller objects and scene details may have changed over time. Second, the geometry used for real-time ray casting is simplified and organized for efficient BVH-based intersection queries. This simplified representation is suitable for simulation performance, but it cannot be expected to reproduce every geometric detail visible to the real LiDAR sensor. Therefore, the evaluation method should not require exact geometric correspondence between individual real and simulated points.

To address these limitations, this thesis evaluates LiDAR intensity using a geometry-tolerant frame-level descriptor based on spherical harmonic approximation. Each LiDAR frame is first represented as an angular intensity function on the sphere. Instead of comparing individual points, the method approximates the frame-level intensity distribution using smooth spherical harmonic basis functions. This reduces sensitivity to local geometric differences and sharp pointwise mismatches. The comparison is then performed using the energy at each spherical harmonic degree, which is invariant to rotation. As a result, the proposed evaluation method compares the overall angular structure of the intensity distribution while being tolerant to small pose differences and geometric inconsistencies between the real scene and the digital twin.

3.3.1 Spherical Intensity Representation

The evaluation method first transforms each LiDAR frame from Cartesian point cloud form into a spherical intensity representation. A LiDAR point is originally represented in XYZI format as

$$\mathbf{p}_k = (x_k, y_k, z_k, i_k) \tag{3.15}$$

where (x_k, y_k, z_k) is the 3D position of point k in the LiDAR coordinate frame and i_k is its intensity value. In the evaluated data, the vehicle motion during one LiDAR frame is small because the truck drives at low speed while the LiDAR frame rate is relatively high. Therefore, each frame is approximated as being captured from a fixed sensor origin. Under this assumption, each point can be represented by its angular direction from the LiDAR origin, which provides a natural representation for comparing the frame-level intensity distribution.

For each valid point, the Cartesian coordinates are converted to spherical coordinates as

$$r_k = \sqrt{x_k^2 + y_k^2 + z_k^2} \tag{3.16}$$

$$\theta_k = \arccos\left(\frac{z_k}{r_k}\right) \tag{3.17}$$

$$\phi_k = \mathrm{mod}\left(\mathrm{atan2}(y_k, x_k), 2\pi\right) \tag{3.18}$$

where r_k is the range, $\theta_k \in [0, \pi]$ is the polar angle measured from the positive z-axis, and $\phi_k \in [0, 2\pi)$ is the azimuth angle. Points with zero or invalid range are removed before this transformation.

After this conversion, each LiDAR frame is represented as a set of angular intensity samples,

$$\mathcal{S} = \{(\theta_k, \phi_k, i_k)\}_{k=1}^{N} \tag{3.19}$$

where N is the number of valid points in the frame. This can be interpreted as samples of an intensity function on the sphere,

$$I(\theta_k, \phi_k) = i_k \tag{3.20}$$

The subsequent spherical harmonic approximation is applied to this angular intensity function rather than to the original Cartesian point positions. This makes the comparison focus on the directional distribution of intensity in the LiDAR frame, instead of requiring exact point-wise correspondence in 3D space. Figure 3.6 gives a visual representation of how the cartesian coordinates are transformed into the spherical projection.

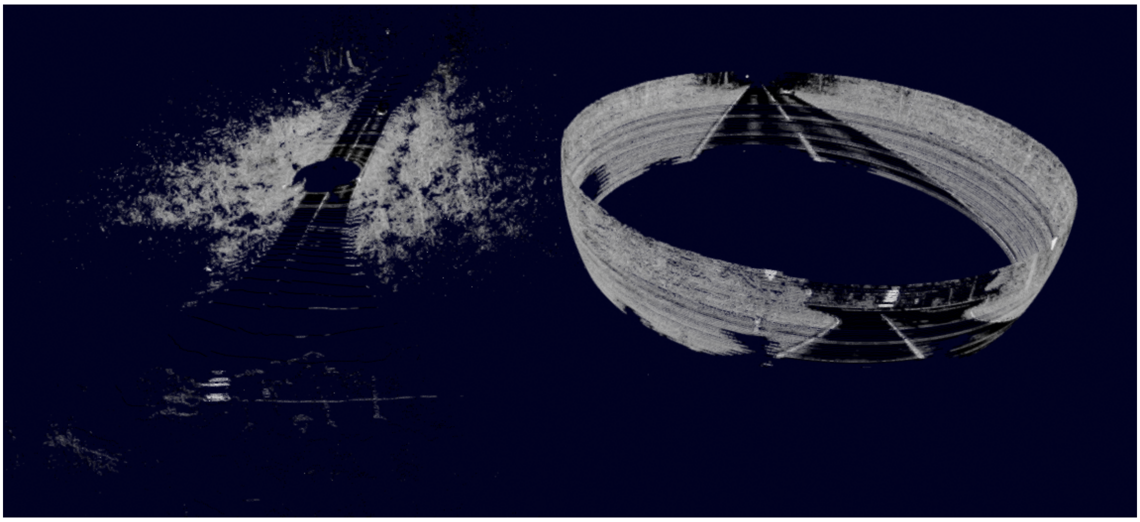


Figure 3.6: Illustration of spherical projection for LiDAR data. The left side shows the original LiDAR point cloud in Cartesian space, while the right side shows the same data mapped onto the spherical domain according to the angular coordinates of each point. This representation preserves the directional structure of the LiDAR frame and provides the basis for subsequent spherical harmonic analysis.

3.3.1.1 Function Space and Inner Product

The spherical intensity representation allows one LiDAR frame to be interpreted as samples of a scalar function $f(\theta, \phi)$ defined on the sphere. To approximate this function systematically, the set of square-integrable real-valued functions on the sphere is considered as a vector space, denoted by $L^2(\mathbb{S}^2)$. In this space, functions play the role of vectors: they can be added, multiplied by scalar coefficients, and represented using suitable basis functions.

In a vector space, an arbitrary element can be represented by a weighted combination of basis elements. Similarly, a function f on the sphere can be approximated by a finite expansion using K basis functions,

$$f(\theta, \phi) \approx \sum_{j=1}^{K} c_j \varphi_j(\theta, \phi) \tag{3.21}$$

where φ_j denotes the j-th basis function and c_j is its corresponding coefficient. To determine these coefficients, an inner product is introduced. The inner product defines projection and similarity in the function space, analogous to the dot product in a finite-dimensional vector space.

If the basis functions are orthonormal, they satisfy

$$\langle \varphi_j, \varphi_k \rangle = \begin{cases} 1, & j = k, \\ 0, & j \neq k. \end{cases} \tag{3.22}$$

The coefficient c_k can then be derived by taking the inner product between f and the basis function φ_k:

$$\langle f, \varphi_k \rangle \approx \left\langle \sum_{j=1}^{K} c_j \varphi_j, \varphi_k \right\rangle \tag{3.23}$$

Using the linearity of the inner product gives

$$\langle f, \varphi_k \rangle \approx \sum_{j=1}^{K} c_j \langle \varphi_j, \varphi_k \rangle \tag{3.24}$$

Because the basis is orthonormal, all terms vanish except the one with $j = k$, giving

$$c_k = \langle f, \varphi_k \rangle \tag{3.25}$$

Thus, each coefficient is obtained by projecting the function onto the corresponding basis function. For real-valued functions on the sphere, the inner product used in this thesis is defined as

$$\langle f, g \rangle = \int_0^{2\pi} \int_0^{\pi} f(\theta, \phi) g(\theta, \phi) \sin\theta \, d\theta \, d\phi \tag{3.26}$$

where $\sin\theta \, d\theta \, d\phi$ is the surface area element on the sphere. This result provides the mathematical basis for representing the spherical LiDAR intensity function using spherical harmonic basis functions.

3.3.1.2 Spherical Harmonic Basis Functions

Spherical harmonics provide an orthonormal set of basis functions for representing functions defined on the sphere. They are indexed by the degree ℓ and order m, where

$$\ell = 0, 1, \ldots, L, \qquad m = -\ell, \ldots, \ell \tag{3.27}$$

For a maximum degree L, the number of spherical harmonic basis functions is

$$K = (L + 1)^2 \tag{3.28}$$

The complex spherical harmonic basis functions are commonly written as

$$Y_\ell^m(\theta, \phi) = N_{\ell m} P_\ell^m(\cos\theta) e^{im\phi} \tag{3.29}$$

where P_ℓ^m is the associated Legendre polynomial and $N_{\ell m}$ is a normalization factor. One commonly used normalization is

$$N_{\ell m} = \sqrt{\frac{2\ell + 1}{4\pi} \frac{(\ell - m)!}{(\ell + m)!}} \tag{3.30}$$

In this thesis, a real-valued spherical harmonic basis is used because the LiDAR intensity function is real-valued. The real basis functions are constructed from the complex spherical harmonics as

$$Y_{\ell m}^{\text{real}}(\theta, \phi) = \begin{cases} \sqrt{2}(-1)^m \operatorname{Im}\left(Y_\ell^{|m|}(\theta, \phi)\right), & m < 0, \\ Y_\ell^0(\theta, \phi), & m = 0, \\ \sqrt{2}(-1)^m \operatorname{Re}\left(Y_\ell^m(\theta, \phi)\right), & m > 0. \end{cases} \tag{3.31}$$

Using these basis functions, the spherical LiDAR intensity function can be approximated by a truncated spherical harmonic expansion,

$$I(\theta, \phi) \approx \sum_{\ell=0}^{L} \sum_{m=-\ell}^{\ell} c_{\ell m} Y_{\ell m}^{\text{real}}(\theta, \phi) \tag{3.32}$$

where $c_{\ell m}$ is the coefficient corresponding to degree ℓ and order m. The degree ℓ controls the spatial frequency of the basis function on the sphere. Low-degree components describe coarse, global angular intensity patterns, while higher-degree components represent finer local variation. Truncating the expansion at a finite order L therefore produces a smooth approximation of the frame-level intensity distribution, which helps reduce sensitivity to small geometric differences and local pointwise mismatches.

3.3.1.3 Discrete Coefficient Estimation from LiDAR Samples

In the continuous case, the spherical harmonic coefficient is defined by projecting the angular intensity function onto each basis function,

$$c_{\ell m} = \left\langle I, Y_{\ell m}^{\text{real}} \right\rangle = \int_0^{2\pi} \int_0^{\pi} I(\theta, \phi) Y_{\ell m}^{\text{real}}(\theta, \phi) \sin\theta \, d\theta \, d\phi. \tag{3.33}$$

This expression assumes that the intensity function $I(\theta, \phi)$ is continuously defined over the sphere. In a LiDAR frame, however, the intensity information is only available at a finite set of measured beam directions,

$$\mathcal{S} = \{(\theta_k, \phi_k, i_k)\}_{k=1}^{N}, \tag{3.34}$$

where (θ_k, ϕ_k) denotes the angular direction of the k-th point and i_k denotes its corresponding intensity value.

A direct numerical approximation of the continuous spherical integral would require appropriate surface-area quadrature weights. However, raw LiDAR samples are not uniformly distributed over the sphere. The sensor only observes a limited field of view, and its scan pattern is determined by the LiDAR beam layout rather than by uniform spherical-area sampling. Therefore, in this work the coefficients are not interpreted as unbiased estimates of the continuous spherical harmonic projection over the full sphere.

Instead, spherical harmonics are used as a structured angular basis to define a consistent empirical descriptor over the observed LiDAR directions. For each real spherical harmonic basis function, the coefficient is computed by the empirical weighted projection

$$\hat{c}_{\ell m} = \sum_{k=1}^{N} w_k i_k Y_{\ell m}^{\text{real}}(\theta_k, \phi_k). \tag{3.35}$$

In the implementation, uniform normalized sample weights are used,

$$w_k = \frac{1}{N}, \tag{3.36}$$

unless otherwise specified. This gives

$$\hat{c}_{\ell m} = \frac{1}{N} \sum_{k=1}^{N} i_k Y_{\ell m}^{\text{real}}(\theta_k, \phi_k). \tag{3.37}$$

This formulation should be understood as an empirical inner product over the LiDAR sampling distribution,

$$\langle f, g \rangle_{\mathcal{S}} = \sum_{k=1}^{N} w_k f(\theta_k, \phi_k) g(\theta_k, \phi_k), \tag{3.38}$$

rather than as an exact numerical integration over the full spherical surface. Although this empirical projection is biased with respect to the ideal continuous spherical-area integral, the bias is systematic when all frames are acquired from the same LiDAR sensor and processed using the same weighting rule. Therefore, the resulting coefficients remain comparable as frame-level descriptors.

Let the spherical harmonic basis functions be flattened into a single index j, where

$$A_{kj} = \varphi_j(\theta_k, \phi_k), \tag{3.39}$$

and φ_j denotes the corresponding real spherical harmonic basis function. The intensity samples are collected as

$$\mathbf{i} = [i_1, i_2, \ldots, i_N]^\top. \tag{3.40}$$

The empirical projection can then be written in matrix form as

$$\hat{\mathbf{c}} = A^\top W \mathbf{i}, \tag{3.41}$$

where W is a diagonal matrix containing the sample weights. With uniform normalized weights, this becomes

$$\hat{\mathbf{c}} = \frac{1}{N} A^\top \mathbf{i}. \tag{3.42}$$

The resulting coefficient vector is ordered by increasing spherical harmonic degree and order,

$$\hat{\mathbf{c}} = [c_{00}, c_{1,-1}, c_{10}, c_{11}, \ldots, c_{L,L}]^\top. \tag{3.43}$$

This vector provides a compact empirical representation of the angular intensity distribution observed by the LiDAR frame. It is not intended to reconstruct the full continuous intensity function exactly, but rather to encode the observed frame in a consistent spherical harmonic feature space. The coefficient vector is then used to construct the rotationally invariant energy descriptor described in the following subsection.

