LiDAR Data Augmentation for Semantic Segmentation: A Statistical Intensity Model

Pablo Aguilar-Pérez [1,2,*] and Rebeca P. Díaz-Redondo [1,*]

[1] atlanTTic Research Center, Information & Computing Lab. Telecommunication Engineering School, Universidade de Vigo. Vigo, 36310, Spain
[2] Centro Tecnológico de Automoción de Galicia, Avenida Principal, 2, 36475 O Porriño, Pontevedra, Spain
Email: pablo.aguilar@uvigo.gal (P.A); rebeca@det.uvigo.es (R.D.)
*Corresponding author

***Abstract*—Light Detection and Ranging (LiDAR) sensors play a crucial role in autonomous driving, specially in HDMaps generation systems, which contain semantic information of the road scenarios. Gathering this information is not easy, so in the literature deep learning-based semantic segmentation on LiDAR point clouds is used to accelerate this process. However, it requires large amounts of data for training, which are time-consuming and computationally intensive to obtain. In this paper, we introduce a method to use the synthetic dataset KITTI-CARLA to reduce the amount of real data needed to train a Deep Learning-based model on SemanticKITTI. The main challenge is that in KITTI-CARLA the intensity field is not computed. To overcome this, we define a statistic module that infer the real distribution of intensities of SemanticKITTI into KITTI-CARLA. We tested our method with the Point-to-Voxel-KD model. Extensive experiments show that results near the state of the art can be obtained with half the amount of real data when KITTI-CARLA is incorporated to the training set. The source code for the intensity distribution calculation and the conversion process from KITTI-CARLA to SemanticKITTI is published in https://github.com/pabloaguilarp/KITTI-CARLA-converter.**

***Keywords*—Light Detection and Ranging (LiDAR), deep learning, semantic segmentation, simulation**

I. Introduction

Semantic segmentation on Light Detection and Ranging (LiDAR) point clouds is a crucial part of the workflow in the field of autonomous driving, both for real time on-board systems and for offline mapping systems. LiDAR sensors, unlike cameras, capture data in an unorganized manner, which makes the use of convolutional networks hard to optimize.

Deep Learning-based algorithms for semantic segmentation are known to provide the most accurate results for LiDAR point clouds [1]. These algorithms are usually tested against public data sets, such as SemanticKITTI [2] or nuScenes [3], that provide a leaderboard where researchers can submit their algorithms and publish their results. Despite of the clear advantages of these tools, there are still some limitations due to the nature of the deep learning algorithms, which are really good to perform specific tasks but not flexible enough at performing different tasks that the one they have been trained for. In the domain of semantic segmentation on LiDAR point clouds, this entails that segmenting the SemanticKITTI dataset and the nuScenes dataset are two different tasks, even the same backbone Neural Network (NN) is used. Therefore, the NN must be trained twice, once with each dataset. What is more, if there is no public data for a given use case, the training data must be acquired and manually labeled.

Within this context, our intention is to leverage the advancements in simulation technologies to reduce the amount of real data needed to train the Deep Learning-based system by providing synthetic data to the training set. Our proposal contributes to the state-of-the-art in the following aspects: (i) It extends the existing SemanticKITTI dataset with augmented existing synthetic data; (ii) then we introduce a labels matching method that combines two datasets with different labels; and, finally, (iii) we implemented a statistical intensity module able to infer the intensities of SemanticKITTI data.

The structure of the paper is as follows. In Section II.A, a review of the different public datasets that nowadays exist is made, in Section II.B, the different simulation technologies and solutions are explained, and in Section II.C, the evolution of the different approaches for Deep Learning-based semantic segmentation algorithms is described. In Section III, we define the mathematical notation of the problem and we discuss the solution design. In Section IV.A, the conducted experiments and their results are exposed, and in Section V, the general conclusions of the work are discussed.

II. Related Work

As it was previously mentioned, having a large, diverse and already labeled set of data is crucial for Deep Learning (DL)/Machine Learning (ML)-based algorithms to offer their best results. This is, indeed, the most demanding task in terms of material and human resources,

Manuscript received August 23, 2024; revised September 26, 2024; accepted November 12, 2024; published February 17, 2025.

since massive amounts of data must be gathered first and manually labeled afterwards. Moreover, in order to capture the data it is required a mapping system, usually in the form of a Mobile Mapping System (MMS). An MMS generally consists of one or several LiDAR sensors and a positioning system such as a Global Navigation Satellite System (GNSS) and Inertial Measurement Unit (IMU). Additionally, it also can have some RGB and stereo cameras. This mapping system is time consuming and computationally intensive, but there are two tools that allow the research community to overcome these difficulties: using public data sets (Section II.A) and using simulation tools and synthetic data (Section II.B). Additionally, in this section we also summarize the most relevant advances in deep learning based semantic segmentation models (Section II.C).

A. Public Datasets

Public datasets are generally gathered by institutions and are manually labeled, offering a straightforward solution for researchers. However, this also has a drawback: it is not possible to extend the datasets with more data, that cannot be collected in the same way, and even if it would be possible to manually re-label the whole dataset if different labels are needed, the process would be extremely tedious. Anyway, for the sake of automotive segmentation tasks, there are two main types of datasets: road-level and urban-level.

The road-level datasets are generally collected using a land vehicle with an MMS system. Its scope is the road and its surroundings, within a maximum radius of a few tens of meters. The main available road-level datasets available are the following ones: SemanticKITTI [2] (based on the KITTI dataset [4]), nuScenes [3], Waymo dataset [5], Toronto-3D [6], Paris-Lille-3D [7], SensatUrban [8], Semantic-POSS [9], A2D2 [10], Argoverse [11], Lyft dataset [12], CSPC dataset [13] and Semantic3D [14].

The urban-level datasets are collected by Aerial Laser Systems (ALS) and its coverage area is generally wider. The most widely used urban-level public datasets are DublinCity [15], DALES [16], SUM [17], Swiss3DCities [18] and UrbanScene3D [19].

It is interesting to note that there is another classification of datasets: bounding-boxed labeled datasets and point-wise labeled datasets. The former are labeled by creating a box that encloses the points corresponding to the object they represent. This is a simple way of labeling but in contrast it makes it more difficult to characterize instances with complex geometries or continuous uncountable elements (like the road). Contrariwise, the latter are assigned a label to each point, so the entire scene gets characterized. Since we focus on semantic segmentation, it is interesting having datasets that are labeled pointwise, this is, assign a class to each point in the dataset. According to this, the bounding-box labeled datasets (Waymo, A2D2, Argoverse and Lyft) are not suitable. However, the most influential point-wise labeled datasets are the ones we intend to use: SemanticKITTI, nuScenes and Waymo.

Additionally, all the previous datasets feature different number of classes (Table I) and include different classes that can be relevant depending on the application. The most important labels, namely car, road, building, vegetation and pole, are present along the majority of datasets. However, nuScenes is a special case since it lacks the road class or any other ground or surface class, although it focuses on a more detailed objects suite instead. For example, nuScenes dataset differs from bendy or rigid bus, or from adult or child pedestrian. It also has not so frequent labels like wheelchairs, debris or bicycle racks, which may be interesting for some use cases. SemanticKITTI and Waymo, on another hand, have a similar distribution of classes. Finally, Paris-Lille-3D is the most complete and varied dataset in terms of its classes. It distinguishes between several types of vehicles (apart from the usual car, truck and bus, it has different labels for scooter motorcycles and vans) and many objects that generally are not included in the rest of datasets. It is also the only dataset that has labels such as bollard, mailbox, windmill, statue, bench, table or chair.