3.3.2 Rotationally Invariant Energy Descriptor

Before constructing the rotationally invariant descriptor, it is useful to consider the direct distance between two spherical harmonic coefficient vectors. Let $f(\theta, \phi)$ and $g(\theta, \phi)$ be two angular intensity functions represented using the same real spherical harmonic basis:

$$f(\theta, \phi) = \sum_{\ell=0}^{L} \sum_{m=-\ell}^{\ell} c^f_{\ell m} Y^{\text{real}}_{\ell m}(\theta, \phi), \qquad g(\theta, \phi) = \sum_{\ell=0}^{L} \sum_{m=-\ell}^{\ell} c^g_{\ell m} Y^{\text{real}}_{\ell m}(\theta, \phi). \tag{3.44}$$

Their difference can be written as

$$f(\theta, \phi) - g(\theta, \phi) = \sum_{\ell=0}^{L} \sum_{m=-\ell}^{\ell} \left(c_{\ell m}^{f} - c_{\ell m}^{g}\right) Y_{\ell m}^{\mathrm{real}}(\theta, \phi) \tag{3.45}$$

The squared L^2 distance between the two functions on the sphere is

$$\|f - g\|_{L^2(\mathbb{S}^2)}^2 = \int_0^{2\pi} \int_0^{\pi} \left(f(\theta, \phi) - g(\theta, \phi)\right)^2 \sin\theta \, d\theta \, d\phi \tag{3.46}$$

Substituting the spherical harmonic expansions gives

$$\|f - g\|_{L^2(\mathbb{S}^2)}^2 = \int_0^{2\pi} \int_0^{\pi} \left(\sum_{\ell=0}^{L} \sum_{m=-\ell}^{\ell} \left(c_{\ell m}^{f} - c_{\ell m}^{g}\right) Y_{\ell m}^{\mathrm{real}}(\theta, \phi) \right)^2 \sin\theta \, d\theta \, d\phi \tag{3.47}$$

Because the real spherical harmonic basis functions are orthonormal under the spherical inner product, all cross terms vanish:

$$\left\langle Y_{\ell m}^{\mathrm{real}}, Y_{\ell' m'}^{\mathrm{real}} \right\rangle = \begin{cases} 1, & \ell = \ell' \text{ and } m = m', \\ 0, & \text{otherwise.} \end{cases} \tag{3.48}$$

Therefore, the squared function-space distance is equal to the squared Euclidean distance between the corresponding coefficient vectors:

$$\|f - g\|_{L^2(\mathbb{S}^2)}^2 = \sum_{\ell=0}^{L} \sum_{m=-\ell}^{\ell} \left(c_{\ell m}^{f} - c_{\ell m}^{g}\right)^2 \tag{3.49}$$

This relationship shows that, when the spherical harmonic basis is orthonormal, comparing the coefficient vectors directly is mathematically equivalent to comparing the two angular intensity functions using the squared L^2 distance on the sphere. If the real and simulated LiDAR frames were perfectly aligned in orientation, this direct coefficient-space distance could therefore be used as a natural function-space comparison metric.

However, direct comparison of individual coefficients is sensitive to orientation differences. A rotation of the input frame changes the individual coefficients $c_{\ell m}$, even if the underlying angular intensity pattern remains similar. In practical sim-to-real comparison, small orientation or pose differences between the real and simulated frames are difficult to avoid. Therefore, a direct coefficient-wise L^2 distance may report a large difference even when the two frames contain similar angular intensity structure.

To improve robustness to orientation mismatch, this thesis uses a degree-wise spherical harmonic energy descriptor. For a fixed degree ℓ, a rotation mixes the coefficients among the orders $m = -\ell, \ldots, \ell$, but preserves the norm of the coefficient band. Therefore, each LiDAR frame is represented by the energy at each spherical harmonic degree:

$$E_\ell = \left(\sum_{m=-\ell}^{\ell} c_{\ell m}^2 \right)^{1/2} \tag{3.50}$$

The full descriptor is then

$$\mathbf{E} = [E_0, E_1, \ldots, E_L]^\top \tag{3.51}$$

This descriptor is rotationally invariant because rotations redistribute coefficient values within the same degree ℓ, but do not change the total energy of that degree. In addition, the descriptor has a multi-scale interpretation. Low degrees describe coarse, global angular intensity structure, while higher degrees describe finer local variations.

In this thesis, the comparison gives higher importance to the lower degrees because the overall intensity distribution is more relevant than sharp local details. This also improves tolerance to geometric mismatch between the real scene and the digital twin. The weighted energy distance between a real frame and a simulated frame is defined as

$$D_\mathrm{w} = \left(\sum_{\ell=0}^{L} w_\ell \left(E_\ell^\mathrm{real} - E_\ell^\mathrm{sim} \right)^2 \right)^{1/2} \tag{3.52}$$

where w_ℓ is a non-negative weight assigned to degree ℓ, normalized as

$$\sum_{\ell=0}^{L} w_\ell = 1 \tag{3.53}$$

In this work, decreasing degree weights are used to emphasize low-order spherical harmonic components and reduce the influence of high-frequency details. Three possible weighting schemes are considered:

$$\begin{aligned} w_\ell^\mathrm{lin} &= \frac{L - \ell + 1}{\sum_{j=0}^{L} (L - j + 1)}, \\ w_\ell^\mathrm{inv} &= \frac{1/(\ell + 1)}{\sum_{j=0}^{L} 1/(j + 1)}, \\ w_\ell^\mathrm{exp} &= \frac{\exp(-\ell)}{\sum_{j=0}^{L} \exp(-j)}. \end{aligned}$$

These choices all assign larger weights to lower spherical harmonic degrees. Therefore, the final distance compares the frame-level intensity structure in a way that is rotation-invariant while remaining tolerant to local geometric differences.

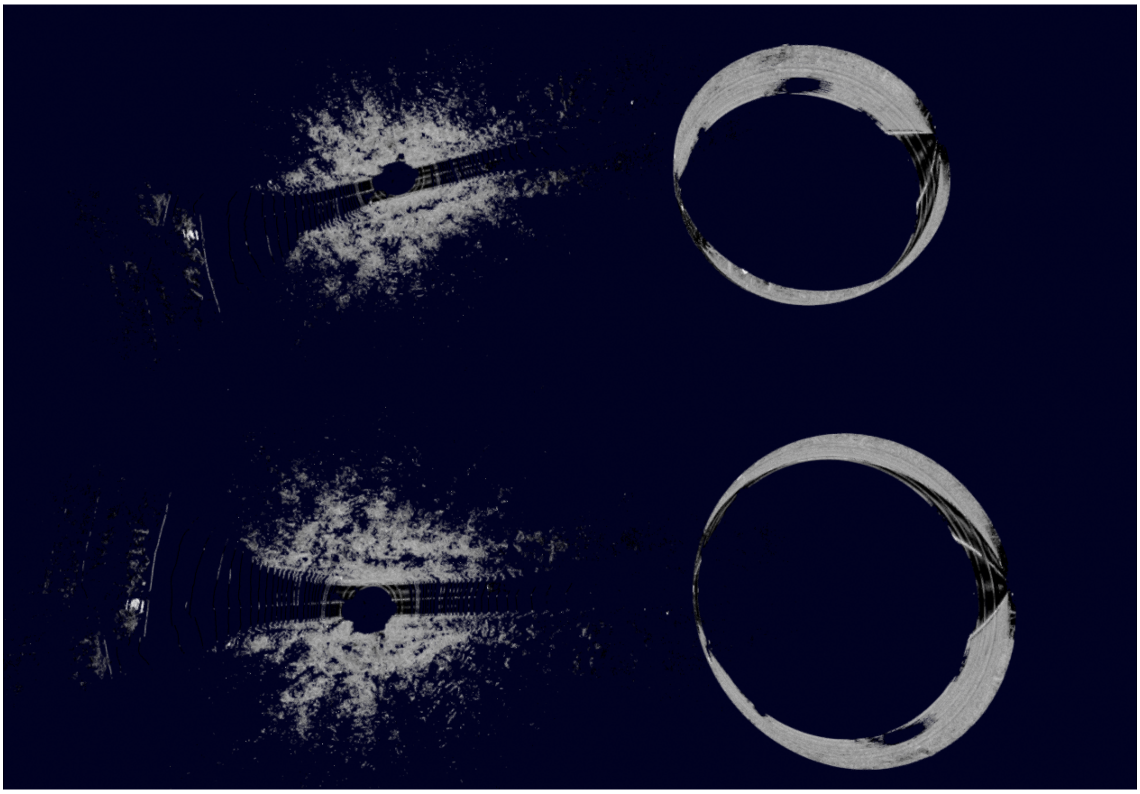


Figure 3.7: Rotational invariance of the spherical harmonic energy descriptor. The two LiDAR frames are rotated relative to each other, but their spherical harmonic energy descriptors remain the same or very similar because the descriptor depends on degree-wise energy rather than the absolute orientation of the frame.

Algorithm 2 summarizes the complete frame-level comparison procedure. Given one real LiDAR frame and one simulated LiDAR frame, the method converts both frames into spherical intensity samples, estimates their spherical harmonic coefficients, computes the degree-wise energy descriptors, and finally evaluates the weighted rotationally invariant distance.

Algorithm 2 Spherical harmonic energy-based frame comparison

```
Require: Real LiDAR frame $\mathcal{P}^{\text{real}}$, simulated LiDAR frame $\mathcal{P}^{\text{sim}}$, maximum SH
    order $L$, degree weights $\{w_\ell\}_{\ell=0}^{L}$
Ensure: Weighted rotationally invariant distance $D_{\text{w}}$
    Procedure ComputeEnergyDescriptor($\mathcal{P}, L$)
Remove invalid points from $\mathcal{P}$
Convert each point $(x_k, y_k, z_k, i_k)$ to angular sample $(\theta_k, \phi_k, i_k)$
Build the real spherical harmonic basis matrix $A$ up to order $L$
Estimate coefficient vector $\hat{\mathbf{c}}$ using empirical projection over the observed LiDAR
    samples
        $$\hat{\mathbf{c}} = A^\top W \mathbf{i}$$
for $\ell = 0$ to $L$ do
    Extract coefficient band $\mathbf{c}_\ell = [c_{\ell,-\ell}, \dots, c_{\ell,\ell}]^\top$
    Compute degree-wise energy
        $$E_\ell = \left( \sum_{m=-\ell}^{\ell} c_{\ell m}^2 \right)^{1/2}$$
end for
Store $\mathbf{E} = [E_0, E_1, \dots, E_L]^\top$

$\mathbf{E}^{\text{real}} \leftarrow$ ComputeEnergyDescriptor($\mathcal{P}^{\text{real}}, L$)
$\mathbf{E}^{\text{sim}} \leftarrow$ ComputeEnergyDescriptor($\mathcal{P}^{\text{sim}}, L$)
Normalize the degree weights so that $\sum_{\ell=0}^{L} w_\ell = 1$
Compute the weighted energy distance
        $$D_{\text{w}} = \left( \sum_{\ell=0}^{L} w_\ell \left( E_\ell^{\text{real}} - E_\ell^{\text{sim}} \right)^2 \right)^{1/2}$$
return $D_{\text{w}}$
```

4

Results

This chapter presents the results of the proposed LiDAR intensity simulation and evaluation pipeline. Reconstructed real-world LiDAR frames from MCAP recordings are compared with corresponding simulated LiDAR frames generated in the CARLA-based AstaZero digital twin. The comparison focuses on the calibrated reflectivity behavior represented by the vendor-specific intensity values in the reference data.

The results first compare the intensity distributions of real-world and simulated points before vendor-specific fitting. An empirical mapping is then fitted to transform the physically motivated simulated intensity into a vendor-specific calibrated reflectivity representation. After applying this mapping, the simulated intensity distribution is compared again with the real-world reference data. The comparison is quantified using traditional distribution-based metrics and the proposed spherical harmonic energy distance.

This chapter therefore evaluates both parts of the proposed approach: the physically motivated intensity model with vendor-specific mapping, and the geometry-tolerant frame-level evaluation method.

4.1 Intensity Distributions Alignment and Fitting Effects

The first evaluation analyzes the intensity profile distributions across three models: the uncalibrated model, and the models using the master Full-FOV equation ($\bar{F}^{\text{full}}_{\text{Ouster}}$), and the master Reduced-FOV equation ($\bar{F}^{\text{red}}_{\text{Ouster}}$). To establish a clear baseline and isolate the effects of the empirical mapping, a single representative LiDAR frame is selected from Scenario A (dry environmental conditions) and another from Scenario B (wet/rainy conditions), matched precisely with their corresponding digital-twin frames in the simulation. These specific frames are chosen such that there are no vehicles, buildings, or tree vegetation in the surrounding area in order to effectively isolate the raw sensor reality gap during the analysis, avoiding external geometric or multi-material variables.

For each selected frame, the intensity distributions are extracted and visualized as histograms in a single comparative matrix, presented in Figure 4.1. The horizontal axis denotes the intensity or calibrated reflectivity values, while the vertical axis indicates the absolute point counts per bin. The columns separate the data by environmental weather state (dry, wet), while the rows demonstrate the step-by-step evolution of the simulation-to-reality alignment.

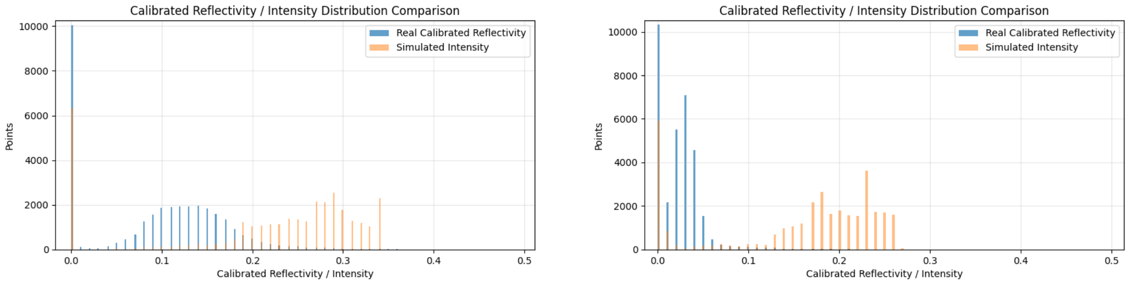
The first row of the matrix (Figures 4.1a and 4.1b) illustrates the uncalibrated single-frame intensity distributions. In both environmental states, an important alignment mismatch is visible between the real-world reference data (R_{RFL8}) and the raw simulation outputs (I_{phys}). The real-world reference data and the simulated data have different distribution shapes, which is expected because the simulated intensity at this stage is still a relative physically motivated return value, it has not yet been transformed into the vendor-specific calibrated reflectivity representation used in the reference data. Furthermore, comparing the left and right columns of this initial row demonstrates that changing the lookup tables from dry to wet materials shifts the raw simulation distributions, yet fails to bridge the reality gap to the true sensor data. This baseline mismatch directly highlights why an analytical translation is mathematically unfeasible and validates the need of the empirical master fitting curves.

To bridge this gap, the empirical quantile regression routine described in the methodology was executed across the four fitting configurations. Figure 4.2 shows the explicit distribution-mapping results for Scenario A (Dry), stacking the Full-FOV configuration (Figure 4.2a) and the Reduced-FOV configuration (Figure 4.2b). Similarly, Figure 4.3 presents the corresponding optimization curves for Scenario B (Wet) for the Full-FOV (Figure 4.3a) and Reduced-FOV (Figure 4.3b) configurations. In all the configurations, the cubic polynomial family came up to be the best to track the empirical quantile matching pairs (x_q, y_q) .