TABLE I. NUMBER OF CLASSES IN VARIOUS DATASETS

Dataset	Classes	Dataset	Classes
SemanticKITTI	28	Semantic-POSS	14
nuScenes	23	A2D2	38
Waymo	23	Argoverse	30
Toronto-3D	9	Lyft	12
Paris-Lille-3D	50	CSPC	6
SensatUrban	13	Semantic3D	8

Attending the data volume, this is, the total size of the dataset in terms of scans, these datasets show a wide range of values. Of all the datasets previously mentioned, the biggest (in number of labeled points) are Waymo, Lyft, SemanticKITTI and Semantic3D.

B. Simulation and Synthetic Data

Simulated environments are often used to generate massive amounts of synthetic data. This approach makes it possible to obtain datasets with enormous flexibility, in addition to allowing the generation of rare cases that would hardly be possible to capture in the real world. Both licensed and open-source solutions are available and enable the simulation of road scenarios and the generation of synthetic data. We highlight here the most relevant ones for our purpose.

CarMaker [20] is a licensed software that includes a big collection of assets and permits the simulation of scenarios under various traffic and weather conditions. It includes a LiDAR sensor simulator that can be used for labeled-data generation. RoadRunner [21] is a part of the MATLAB ecosystem that allows the simulation of road scenarios. LiDAR sensors can be simulated, but no annotated data is generated. Anyverse [22] is a Spanish software that approaches the scenario and sensor simulation process in a realistic, physics-based manner. It allows the generation of annotated LiDAR data with a varied set of pre-created, ready to use assets, and custom 3D entities can also be added. Synkrotron [23] provides a road scenario simulator and a LiDAR sensor simulator,

and annotated point clouds can be obtained. monoDrive [24] is developed by National Instruments and makes the simulation of LiDAR sensors possible. It includes a semantic LiDAR module that can be captured via a dedicated TCP/IP client that connects to the virtualized LiDAR and emulates a data flow in the real format. In this sense, data can be retrieved using the standard decoders as if they were real sensors. As a drawback, only Velodyne VLP16 and HDL32 units are supported.

Various open-source solutions exist to simulate road scenarios and generate synthetic data from them. For the purpose of synthetic datasets generation, the most influential is CARLA [25]. CARLA is an open-source software that allows the simulation of road scenarios with various road sensors and different weather conditions. It features great flexibility in terms of scenario configuration, sensors, traffic conditions and weather conditions. Pre-created scenarios and assets are included, and custom elements and new scenarios can also be defined. The CARLA ecosystem offers the integration with other products, such as ROS[1], OpenDrive[2] or even some licensed platforms such as Synkrotron.

CARLA simulator has been used to create several ready-to-use synthetic scenarios. Two interesting projects have been published under the NPM3D Benchmark Suite (which also includes the aforementioned public dataset Paris-Lille-3D). Paris-CARLA-3D [26] is a pair of datasets, one captured from the real world in Paris and the other simulated with CARLA using the same sensors configuration. Both datasets are tagged alike so that transfer methods between real and synthetic data can be tested. KITTI-CARLA [27] is a synthetic dataset created in CARLA using the same sensors configuration as in KITTI. This dataset addresses the transfer learning methods from synthetic data to the real world KITTI dataset. SimKITTI32 [28] is a synthetic dataset created by simulating a Velodyne HDL32 in a scenario modeled from SemanticKITTI. This is useful for developing transfer learning method from a Velodyne HDL64 (the sensor from the original KITTI dataset) and the virtualized Velodyne HDL32.

C. Deep Learning Semantic Segmentation Models

The advancements in 3D point cloud semantic segmentation started in 2017, when the pioneering work PointNet++ [29] was introduced. PointNet++ is a point-based method that uses a multi-scale MLP (Multi Layer Perceptron) backbone to capture the local features at different contextual scales. Two years later, RandLA-Net [30] was proposed as an improved, lightweight, faster, more efficient and more robust method. By using a random point selector, larger point clouds can be computed in a faster manner. This is an advantage with respect to PointNet++, which can only operate with small point clouds. Another influential Deep Learning-based semantic segmentation model is KPConv [31], which presents the idea of a kernel point convolution to operate directly on the point cloud without intermediate representations. These models are probably the most influential in terms of number of citations; in fact, various models recently developed still use the original PointNet++ backbone or modified versions.

In recent years, the Deep Learning-based semantic segmentation task have evolved with innovative network architectures and methodologies, which show promising results. SalsaNext [32] is a projection-based method that operates on the range image of the LiDAR point cloud. The 3D point cloud is projected into a 2D image, and thus the processing is way faster as one dimension is ignored. When the segmentation is computed, the projection is reverted and the classification results are transferred from the 2D image to the 3D point cloud. Cylinder3D [33] is a voxel-based method that relies on a cylindrical 3D representation of the point cloud, along with a sparse convolution module to obtain State-of-the-Art results. PVKD [34] is a lightweight model based on Cylinder3D that leverages the knowledge distillation from voxel scale to point level to achieve accurate results in an efficient manner.

III. Methodology

We propose a method to combine real and synthetic data to create a training dataset for a specific semantic segmentation task, shown in Fig. 1. Given a specific real-world dataset, a virtualized model of its MMS is used to generate synthetic data. The generated data (point clouds captured by LiDAR sensors) is a virtualized replica of the real dataset, and therefore it is possible to use both synthetic and real data for training purposes.

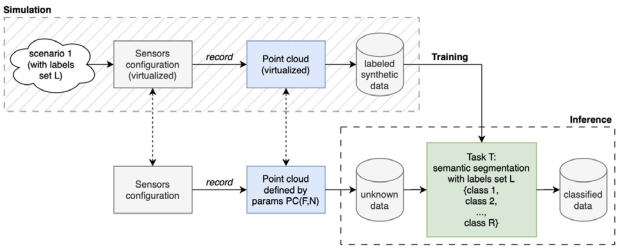

Fig. 1. Proposal that combines synthetic with real data for semantic segmentation.

Attending the capture and processing time, point clouds can be divided into single scans and historic or accumulated point clouds. Single scans are captured one at a time by the sensor. For rotating LiDAR sensors, a scan is the point cloud captured by a single 360º turn of the LiDAR. For solid state LiDARs, a scan is a single shot, like the photograph of a camera. This type of point clouds is the input for real-time or near real-time processing tasks, such as live semantic segmentation or object detection to avoid collisions in autonomous driving. The other mentioned point cloud type is the historic or accumulated point cloud. Generally, mobile mapping systems usually have a positioning unit, such as a

[1] ROS (Robot Operating System) is an open-source framework widely used in autonomous driving applications.

[2] ASAM OpenDrive is a description format that contains all static objects of a road network that allow realistic simulation of vehicles driving on roads.

GNSS/IMU, to match the 3D model captured with the LiDAR to a geographical location. In non-real time applications (such is the case of HDMaps creation), the geographical information can be leveraged to create a larger point cloud by accumulating the scans captured along a certain area. These point clouds contain more information than a single scan and are generally denser (this is, they have a greater number of points per volume unit), which helps better represent the overall shape of the elements.

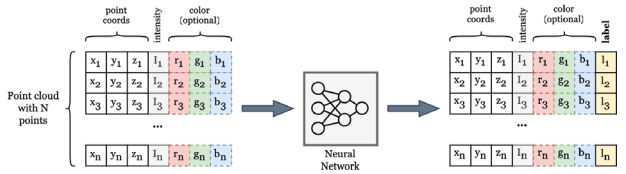

Fig. 2. Representation of the typical point fields in LiDAR point clouds. Note the additional *label* field after semantic segmentation, added to each point. This label represents the class of the object the laser beam impacted with.