To achieve a generalized transformation function, these local curves were combined by averaging the parameters. Figure 4.4 presents all four scenario-specific curves alongside the two resulting master equations. This joint visualization makes the parameter averaging process easily understandable, the master fitting equations act as stable geometric midpoints, successfully balancing the clear, high-return dry reflectance curves with the damp, attenuated characteristics induced by wet surface conditions across both spatial configurations.

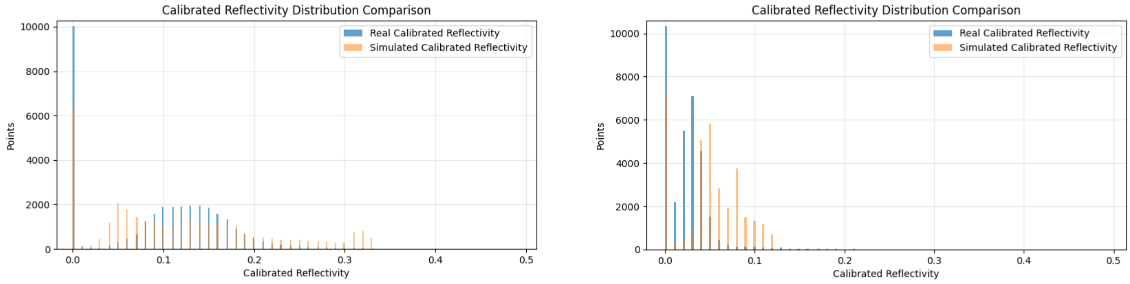
The resulting numerical coefficients (a, b, c, d) extracted from these optimizations are summarized in Table 4.1 with the respective Mean Squared Error (MSE) and Root Mean Squared Error (RMSE). The variation among the parameters highlights how environmental water films and spatial field-of-view filtering affect the local curve fitting, emphasizing that a single scenario run isn't sufficient to define a generalized/master model.

The second row of the matrix (Figures 4.1c and 4.1d) illustrates the distribution profiles after applying the master Full-FOV calibration curve. The mapping successfully shifts the raw simulated intensity domain into the targeted vendor-specific calibrated reflectivity range. However, significant differences in the distributions still remain.



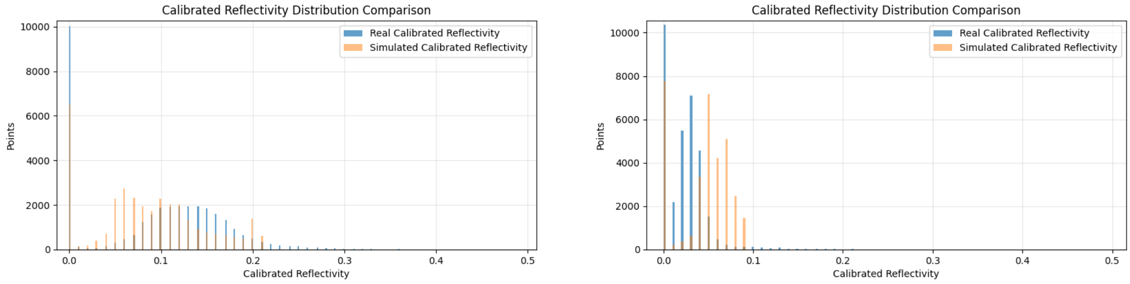
(a) Scenario A (Dry): Uncalibrated Model

(b) Scenario B (Wet): Uncalibrated Model



(c) Scenario A (Dry): Master Full-FOV Model

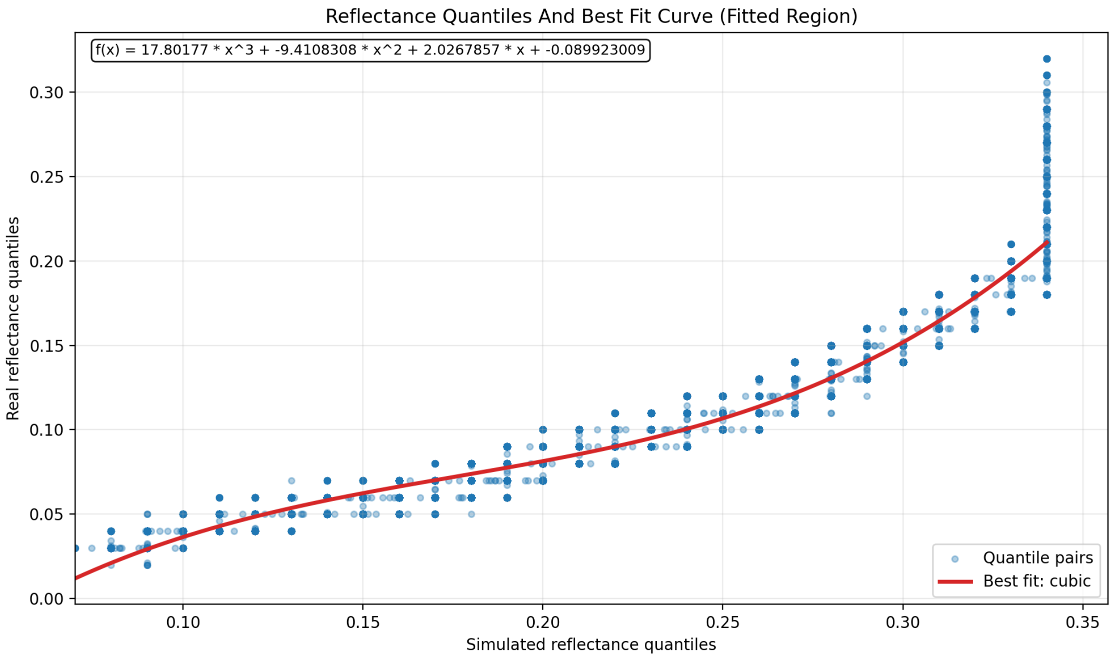
(d) Scenario B (Wet): Master Full-FOV Model



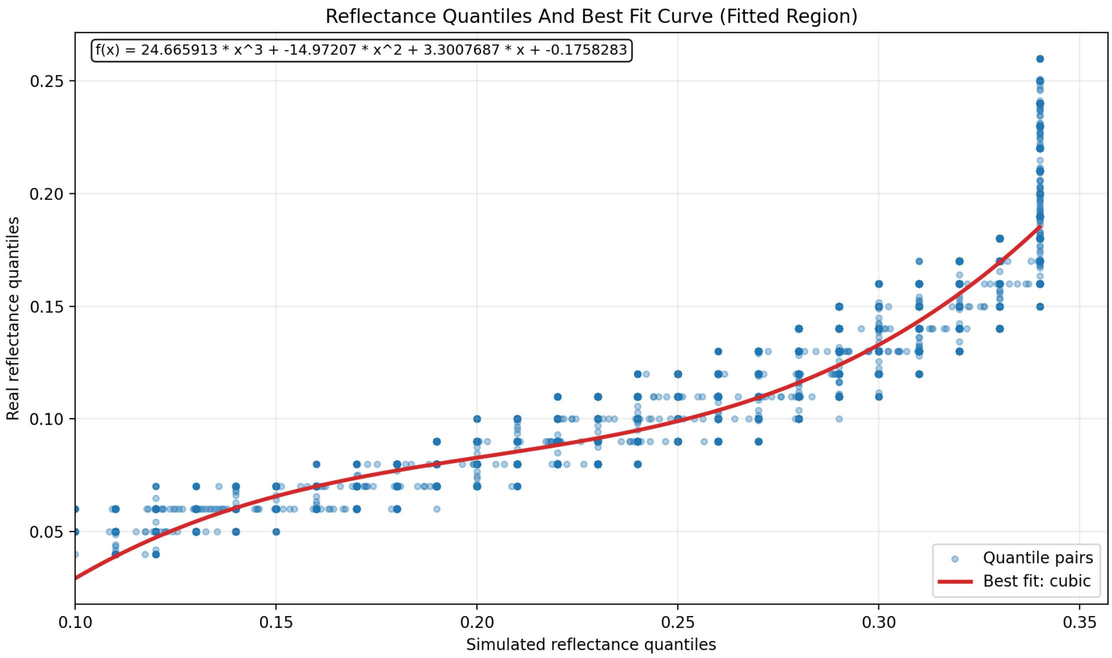
(e) Scenario A (Dry): Master Reduced-FOV Model

(f) Scenario B (Wet): Master Reduced-FOV Model

Figure 4.1: Single-frame intensity and calibrated reflectivity distribution matrices for Scenario A (left column) and Scenario B (right column). The top row (a, b) illustrates the raw, uncalibrated gap between simulated physical intensity (I_{phys}) and real-world reference data (R_{RFL8}). The middle row (c, d) shows the resulting distributions after applying empirically fitted master Full-FOV function ($\bar{F}^{\text{full}}_{\text{Ouster}}$). The bottom row (e, f) shows the alignment achieved using the master Reduced-FOV function ($\bar{F}^{\text{red}}_{\text{Ouster}}$).

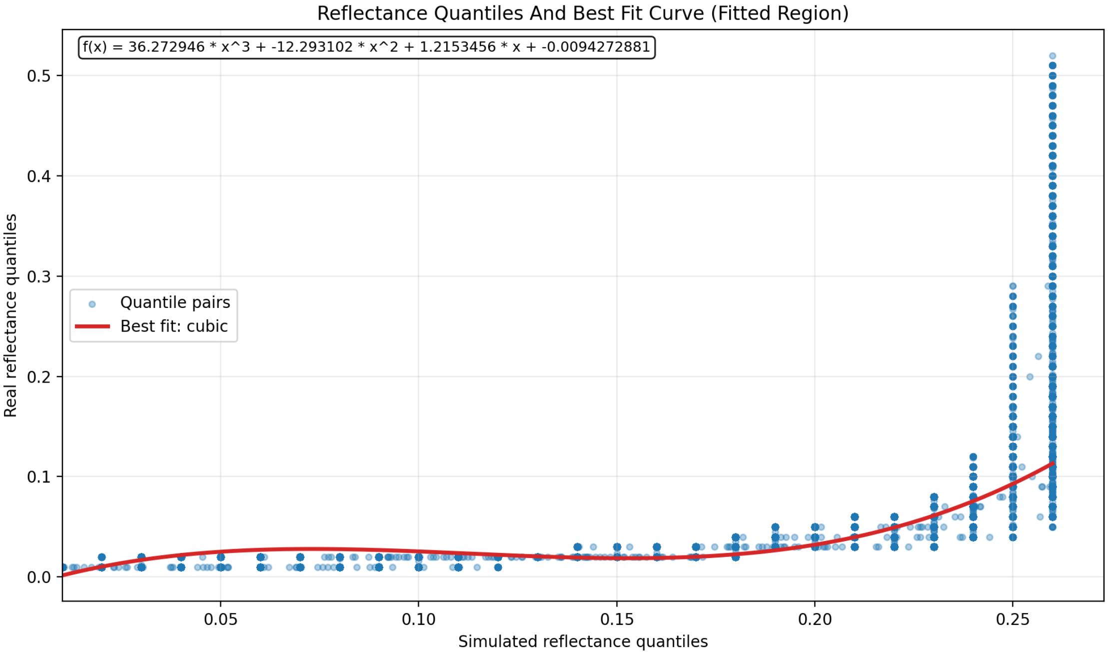


(a) Full FOV Configuration

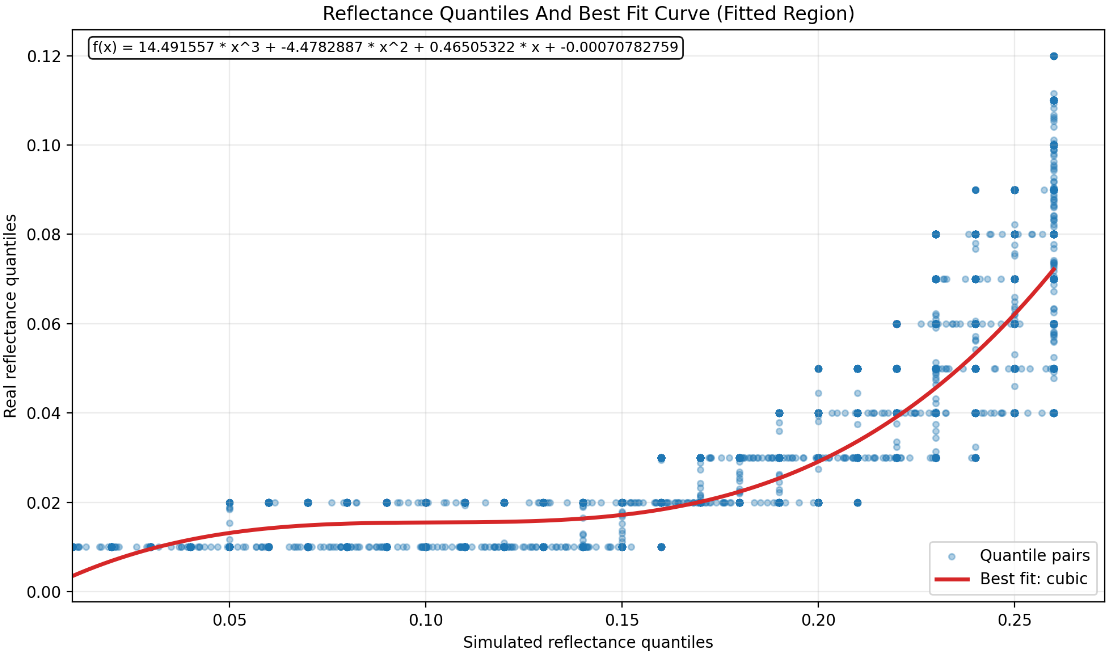


(b) Reduced FOV Configuration

Figure 4.2: Empirical quantile matching pairs with corresponding optimized cubic regression curves for Scenario A (Dry Conditions).



(a) Full FOV Configuration



(b) Reduced FOV Configuration

Figure 4.3: Empirical quantile matching pairs with corresponding optimized cubic regression curves for Scenario B (Wet Conditions).