Regardless of the point cloud type (scan or historical), several common parameters define the input features of the point cloud (Fig. 2). All 3D points have *x*, *y* and *z* values that define their position in the Euclidean space, but additional values can be included in the point structure. Many LiDARs can provide the information of the reflectivity of the objects, normally referred to as "intensity". Objects with little or no reflectivity (such as the asphalt or other dark surfaces) have low intensity values, whereas very reflective elements (traffic signs, road lines and other highly brilliant surfaces) have high intensity values. Another value that is sometimes present in point clouds are the points' normals. These can be obtained via post-processing. If the point cloud is fused with images captured with color cameras, also the RGB fields can be added to the point structure. In Figure 2 the different fields of a point cloud are shown, both before and after semantic segmentation.

A. Notation

Let us define the labels set as:

$$L = \{l_1, l_2, ..., l_R\} \tag{1}$$

Being *R* the number of labels to be detected. These labels could represent any element class in the scene (road, traffic sign, car, pedestrian...). For the most typical scenario where points of the cloud have Euclidean coordinates plus the intensity (these are the fields that the majority of LiDARs return), a point can be defined as:

$$\vec{p_j} = \{x_j, y_j, z_j, I_j\} \tag{2}$$

A point with its label is composed by the point plus its label:

$$\vec{p_{lj}} = \{\vec{p_j}, l_j\}\ ;\ l_j \in L\ \forall j \tag{3}$$

Let us also define the point cloud *PC* and a labeled point cloud *LPC* as sets of *N* points as follows:

$$PC = \{\vec{p_1}, \ldots, \vec{p_N}\} \tag{4}$$

$$LPC = \{\vec{p_{l1}}, \ldots, \vec{p_{lN}}\} \tag{5}$$

Therefore, the task *T* can be defined as a bijective function that maps the unlabeled point cloud to the labeled one:

$$T : PC \mapsto LPC \tag{6}$$

As mentioned above, additional fields, such as RGB color, can be added to the definition of the point, so we could generalize the previous point definition for any given number of fields. Let us define the point in a generalized manner like:

$$\vec{p_j}(M) = \{f_{1j}, f_{2j}..., f_{Mj}\} \tag{7}$$

Being *M* the number of fields. The point cloud *PC* is, therefore, a bidimensional array defined by its number of points *N* and the number of fields of its points *M*:

$$PC(N, M) = \{\vec{p_1}(M), \ldots, \vec{p_N}(M)\} \tag{8}$$

And so the labeled point cloud:

$$LPC(N, M) = \{\vec{p_{l1}}(M), \ldots, \vec{p_{lN}}(M)\} \tag{9}$$

The resulting labeled point cloud is obtained by applying the task function to the unlabeled point cloud.

$$LPC = T(PC(N, M), L) \tag{10}$$

The task function takes the unlabeled point cloud (parameterized by its number of points *N* and point fields *M*) and the labels set *L*, and assigns a label from the set to each point in the cloud. In DL/ML applications, this process is called inference.

B. Selection of Public Datasets

SemanticKITTI [2] is the preferred dataset considering its focus on point-wise labeled data, essential for semantic segmentation tasks. It employs terrestrial mapping systems, rather than aerial-based datasets such as SensatUrban, which are suited for road scenarios. Its large number of labeled classes also makes it valuable for HDMaps, where more detected classes improve mapping quality.

SemanticKITTI consists of 11 labeled sequences with 23,201 scans collected in Karlsruhe (Germany) using a LiDAR Velodyne HDL64, covering 28 classes. An additional 11 test sequences, though not publicly accessible, can be evaluated using SemanticKITTI's tools.

C. Simulation Environment

On the simulation part, open-source solutions offer several key advantages over licensed products. They often have a quite wide and active community of users, who can also contribute to the development of specific

features and software maintenance. Because they are open source, these solutions generally use different public format files, protocols and standards. This is useful to combine different products, contrarily to the rigidness of licensed solutions. Lastly, open-source solutions offer their source code in public repositories so that anyone can contribute to their development.

To reach a conclusion about which specific open-source solution to use, the characteristics of the options mentioned in Section II.B must be taken into account. CARLA is a widely open application that features all functionality needed for road scenarios simulation