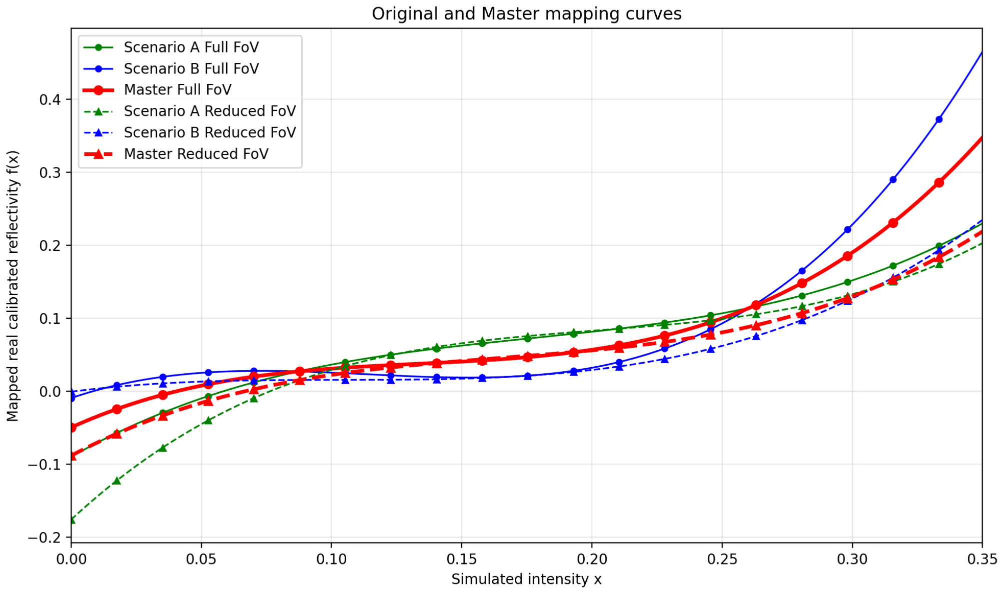


Figure 4.4: Comprehensive comparison of all local scenario-specific calibration curves and their corresponding averaged master functions ($\bar{F}^{\text{full}}_{\text{Ouster}}$ and $\bar{F}^{\text{red}}_{\text{Ouster}}$).

Table 4.1: Optimal cubic regression parameters and fitting errors for the calibration configurations

Configuration	a	b	c	d	**MSE**	**RMSE**
A — Full FOV	17.8017	−9.4108	2.0267	−0.0899	0.000114	0.01067
B — Full FOV	36.2729	−12.2931	1.2153	−0.0094	0.000651	0.02552
Master ($\bar{F}^{\text{full}}$)	27.0373	−10.8519	1.6210	−0.0496	—	—
A — Reduced FOV	24.6659	−14.9720	3.3007	−0.1758	0.000150	0.01227
B — Reduced FOV	14.4915	−4.4782	0.4650	−0.0007	0.000106	0.01031
Master ($\bar{F}^{\text{red}}$)	19.5787	−9.7251	1.8829	−0.0882	—	—

In the dry configuration (Figure 4.1c), while the real reference data effectively terminates near 0.25, the simulated calibrated profile shows a higher calibrated reflectivity tail that extends up to 0.33. Similarly, in the wet configuration (Figure 4.1d), the real sensor returns are mainly compressed below 0.05, but the Full-FOV model distribution is still shifted to the right of the real distribution, having still a relevant amount of returns in the range of 0.04 to 0.12. These results show the impact of utilizing a wide 180° spatial window for calibration. Because the Full-FOV calibration dataset captures uncorrected range-dependent attenuation from peripheral targets, it introduces severe geometric and material noise into the initial quantile regression, leading to still undesired results.

On the other hand, the third row of the matrix (Figures 4.1e and 4.1f) captures the distributions following the application of the master Reduced-FOV calibration profile. By restricting the regression strictly to the centralized 70° × 60-channel grid, the calibration isolates the roughly uniform gravel and soil surfaces of the ground plane, successfully replicating controlled real-world laboratory measurement conditions.

The resulting distribution improvements are noticeable. In the dry configuration (Figure 4.1e), the erroneous high-reflectivity tail is completely eliminated, and the simulated curve mainly stops close to the real 0.21 limit. Although this significantly increases the overlapping bin density between the real and simulated calibrated reflectivity distributions, this curve-fitting underestimates the intensity of the overall distribution, shifting the simulated calibrated reflectivity distribution to the left of the real distribution. In the wet state (Figure 4.1f), the distribution is further compressed towards the origin, strictly bounding the simulated outputs below 0.09. Although the distribution is shifted more towards the origin, it is noticeable that the calibration does not manage to fully approximate the distribution to the lower reflectivity values shown by the real calibrated reflectivity distribution.

These visual results show that isolating the optimization layer to a focused, ground-focused field of view and calculating the arithmetic mean of the cubic polynomial parameters successfully filters out the peripheral noise and range-dependent attenuation, showing a superior simulation-to-reality alignment compared to both the uncalibrated baseline and the noisy Full-FOV model. However, the residual discrepancies, specifically the slight underestimation of the main intensity distribution in the dry state and the incomplete compression toward the lowest reflectivity values in the wet configuration directly expose the inherent fidelity limitations of the digital-twin environment.

While the simulation environment maps the drivable area as a relatively uniform, homogeneous material surface, the actual physical test track presents a heterogeneous, complex mixture of gravel, soil, and distinct road elements that produce highly localized textures and varied reflectivity distributions. This can be observed in Figure 4.5.

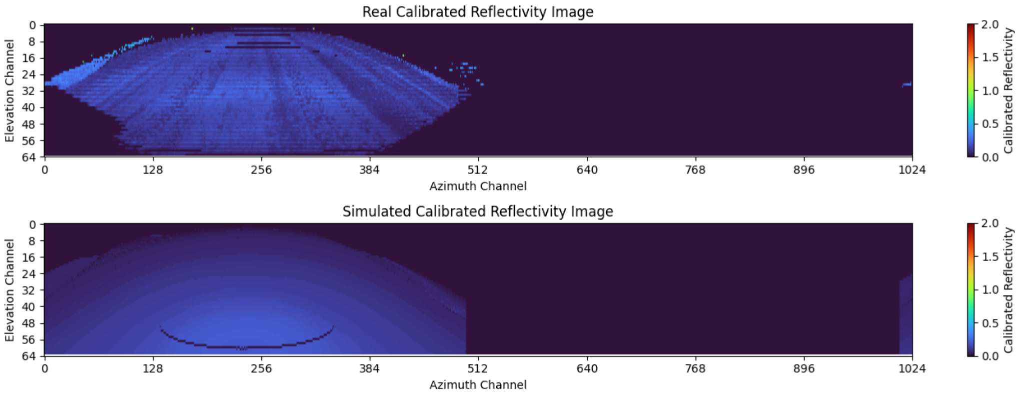


Figure 4.5: Comparison of real-world and simulated LiDAR scan intensity frames under dry conditions (Scenario A). The real-world reference data captures the material heterogeneity and structural variations between gravel and soil on the track. On the other hand, the digital-twin simulation renders the drivable area as a completely homogeneous material surface.

Furthermore, during rainy scenarios as the one shown in Figure 4.6, the real-world environment naturally forms scattered puddles, whereas the digital twin applies a uniform moisture lookup distribution across the entire ground layer. This real-world surface variability generates a complex, scattered reflectivity behavior that a single one-dimensional polynomial transfer function cannot fully map. Nevertheless, the Reduced-FOV master calibration establishes a robust and stable baseline for sensor rendering, providing a physical foundation that is further evaluated using the spatial metrics detailed in the subsequent sections.

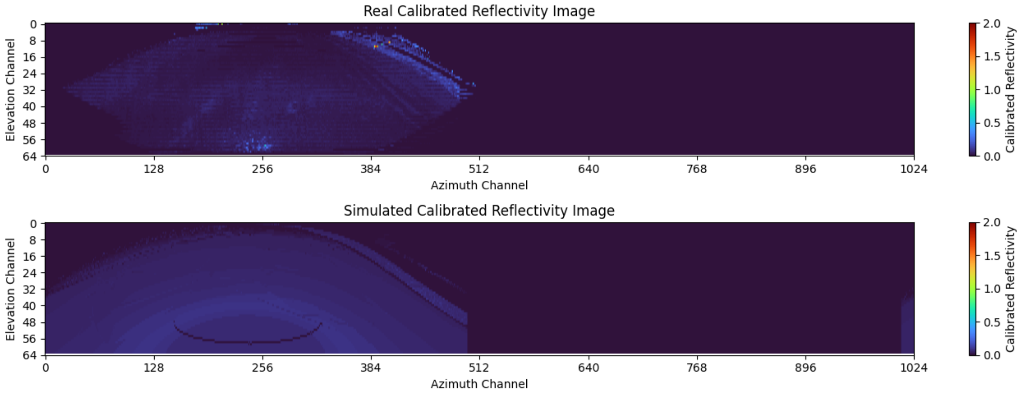


Figure 4.6: Comparison of real-world and simulated LiDAR scan intensity frames under wet conditions (Scenario B). The real-world reference data shows highly localized low-reflectivity regions caused by water puddles that scatter the laser beams away from the receiver. In contrast, the simulation treats the entire surface with a uniform, unvaried wet material profile, failing to replicate the irregular spatial distribution of surface water.

4.2 Traditional Distribution-Based Frame Comparison

To establish a classical benchmark and evaluate how our calibration layers impact intensity alignment, we use two standard distribution-based metrics: the Wasserstein distance and the Jensen–Shannon distance, using their respective implementations in the SciPy library [20, 21]. These metrics treat a frame's LiDAR intensity values as a numeric distribution and evaluate how closely the simulation matches real-world logs.

- **Wasserstein Distance (WD):** Often called the Earth Mover's Distance, WD measures the total statistical work required to shift and reshape one distribution into another. It is geometry-aware, which means it tracks global shape alignment and horizontal shifts between curves.
- **Jensen–Shannon Distance (JSD):** This metric measures strict, bin-by-bin vertical overlap based on information entropy. It is highly sensitive to alignment, meaning even a tiny horizontal offset between two identical, sharp spikes results in a high error penalty, regardless of overall shape similarity.

A comprehensive statistical summary of the temporal mean, variance (Var.), standard deviation (SD, σ), minimum (Min), and maximum (Max) values across all four testing scenarios is shown in Table 4.2 for the WD metric and Table 4.3 for the JSD metric.

Table 4.2: Statistics of frame-level Wasserstein distance (WD) across all scenarios and calibration configurations. Lower values indicate closer frequency distribution agreement with the real-world sensor logs.

Scenario	**Model Layer**	**Mean**	**Var.** ($\times 10^{-6}$)	**SD** (σ)	**Min**	**Max**
A (Dry)	BL	0.7022	72.2124	0.0085	0.6907	0.7318
	PM-BF	0.1228	14.6298	0.0038	0.1129	0.1305
	$F_{A,\text{full}}$	0.0166	7.0865	0.0027	0.0128	0.0213
	$F_{A,\text{red}}$	0.0174	3.1169	0.0018	0.0150	0.0228
	$\bar{F}^{\text{full}}_{\text{Ouster}}$	0.0299	6.7679	0.0026	0.0242	0.0353
	$\bar{F}^{\text{red}}_{\text{Ouster}}$	0.0160	8.9661	0.0030	0.0111	0.0240
B (Wet)	BL	0.7981	606.9812	0.0246	0.7486	0.8400
	PM-BF	0.1257	36.0991	0.0060	0.1123	0.1329
	$F_{B,\text{full}}$	0.0116	12.4815	0.0035	0.0063	0.0215
	$F_{B,\text{red}}$	0.0109	29.1016	0.0054	0.0065	0.0287
	$\bar{F}^{\text{full}}_{\text{Ouster}}$	0.0240	11.6186	0.0034	0.0165	0.0291
	$\bar{F}^{\text{red}}_{\text{Ouster}}$	0.0214	13.0256	0.0036	0.0138	0.0293
C (Dry)	$\bar{F}^{\text{full}}_{\text{Ouster}}$	0.0360	18.3784	0.0043	0.0244	0.0493
	$\bar{F}^{\text{red}}_{\text{Ouster}}$	0.0112	6.5465	0.0026	0.0077	0.0203
D (Wet)	$\bar{F}^{\text{full}}_{\text{Ouster}}$	0.0201	0.1470	0.0004	0.0192	0.0219
	$\bar{F}^{\text{red}}_{\text{Ouster}}$	0.0168	0.2057	0.0005	0.0157	0.0179

BL: baseline model; PM-BF: proposed model before fitting;
$F_{A,\text{full}}$ / $F_{B,\text{full}}$: localized Full Field-of-View (FOV) fitting curves;
$F_{A,\text{red}}$ / $F_{B,\text{red}}$: localized Reduced-FOV fitting curves;
$\bar{F}^{\text{full}}_{\text{Ouster}}$: master Full-FOV fitting curve;
$\bar{F}^{\text{red}}_{\text{Ouster}}$: master Reduced-FOV fitting curve.

Table 4.3: Statistics of frame-level Jensen–Shannon distance (JSD) across all scenarios and calibration configurations. Lower values indicate closer frequency distribution agreement with the real-world sensor logs.

Scenario	**Model Layer**	**Mean**	**Var.** ($\times 10^{-3}$)	**SD** (σ)	**Min**	**Max**
A (Dry)	BL	0.8738	0.0284	0.0053	0.8661	0.8881
	PM-BF	0.7595	0.5475	0.0234	0.6709	0.8133
	$F_{A,\text{full}}$	0.5072	6.5142	0.0807	0.2634	0.7034
	$F_{A,\text{red}}$	0.5195	5.7295	0.0757	0.2774	0.6984
	$\bar{F}^{\text{full}}_{\text{Ouster}}$	0.5540	4.3601	0.0660	0.3422	0.7191
	$\bar{F}^{\text{red}}_{\text{Ouster}}$	0.5424	5.5854	0.0747	0.2852	0.7278
B (Wet)	BL	0.8952	0.2096	0.0145	0.8718	0.9230
	PM-BF	0.8044	0.7284	0.0270	0.7405	0.8693
	$F_{B,\text{full}}$	0.5142	5.5400	0.0744	0.3268	0.7018
	$F_{B,\text{red}}$	0.5285	12.3498	0.1111	0.3201	0.7412
	$\bar{F}^{\text{full}}_{\text{Ouster}}$	0.6093	2.0476	0.0452	0.4596	0.7364
	$\bar{F}^{\text{red}}_{\text{Ouster}}$	0.6187	1.8500	0.0430	0.4835	0.7382
C (Dry)	$\bar{F}^{\text{full}}_{\text{Ouster}}$	0.4414	34.1157	0.1847	0.0120	0.7265
	$\bar{F}^{\text{red}}_{\text{Ouster}}$	0.3837	47.5624	0.2181	0.0204	0.7111
D (Wet)	$\bar{F}^{\text{full}}_{\text{Ouster}}$	0.5369	2.8761	0.0536	0.3570	0.7341
	$\bar{F}^{\text{red}}_{\text{Ouster}}$	0.5705	1.3179	0.0363	0.4375	0.7260

BL: baseline model; PM-BF: proposed model before fitting;
$F_{A,\text{full}}$ / $F_{B,\text{full}}$: localized Full Field-of-View (FOV) fitting curves;
$F_{A,\text{red}}$ / $F_{B,\text{red}}$: localized Reduced-FOV fitting curves;
$\bar{F}^{\text{full}}_{\text{Ouster}}$: master Full-FOV fitting curve;
$\bar{F}^{\text{red}}_{\text{Ouster}}$: master Reduced-FOV fitting curve.

4.2.1 Evaluation in Generalization Scenarios (Scenarios A and B)

The continuous frame-by-frame metric evaluation for Scenario A (Dry) and Scenario B (Wet) are shown in Figures 4.7 and 4.8. To fully focus on the performance of the fitted mapping, the high-error uncalibrated models are omitted from these figures.

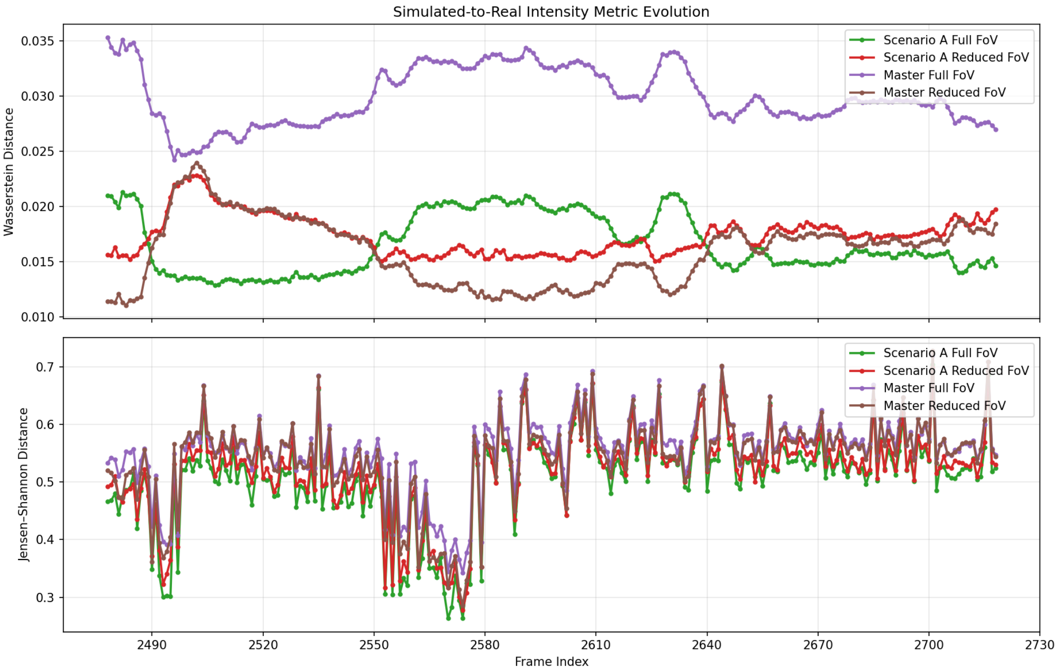


Figure 4.7: Continuous frame-level temporal evolution of the Wasserstein distance (top) and Jensen–Shannon distance (bottom) across the Scenario A (Dry) trajectory, focusing on the scenario specific and master configurations of the proposed calibration model.

As reflected in Table 4.2, the baseline simulated model (BL) performs poorly in the Wasserstein distance, obtaining high mean errors of 0.7022 (Scenario A) and 0.7981 (Scenario B) because it ignores key physical factors like incidence angles and material-specific reflectivity behavior. Introducing our uncalibrated physics model (PM-BF) decreases the Wasserstein mean to 0.1228, proving that the range equations and material-dependent reflectivity successfully introduce these characteristics into the model.

Using the empirically mapped fitting curves allows to reduce the order-of-magnitude of the error, collapsing the Wasserstein mean to 0.0166 for $F_{A,\text{full}}$ and 0.0116 for $F_{B,\text{full}}$. The continuous timelines show better results for the Reduced-FOV configurations (F_{red} and $\bar{F}^{\text{red}}$), although in the case of Scenario B, the scenario-specific fitting curves clearly perform better than the master functions. When comparing the generalized master curves ($\bar{F}_{\text{Ouster}}$), it is observed that the Reduced-FOV master curve $\bar{F}^{\text{red}}_{\text{Ouster}}$ consistently performs better than the Full-FOV master curve $\bar{F}^{\text{full}}_{\text{Ouster}}$ in both scenarios.

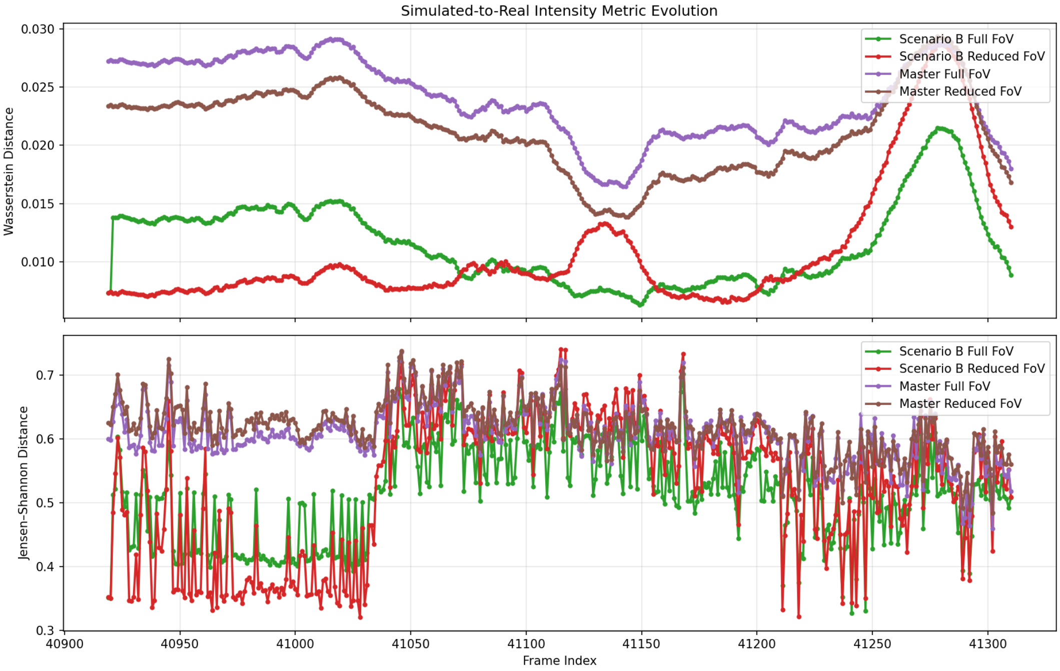


Figure 4.8: Continuous frame-level temporal evolution of the Wasserstein distance (top) and Jensen–Shannon distance (bottom) across the Scenario B (Wet) trajectory, focusing on the scenario specific and master configurations of the proposed calibration model.

4.2.2 Evaluation in Validation Scenarios (Scenarios C and D)

To prove the generalization capacity of the fitting curves and models, Figures 4.9 and 4.10 show the continuous evaluation of the traditional metrics in the validation scenarios using only the generalized master curves.

The statistical analysis shown in Table 4.2 confirms the stability of the master curves in unseen scenarios. In Scenario C, the master Reduced-FOV calibration establishes the lowest error with a mean WD of 0.0112 (compared to 0.0359 for the Full-FOV fitting curve). In the wet track (Scenario D), the error presents a mean WD of 0.0201 (Full-FOV) and 0.0168 (Reduced-FOV). The roughly constant and flat lines along the whole recording demonstrate that the calibration parameters are correctly approximating the physical hardware behavior rather than overfitting to specific environmental layouts.

4.2.3 Jensen–Shannon Distance Analysis

In sharp contrast to the descriptive and clear differences shown in the Wasserstein profiles, the JSD results (lower panels across Figures 4.7 to 4.10) collapse into a tight, highly synchronized region, following nearly identical paths regardless of the fitting curve used.

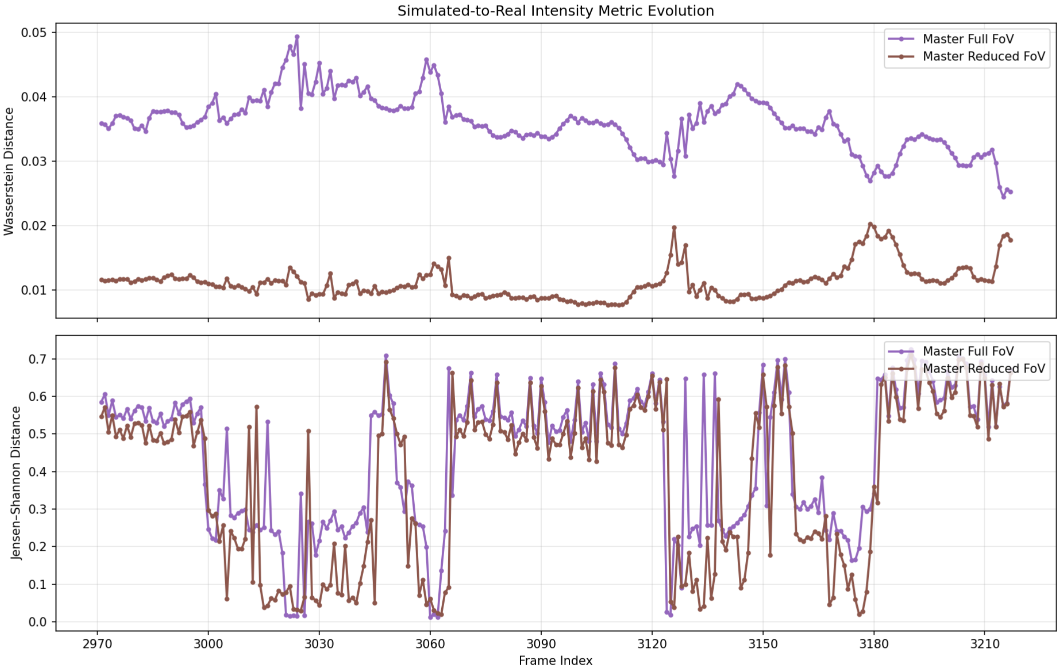


Figure 4.9: Continuous frame-level temporal evolution of the Wasserstein distance (top) and Jensen–Shannon distance (bottom) across the validation Scenario C (Dry) using master fitting curves.

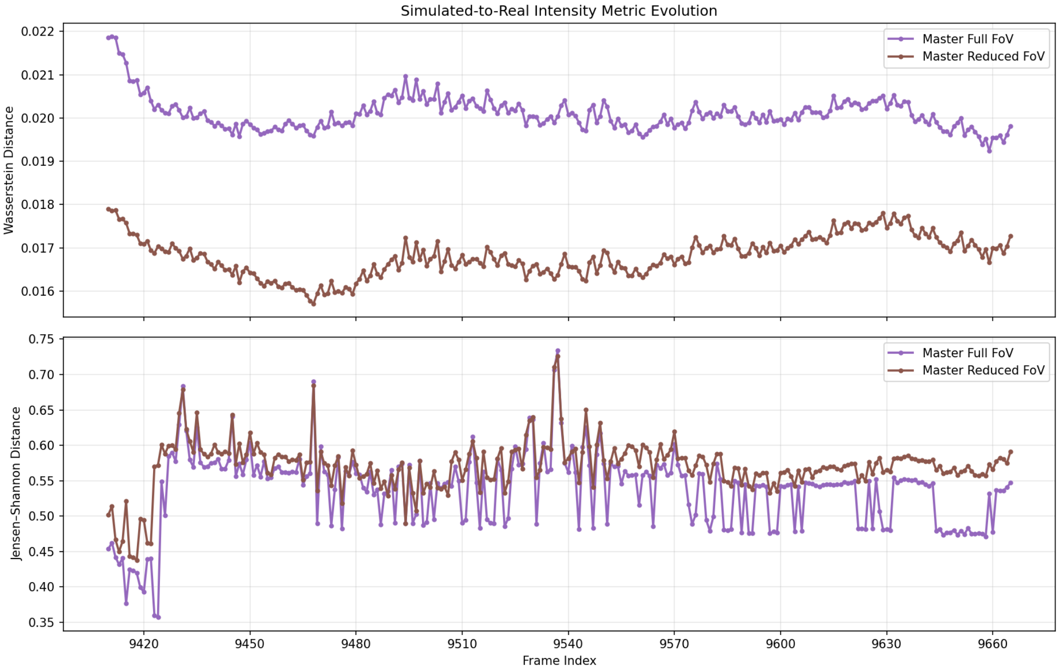


Figure 4.10: Continuous frame-level temporal evolution of the Wasserstein distance (top) and Jensen–Shannon distance (bottom) across the validation Scenario D (Wet) using master fitting curves.

This behavior occurs because raw LiDAR intensity histograms present sharp, discrete spikes and valleys. Because JSD relies on a strict bin-to-bin comparison, any small horizontal shift or bin offset between simulation and reality triggers a penalty, masking the broad alignment improvements achieved by calibration.

4.3 Spherical Harmonic Energy Distance Evaluation

The final evaluation utilizes the proposed spherical harmonic (SH) energy descriptor to compare the angular intensity structures of real and simulated LiDAR point clouds. While traditional distribution-based metrics analyze the frequency of reflective values, they entirely discard spatial topology by treating points independently of their physical location. In contrast, the SH energy method preserves the directional distribution of intensity across the sensor's field of view. Hence, it evaluates not only which intensity values are produced in the simulation, but also whether they are generated in the correct spatial geometries within the scene.

Figure 4.11 illustrates the complete visual pipeline of this evaluation framework. For any given frame, the raw 3D point clouds from both the real log and the digital twin are converted into spherical coordinate intensity samples. The SH coefficients are then estimated for each individual frame, from which the coordinate-invariant energy descriptors are computed. The final frame-level distance is calculated as the weighted structural difference between the real and simulated energy descriptors. Because this metric remains sensitive to frame-level spatial intensity distributions while remaining tolerant to local geometric mismatches and small sensor/vehicle pose discrepancies, it is a more suitable tool for evaluating simulated LiDAR intensity in continuous simulation along extended trajectories.

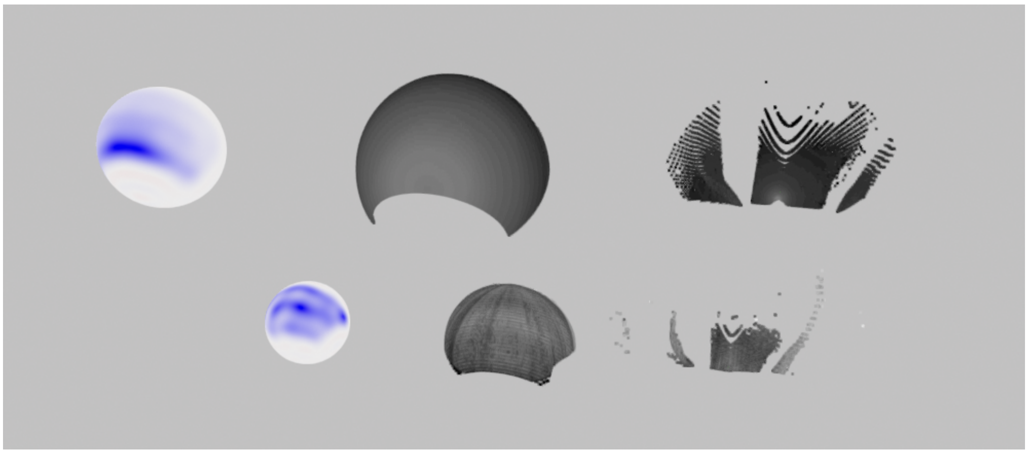


Figure 4.11: Workflow of the spherical harmonic energy distance evaluation. Raw point clouds are mapped into spherical intensity projections, followed by degree-wise spherical harmonic coefficient expansions to derive a rotation-invariant distance metric.

4.3.1 Validation of the Spherical Harmonic Evaluation Metric: Frame-Shift Sensitivity Analysis

To establish the mathematical validity, sensitivity, and baseline behavior of the proposed spherical harmonic energy distance metric prior to evaluating the simulation framework, an autocorrelation check was conducted. This check isolates the metric from any simulation artifact or effect by comparing a real-world log sequence from Scenario A against temporally shifted versions of itself.

This validation is framed through two different but complementary perspectives: a continuous temporal profile and an aggregated displacement analysis. First, the continuous real-world point cloud stream was evaluated against its own timeline using short, fixed frame offsets of $\Delta t \in \{0, 5, 10, 15\}$ frames. Because the sensor pose is different along the trajectory, increasing the frame offset directly correlates to a progressive geometric and environmental decorrelation between the paired frames, while a shift of zero frames serves as a baseline control.

Figure 4.12 illustrates the continuous evolution of the energy distance under these controlled frame-shift configurations. At an offset of zero frames ($\Delta t = 0$, representing identity mapping), the energy distance evaluates to exactly zero across the entire log sequence, verifying the absence of systemic mathematical bias or background noise within the calculation pipeline. Furthermore, as the frame shift scales to 5, 10, and 15 frames, the energy distance displays increasing or pronounced peaks and overall energy distance.

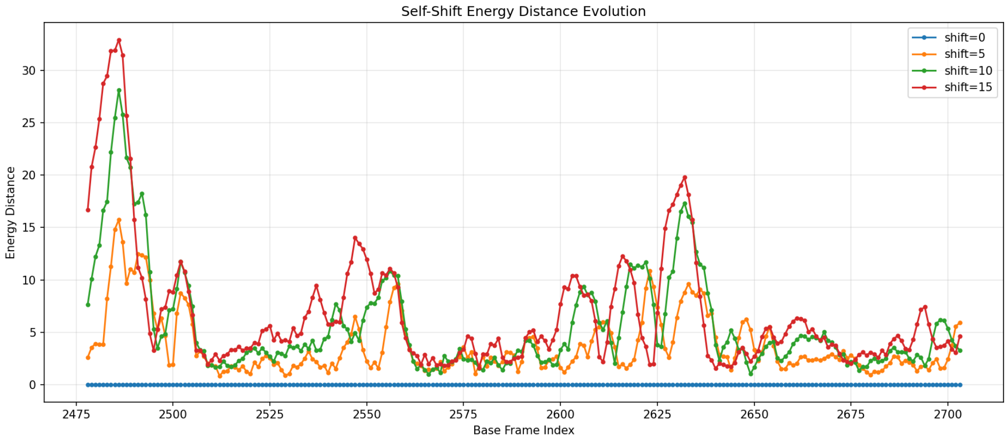


Figure 4.12: Autocorrelation validation of the spherical harmonic energy distance metric using short temporally shifted sequences ($\Delta t \in \{0, 5, 10, 15\}$ frames) of the real-world Scenario A log.

To map this behavior across a wider operational spectrum, a secondary analysis was conducted. Figure 4.13 plots the discrete profile of the energy distance computed across the entire recording for every individual frame shift step from 0 to 20 on the horizontal axis. Each individual displacement increment is represented by a central point marker indicating the energy distance mean, bounded by a vertical error bar denoting ± 1 standard deviation (σ).

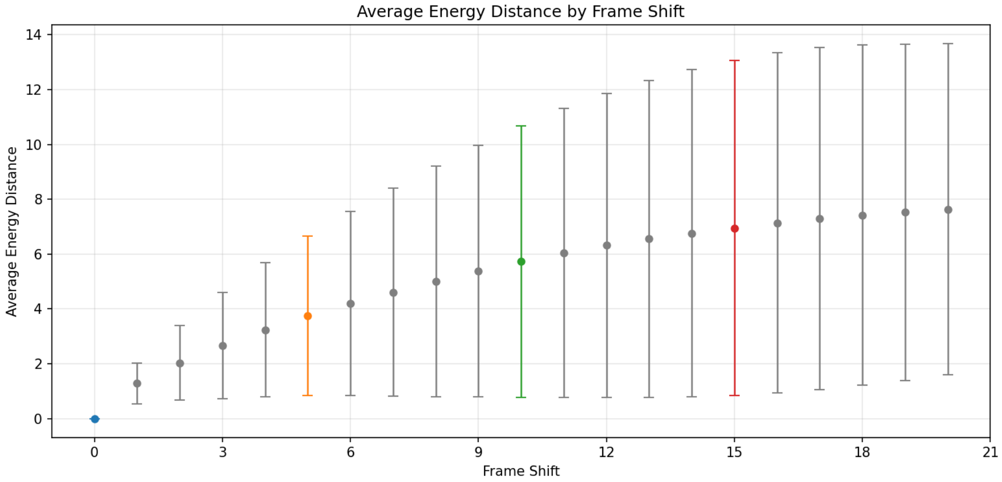


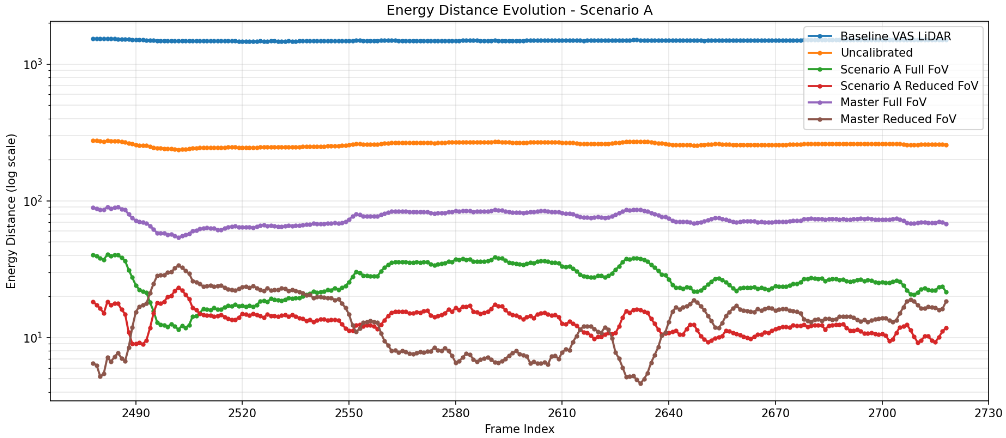
Figure 4.13: Sensitivity profile of the spherical harmonic energy distance as a function of discrete frame shift displacement from 0 to 20 frames. Central point markers denote the energy distance mean, while the vertical error bars represent the ±1 standard deviation (σ).

The resulting sensitivity curve reveals a clear and steady progression: as the frame displacement expands from 0 to 20, the discrete point markers' average value increase, reflecting the gradual decay of environmental similarity. The vertical error bars illustrate the structural variance encountered along the trajectory, where the expansion of the error intervals at higher frame shifts reflects the alternating geometry similarity of the environment in each of the scenes. Because the error bars display a controlled evolution and do not exhibit random overlapping between distant steps, this analysis confirms that the spherical harmonic energy distance metric works as a highly stable, predictable, and mathematically valid metric for spatial intensity distribution validation.

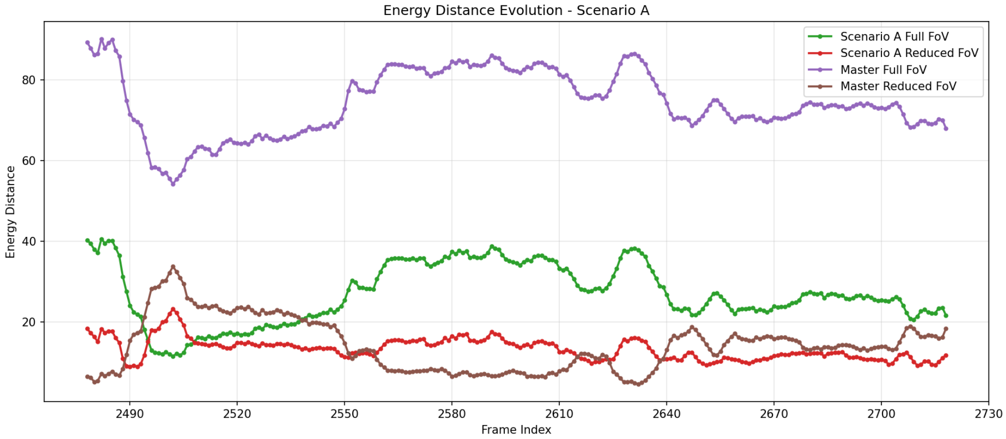
4.3.2 Continuous Distance Evolution in Reference Configurations (Scenarios A and B)

As it was done in the traditional evaluation metrics section, instead of limiting the evaluation with the SH energy distance metric to isolated static frames, the energy distance was computed continuously across the entire duration of each scenario log. This continuous analysis gives a better understanding of the sensor behavior and reaction to changes in the environment.

Figures 4.14 and 4.15 present the continuous time-series profiles of the energy distance along Scenario A (Dry) and Scenario B (Wet), respectively. To provide a thorough analysis of the different uncalibrated and calibrated models, the evaluation is divided into two distinct graphs. Panel (a) in each figure displays the comprehensive baseline comparison, plotting the original basic implementation (BL), the proposed model before fitting (PM-BF), the individual scene-specific fitting models' curves, and the two final master fitting curves ($\bar{F}^{\text{full}}_{\text{Ouster}}$ and $\bar{F}^{\text{red}}_{\text{Ouster}}$). Panel (b) focuses the analysis strictly to the four newly developed cubic fitting curves to examine the exact performance trade-offs between scene-specific and generalized parameter mappings.



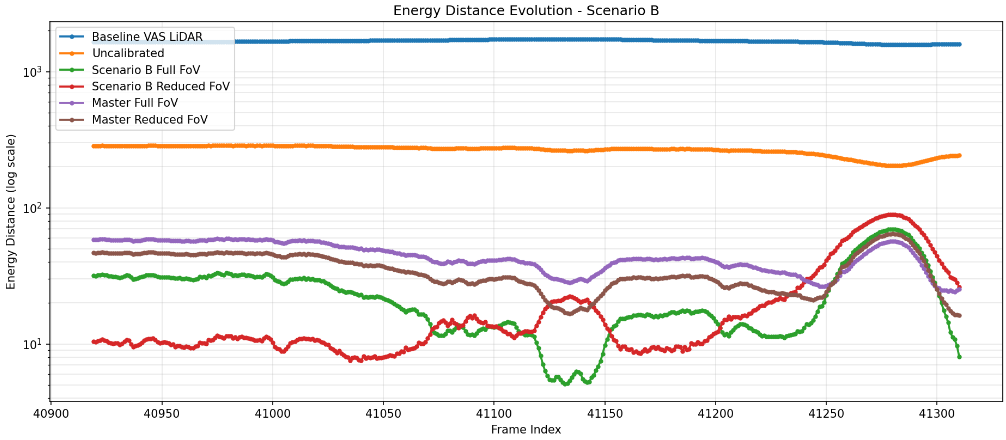
(a) All Models Comparison



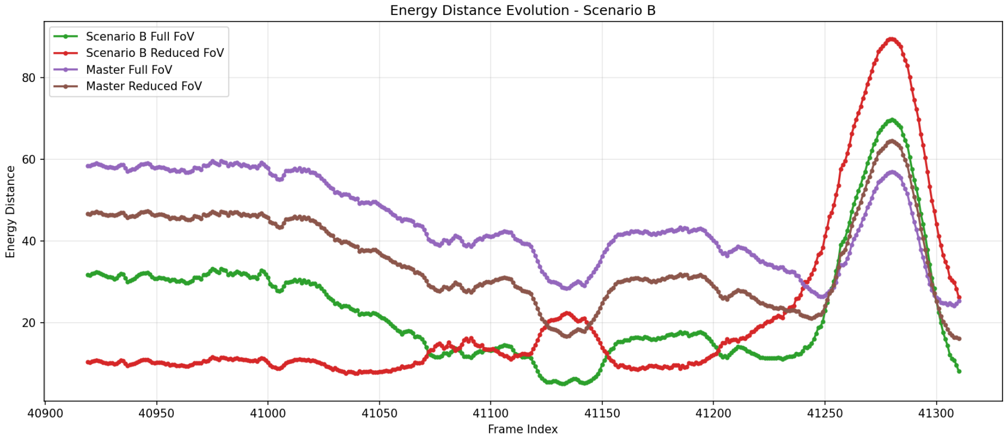
(b) Fitted Models Comparison

Figure 4.14: Continuous spherical harmonic energy distance evolution along the full recording of Scenario A (Dry Environment). Panel (a) contrasts the uncalibrated models against the fitted curves, while Panel (b) presents the differences between scene-specific and master curves.

Figures 4.14a and 4.15a reflect the high sim-to-real gap between the real sensor behavior and the simulated output of the baseline and uncalibrated models. However, once any variations of the cubic polynomial fitting are applied, the energy distance drops significantly and remains consistently low throughout the entire duration of the recordings.



(a) All Models Comparison



(b) Fitted Models Comparison

Figure 4.15: Continuous spherical harmonic energy distance evolution along the full recording of Scenario B (Wet Environment). Panel (a) contrasts the uncalibrated models against the fitted curves, while Panel (b) presents the variances between scene-specific and master curves.

When analyzing the models using the fitted curves (Figures 4.14b and 4.15b), a similar behavior to the one presented in the evaluation using WD, is observed: the scene-specific fitting curves which are optimized directly on the unique material configurations and geometries of their respective trajectories, yield in general the lowest energy distance values, but this might be due to an overfitting of the specific scenarios. More importantly, the master curves, especially the master Reduced-FOV curve $\bar{F}^{\text{red}}_{\text{Ouster}}$, closely track these scene-specific curves in both scenarios. This small performance degradation observed when switching from a scene-specific local curve to an averaged master function validates the stability of the parameter averaging process, confirming that the master equations retain spatial accuracy without overfitting to a single weather state.

4.3.3 Generalization Capabilities Across Unseen Trajectories (Scenarios C and D)

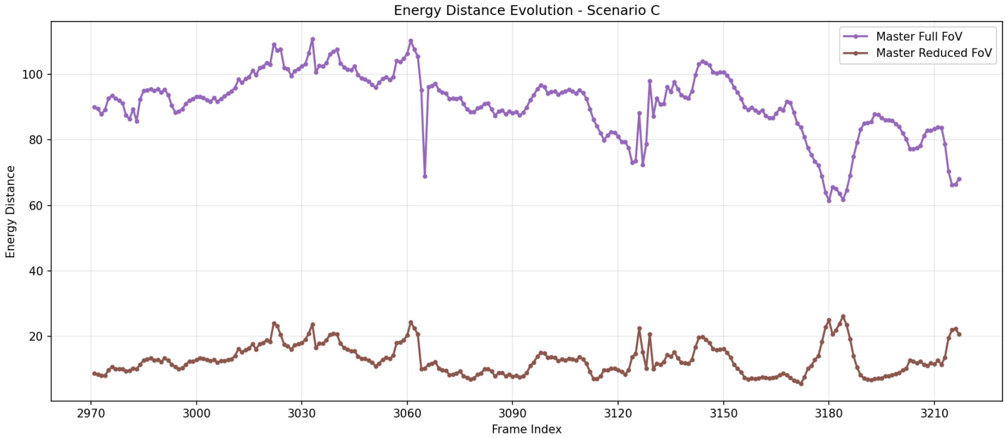
To evaluate the true capacity of generalization of the proposed framework and prevent over-parameterization, the master calibration curves were used again within the unseen testing environments, Scenario C and Scenario D. Because these scenarios were entirely omitted from the initial quantile-matching fitting phases, they serve as a strict validation layer. Hence, scene-specific curves are not analyzed for these trajectories.

Figure 4.16 tracks the continuous energy distance profiles along Scenarios C and D, comparing only the master Full-FOV ($\bar{F}^{\text{full}}_{\text{Ouster}}$) and master Reduced-FOV ($\bar{F}^{\text{red}}_{\text{Ouster}}$) equations against the uncalibrated baseline outputs. The steady energy distance values observed across the whole recordings, demonstrates that the master calibration curves do not only correct for localized spatial errors, but instead successfully represent a generalized sensor transfer function capable of executing reliably across diverse simulation environment layouts and weather conditions. It is observed that clearly the Reduced-FOV mapped curve $\bar{F}^{\text{red}}_{\text{Ouster}}$ outperforms the Full-FOV fitting curve $\bar{F}^{\text{full}}_{\text{Ouster}}$.

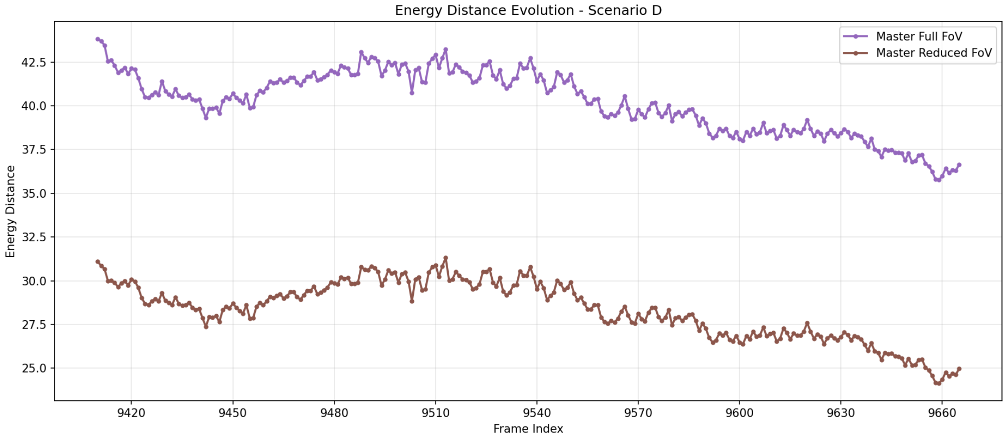
4.3.4 Quantitative Statistical Summary

To complement the time-series profiles with exact numerical bounds, a complete descriptive statistical analysis was conducted for every scenario across each model. Table 4.4 aggregates the calculated temporal mean, standard deviation (SD), temporal variance (Var.), minimum, and maximum values extracted from the continuous energy distance logs.

The numerical distributions show agreement and correlation with the visual trends from the time-series figures. The master curves exhibit not only a significantly lower mean distance compared to the baseline, but also a compressed standard deviation and tighter Min/Max bounds. This statistical decrease proves that the proposed empirical calibration method minimizes erratic intensity fluctuations and stabilizes the overall energy structure of the simulated point clouds.



(a) Scenario C: Master Curves



(b) Scenario D: Master Curves

Figure 4.16: Continuous spherical harmonic energy distance tracking across unseen Scenarios C and D, demonstrating the generalization capability of the two master calibration curves.

Table 4.4: Comprehensive descriptive statistics of the continuous frame-level spherical harmonic energy distance logs across all scenarios and model configurations. Lower values indicate closer agreement with the real-world sensor logs.

Scenario	**Model**	**Mean**	**Var.**	**SD (σ)**	**Min**	**Max**
A (Dry)	BL	1490.14	191.63	13.84	1467.91	1538.24
	PM-BF	259.11	75.39	8.68	236.71	276.01
	$F_{A,\text{full}}$	26.91	54.54	7.38	11.48	40.54
	$F_{A,\text{red}}$	13.31	6.94	2.63	8.96	23.19
	$\bar{F}^{\text{full}}_{\text{Ouster}}$	74.22	64.43	8.03	54.17	90.10
	$\bar{F}^{\text{red}}_{\text{Ouster}}$	14.24	42.18	6.49	4.63	33.71
B (Wet)	BL	1676.12	1835.56	42.84	1575.45	1738.01
	PM-BF	266.77	481.33	21.94	204.23	287.36
	$F_{B,\text{full}}$	23.63	205.71	14.34	5.10	69.65
	$F_{B,\text{red}}$	21.01	426.55	20.65	7.58	89.47
	$\bar{F}^{\text{full}}_{\text{Ouster}}$	44.60	110.22	10.50	24.17	59.55
	$\bar{F}^{\text{red}}_{\text{Ouster}}$	35.54	124.91	11.18	16.19	64.46
C (Dry)	$\bar{F}^{\text{full}}_{\text{Ouster}}$	90.93	98.30	9.91	61.37	110.88
	$\bar{F}^{\text{red}}_{\text{Ouster}}$	12.88	20.50	4.53	5.56	26.13
D (Wet)	$\bar{F}^{\text{full}}_{\text{Ouster}}$	40.20	3.29	1.81	35.76	43.84
	$\bar{F}^{\text{red}}_{\text{Ouster}}$	28.34	2.73	1.65	24.15	31.32

BL: baseline model; PM-BF: proposed model before fitting;
$F_{A,\text{full}}$ / $F_{B,\text{full}}$: localized Full Field-of-View (FOV) fitting curves;
$F_{A,\text{red}}$ / $F_{B,\text{red}}$: localized Reduced-FOV fitting curves;
$\bar{F}^{\text{full}}_{\text{Ouster}}$: master Full-FOV fitting curve;
$\bar{F}^{\text{red}}_{\text{Ouster}}$: master Reduced-FOV fitting curve.

5

Discussion

This chapter discusses the main findings from the results and their implications for LiDAR intensity simulation and evaluation. The discussion focuses on three aspects: the interpretation of the proposed intensity model, the role of vendor-specific fitting, and the difference between traditional distribution-based metrics and the proposed geometry-tolerant evaluation method. Limitations are discussed within each part where they are most relevant.

5.1 Interpretation of the Intensity Model Results

This section discusses how the intensity model results should be interpreted. The focus is not only on whether the proposed model gives a closer match to the real-world reference data, but also on the factors that limit this comparison. These include the choice of fitting data, the difficulty of reproducing complex materials in the digital twin, variations in surface reflectivity under different conditions, and the remaining gap caused by sensor-specific processing.

5.1.1 Overall Improvement of the Proposed Model

The results indicate that the proposed intensity model improves the agreement between simulated LiDAR intensity and the real-world reference data. Compared with the baseline model, the proposed model introduces physically motivated dependencies on range, incidence angle, and material reflectivity. These factors allow the simulated intensity to vary more realistically across different parts of the scene, instead of being determined only by a simple or fixed intensity assignment.

This improvement is visible in the intensity distributions and in the numerical comparison metrics. After applying the complete intensity pipeline, the simulated intensity distribution becomes closer to the reference distribution. This suggests that the proposed model captures part of the behavior observed in the real LiDAR data. In particular, it helps represent the fact that different surfaces do not produce the same return strength, and that the same surface can produce different responses depending on distance and viewing angle.

However, the improvement should be interpreted carefully. The reference intensity used in this thesis is a calibrated reflectivity-derived quantity rather than raw received optical power. Therefore, the objective is not to reproduce the complete physical sensing process inside the LiDAR sensor. Instead, the model aims to produce simulated intensity values that are more comparable to the recorded reference representation. In this sense, the results show improved similarity to the real-world reference data, but they should not be interpreted as a full physical validation of the sensor response.

5.1.2 Choice of Fitting Data

A key factor in the fitting result is the choice of data used to estimate the relationship between the simulated model output and the real-world reference intensity. In principle, it would be desirable to fit the model using data that covers the full range of reflectivity values observed in a real driving environment. This would include low-reflectivity road surfaces, medium-reflectivity infrastructure, vegetation, vehicles, signs, lane markings, and other highly reflective objects. In practice, however, constructing such a fitting dataset is difficult because the same range of materials and objects must be present in both the real-world recording and the corresponding simulated scene.

Two possible strategies can be considered. The first strategy is to use a LiDAR configuration or selected view where the sensor mainly observes the ground. In this case, the visible material range is more limited and the scene is easier to control. The intensity distribution is dominated by a smaller number of surface types, such as asphalt, gravel, dirt, or rubble-like road materials. Since these materials are relatively easier to reproduce in the digital twin, the comparison between real and simulated data becomes more stable. This makes the fitting result easier to interpret, because fewer errors are introduced by mismatched objects or complex material composition.

The second strategy is to use a LiDAR view that observes a wider surrounding environment. This can include vegetation, buildings, road infrastructure, other trucks, fences, and other scene objects. Such data may cover a broader reflectivity range and may therefore seem more suitable for fitting a general intensity model. However, it is also much harder to reproduce accurately in simulation. Even when the simulated scene is built to correspond to the real environment, there may be differences in object placement, object shape, surface material, and local details. These differences affect the fitting result and make it harder to distinguish whether an error is caused by the intensity model or by the mismatch between the real and simulated scene.

For this reason, the main experiments focus on the more controlled fitting setup. The ground-focused case provides a cleaner comparison because the number of dominant materials is smaller and the simulated scene can better approximate the real one. This does not limit the model's utility to ground surfaces, but indicates that the fitting result from this setup is less affected by uncontrolled scene differences. A wider surrounding-scene fitting setup can still be useful in future work, but its result should be interpreted with greater caution.

5.1.3 Scene and Material Mismatch

One important limitation of fitting the model using complex scenes is the mismatch between real-world materials and their simulated representation. In the real world, a single object can contain many different surface materials. For example, a truck may contain rubber tires, glass windows, painted metal, plastic parts, lights, and reflective elements. Each of these surfaces can have a different reflectivity response. In the simulation, however, the ray-casting pipeline may only provide a simplified material or object-level description at the hit point. If the whole object is represented using one material class, then the simulated return cannot reproduce the material variation present in the real object.

This creates a gap between the real and simulated intensity even if the geometric position of the object is approximately correct. A point hitting a truck in the real-world data may correspond to a tire, a window, or a painted metal surface, while the corresponding simulated point may only be treated as a generic truck material. The fitted model must then compensate for these material-label errors, which can distort the estimated mapping. In this case, the fitting error does not only reflect the quality of the intensity model, but also the quality of the scene representation and material annotation.

The same problem also appears for other complex objects. Vegetation can vary strongly depending on leaf density, branch structure, moisture, and viewing direction. Buildings can contain glass, concrete, metal, painted surfaces, and shadows. Road infrastructure can include signs, poles, lane markings, barriers, and reflective materials. If these are simplified in the digital twin, then the simulated intensity distribution may differ from the real-world reference data for reasons unrelated to the intensity formula itself.

This complicates fitting over the full reflectivity range. A broad material range is useful for learning a general mapping, but it also increases the probability of mismatch between the real and simulated scene. Therefore, the fitting result should be interpreted as the combined effect of the intensity model, the material assignment, and the fidelity of the digital twin. Improving the intensity model alone may not be sufficient if the simulation does not provide sufficiently accurate material information at the ray hit points.

5.1.4 Reflectivity Variation Under Different Conditions

Another source of uncertainty is that real surface reflectivity is not constant. Even for the same material, the measured LiDAR intensity can change depending on surface condition and environment. Road surfaces may appear different when they are dry, wet, dusty, worn, or covered by small particles. Real surfaces also vary due to aging, dirt, moisture, texture, and local roughness. These factors change how the laser pulse interacts with the surface and can alter the returned intensity.

Weather conditions further increase this variation. In sunny and dry conditions, the reflectivity behavior may be relatively stable for many road materials. Under rainy conditions, however, wet surfaces can produce different responses due to changes in surface scattering and specular reflection. Water on asphalt, for example, may change both the magnitude and angular behavior of the returned signal. Rain, fog, or airborne particles can also affect propagation and the received return. Therefore, a mapping fitted under one condition may not directly generalize to another condition.

In the current work, the main experiments are based on controlled conditions, such as sunny or dry data. This makes the fitting problem more manageable and reduces the number of uncontrolled variables. However, it also means that the resulting model should not be assumed to fully describe all weather conditions. A possible extension is to perform separate fittings for different environmental conditions, such as dry and rainy cases. These fitted relationships could then be interpolated or blended based on a rain-level or surface-wetness parameter.

Such interpolation would still be an approximation. The real change in reflectivity under weather variation is likely nonlinear and material-dependent. For example, wet asphalt, wet vegetation, and wet metal surfaces may not change in the same way. The effect may also depend on incidence angle and range. Therefore, a simple averaged or interpolated mapping can improve practical simulation behavior, but it should not be interpreted as a complete physical model of weather-dependent reflectivity.

5.1.5 Sensor-Specific Processing Gap

A final gap comes from the sensor-specific processing of the recorded intensity value. The simulated model computes a physically motivated return value based on range, incidence angle, and material reflectivity. However, the intensity stored in the real-world point cloud is not simply the raw optical power received by the sensor. It is a calibrated and encoded value produced after internal sensor processing. This processing may include range compensation, sensitivity calibration, nonlinear scaling, saturation handling, noise filtering, and other vendor-specific operations.

Because the complete internal processing pipeline is not available, it is difficult to derive an exact analytical function that maps the simulated physical return to the recorded reference intensity. Simple functions such as linear, logarithmic, power-law, or saturating mappings can approximate part of the relationship, but they cannot guarantee a perfect match over the full intensity range. This is especially true for very low or very high reflectivity values, where sensor noise, thresholding, saturation, or nonlinear encoding may have a stronger influence.

Therefore, the fitted mapping used in this thesis should be understood as an empirical approximation. Its role is to make the simulated output comparable to the reference intensity representation, not to reveal or reproduce the exact internal behavior of the sensor. This distinction is important when interpreting the results. A close fit indicates that the simulated intensity can be adjusted to resemble the recorded calibrated reflectivity distribution under the evaluated conditions. It does not imply that the underlying sensor electronics or proprietary processing have been physically reconstructed.

This sensor-specific gap also affects generalization. A mapping fitted for one sensor type, configuration, or data representation may not transfer directly to another LiDAR sensor or another intensity encoding. Even for the same sensor, different firmware settings, return profiles, or calibration procedures may affect the recorded intensity values. As a result, the proposed pipeline should be seen as a flexible framework: the physically motivated model provides the basic structure, while the empirical fitting step adapts the output to the specific reference data used for evaluation.

5.2 Comparison Between Traditional Metrics and the Proposed Evaluation Method

This section discusses the difference between the traditional distribution-based metrics and the proposed spherical harmonic energy-based evaluation method. Both types of metrics are useful, but they measure different aspects of the similarity between real and simulated LiDAR intensity. The traditional metrics provide a direct comparison of intensity value distributions, while the proposed method also considers how intensity is distributed around the sensor.

5.2.1 Role of Traditional Distribution-Based Metrics

The Wasserstein distance and Jensen–Shannon distance provide useful baseline metrics for comparing real and simulated LiDAR intensity. They are simple to compute, do not require point-wise correspondence, and directly measure whether two frames contain similar intensity values in similar proportions. This makes them suitable for a first comparison between the reference data and different simulation models.

However, these metrics only compare the intensity distribution of each frame. They do not consider the spatial or angular location of the points. For example, a simulated frame may contain a similar number of low-, medium-, and high-intensity points as the real-world reference frame, but these values may appear in different directions or on different scene structures. In this case, the histogram-based distance may still be low even though the simulated intensity pattern is not physically or visually similar to the real frame.

Conversely, a high distribution-based distance may occur even when the angular structure is well preserved. A simulated frame may reproduce the general directional structure of the real intensity pattern, but still have a global scale difference or a slightly different intensity distribution. Traditional distribution-based metrics may then report a larger difference, even though the simulated frame preserves important structural characteristics. Therefore, these metrics are useful for measuring global intensity statistics, but they are not sufficient for evaluating the spatial organization of intensity within the LiDAR frame.

5.2.2 Need for Directional Intensity Comparison

A more informative evaluation should consider not only which intensity values occur, but also where they occur around the sensor. In LiDAR data, different directions correspond to different scene regions, such as road surfaces, vegetation, buildings, or vehicles. A realistic simulation should therefore reproduce not only the overall intensity distribution, but also its approximate angular structure.

At the same time, exact point-wise comparison is not reliable because the digital twin is not a perfect copy of the real environment, the geometry used for ray casting is simplified, and small pose differences can shift many points. The proposed spherical harmonic energy distance addresses this by representing each frame as an angular intensity function and comparing degree-wise energy, avoiding direct point-to-point matching while preserving directional structure.

In this thesis, relatively low-order spherical harmonic components and decreasing degree weights were used. This emphasizes the overall angular pattern rather than high-frequency local details, making the descriptor more tolerant to local geometric mismatch and small pose differences.

5.2.3 Limitations of the Spherical Harmonic Energy Method

Although the spherical harmonic energy distance provides a useful geometry-tolerant frame-level comparison, it also has several limitations. First, because the descriptor uses a limited maximum spherical harmonic degree and emphasizes lower-order components, it mainly captures coarse angular intensity structure. Small high-reflectivity objects, narrow lane markings, or sharp local intensity changes may be suppressed. Increasing the maximum degree would allow the method to capture finer details, but it would also make the descriptor more sensitive to noise, non-uniform sampling, and local geometric mismatch. Therefore, the choice of maximum degree and weighting scheme controls the balance between robustness and sensitivity.

Second, the method produces a frame-level distance rather than a localized error map. It can indicate whether two frames have similar angular intensity structure, but it does not directly identify which object or region causes the mismatch. This limits its ability to diagnose local simulation errors. For example, a high distance value may indicate a difference in angular intensity structure, but additional qualitative inspection or local region-based analysis is still needed to determine whether the error comes from material mismatch, missing objects, pose error, or intensity scaling.

Third, the coefficient estimation can be affected by the LiDAR field of view, scan pattern, point density, and preprocessing choices such as invalid-point filtering and intensity normalization. Since the spherical harmonic coefficients are estimated from discrete LiDAR samples rather than from a continuous function on the sphere, non-uniform angular sampling can influence the fitted coefficients. This means that the descriptor may reflect not only the underlying intensity structure, but also the sampling properties of the LiDAR sensor and the preprocessing pipeline.

A further limitation is related to angular coverage. The spherical harmonic representation is most meaningful when the LiDAR frame provides sufficiently broad coverage of the spherical domain. In the ideal case, the intensity samples cover a large field of view around the sensor, so that the fitted spherical harmonic coefficients are constrained by observations from many directions. If the available data only occupy a narrow angular sector, for example a limited horizontal range, the function approximation on the sphere becomes poorly constrained outside the observed region. In such cases, the estimated coefficients may be strongly influenced by the limited sampling pattern rather than by the true global angular intensity structure. Therefore, the spherical harmonic energy distance is more suitable for full-frame or wide-field LiDAR comparisons, and should be interpreted with caution when applied to narrow field-of-view subsets.

5.2.4 Complementary Interpretation of the Metrics

The traditional distribution-based metrics and the proposed spherical harmonic energy distance should be interpreted as complementary rather than competing measures. The Wasserstein and Jensen–Shannon distances answer whether the simulated frame has a similar overall intensity distribution to the reference frame. The spherical harmonic energy distance answers whether the simulated frame has a similar angular intensity structure.

When both types of metrics improve, this suggests that the simulated intensity becomes closer to the reference data both statistically and structurally. If the distribution-based metrics improve but the spherical harmonic energy distance does not, the simulated model may reproduce the correct intensity histogram but place the intensity values in different parts of the scan. If the spherical harmonic energy distance decreases but the histogram metrics do not, the simulated frame may better reproduce the directional structure while still having differences in the global intensity scale or distribution.

For these reasons, the spherical harmonic energy distance should not be interpreted as a complete replacement for traditional metrics or qualitative inspection. Instead, it provides an additional evaluation perspective that is more informative than global histograms when geometric mismatch is present, while remaining less sensitive to local alignment errors than point-wise comparison.

5.3 Future Work

Future work should evaluate the proposed pipeline on a larger number of frames, locations, and driving scenarios. This would make it possible to assess how well the fitted vendor-specific mapping generalizes beyond the selected evaluation data.

The material reflectivity model could also be improved by using more detailed spectral and angular reflectance data for relevant road-scene materials, such as asphalt, lane markings, concrete, vegetation, and metallic objects. Environmental effects such as wet surfaces, rain, fog, and changing illumination conditions could also be included.

The empirical mapping could be further developed using a larger calibration dataset or more flexible regression methods. This may improve the approximation of calibrated reflectivity behavior, especially for nonlinear response regions and high-reflectivity surfaces.

Finally, the spherical harmonic evaluation method could be extended by studying the influence of maximum harmonic degree, weighting scheme, and frame selection strategy. It could also be combined with local region-based descriptors to preserve more spatial detail while maintaining tolerance to imperfect alignment between the real environment and the digital twin.

6

Conclusion

This thesis investigated realistic LiDAR intensity simulation for autonomous driving systems using a CARLA based digital twin of the AstaZero proving ground. The work focused on two main objectives: improving the simulated LiDAR intensity model and developing an evaluation method that can compare simulated and real-world intensity without requiring exact point-wise alignment.

The real-world reference data were reconstructed from MCAP recordings containing PointCloudSliver messages. The intensity values were interpreted as a vendor-specific calibrated reflectivity representation. In the simulation environment, a physically motivated intensity model was introduced based on range dependent attenuation, incidence angle, and material reflectivity. Since the reference intensity is sensor-processed, an empirical vendor-specific mapping was fitted to transform the simulated return values into a calibrated reflectivity representation.

The results show that the physically motivated model alone does not directly match the intensity distribution of the reference data. This is expected because the reference intensity includes vendor-specific calibration and encoding. After applying the fitted mapping, the simulated intensity distribution becomes more comparable to the real world reference distribution. Traditional distribution based metrics provide a useful frame level comparison of intensity histograms, but they do not consider the angular or geometric distribution of the points.

To address this limitation, a spherical harmonic energy based evaluation method was proposed. The method represents each LiDAR frame as an angular intensity function and compares degree wise spherical harmonic energy descriptors. This provides a geometry tolerant frame level comparison that preserves information about the angular intensity structure while avoiding strict point-wise correspondence.

Overall, the thesis demonstrates a practical pipeline for reconstructing real world LiDAR reference frames, generating physically motivated simulated intensity, fitting a vendor-specific intensity mapping, and evaluating sim-to-real intensity similarity using both traditional and geometry-tolerant metrics.

Bibliography

[1] A. G. Kashani, M. J. Olsen, C. E. Parrish, and N. Wilson, “A Review of LIDAR Radiometric Processing: From Ad Hoc Intensity Correction to Rigorous Radiometric Calibration”, *Sensors*, vol. 15, no. 11, pp. 28099–28128, 2015.

[2] N. Sanchiz-Viel, E. Bretagne, E. Mouaddib, and P. Dassonvalle, “Radiometric correction of laser scanning intensity data applied for terrestrial laser scanning”, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 172, pp. 1–16, 2021.

[3] S. Muckenhuber, H. Holzer, J. Rübsam, and Z. Böcskei, “Automotive Lidar Modelling Approach Based on Material Properties and Lidar Capabilities”, *Sensors*, vol. 20, no. 11, article 3309, 2020.

[4] A. Haider, M. Pigniczki, M. H. Köhler, and A. W. Koch, “Development of High-Fidelity Automotive LiDAR Sensor Model with Standardized Interfaces”, *Sensors*, vol. 22, no. 19, article 7556, 2022.

[5] S. Li, D. Zheng, D. Yue, C. Hu, and X. Ma, “A Method for Point Cloud Accuracy Analysis Based on Intensity Information”, *Sensors*, vol. 23, no. 22, p. 9135, 2023.

[6] W. Y. Yan, A. Shaker, A. Habib, and A. P. Kersting, “Improving classification accuracy of airborne LiDAR intensity data by geometric calibration and radiometric correction”, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 67, pp. 35–44, 2012.

[7] V. Anand, B. Lohani, G. Pandey, and R. Mishra, “Advancing LiDAR Intensity Simulation Through Learning With Novel Physics-Based Modalities”, *IEEE Transactions on Intelligent Transportation Systems*, 2025.

[8] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An Open Urban Driving Simulator”, in *Proceedings of the 1st Annual Conference on Robot Learning (CoRL)*, vol. 78 of *Proceedings of Machine Learning Research*, pp. 1–16, 2017.

[9] C. Li, J. Sifakis, Q. Wang, R. Yan, and J. Zhang, “Simulation-Based Validation for Autonomous Driving Systems”, arXiv:2301.03941, 2023.

[10] J. Wishart, S. Como, U. Forgione, J. Weast, and others, “Literature Review of Verification and Validation Activities of Automated Driving Systems”, *SAE International Journal of Connected and Automated Vehicles*, vol. 3, no. 4, pp. 267–323, 2020.

[11] CARLA Team, “CARLA Simulator Documentation”, Available online: `https://carla.readthedocs.io/`.

[12] CARLA Simulator Documentation, “Python API reference”, 2026. Available online: `https://carla.readthedocs.io/en/latest/python_api/`.

[13] Foxglove, “MCAP File Format Specification”, Available online: `https://mcap.dev/`.

[14] Ouster, Inc., “Ouster Sensor Docs: Sensor Data Architecture and Return Profiles”, Available online: `https://static.ouster.dev/sensor-docs/image_route1/image_route3/sensor_data/sensor-data.html#rng19-rfl8-sig16-nir16-return-profile`.

[15] Ouster, Inc., “Ouster Sensor Docs: Reflectivity Data Mapping”, Available online: `https://static.ouster.dev/sensor-docs/image_route1/image_route3/sensor_data/sensor-data.html#reflectivity-data-mapping`.

[16] D. J. Ritter, R. Rott, B. Schlager, S. Muckenhuber, S. Genser, M. Kirchengast, and M. Hennecke, “Angle-dependent spectral reflectance material dataset based on 945 nm time-of-flight camera measurements”, *Data in Brief*, vol. 48, p. 109031, 2023. `https://doi.org/10.1016/j.dib.2023.109031`

[17] P. Virtanen et al., “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”, *Nature Methods*, vol. 17, pp. 261–272, 2020. `https://doi.org/10.1038/s41592-019-0686-2`

[18] H. Holzhüter, J. Bödewadt, S. Bayesteh, A. Aschinger, and H. Blume, “Technical concepts of automotive LiDAR sensors: a review”, *Optical Engineering*, vol. 62, no. 3, p. 031213, 2023.

[19] S. Manivasagam, I. A. Bârsan, J. Wang, Z. Yang, and R. Urtasun, “Towards Zero Domain Gap: A Comprehensive Study of Realistic LiDAR Simulation for Autonomy Testing”, in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 8272–8282.

[20] SciPy Documentation, “scipy.stats.wasserstein_distance”, `https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.wasserstein_distance.html`.

[21] SciPy Documentation, “scipy.spatial.distance.jensenshannon”, `https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.jensenshannon.html`.

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY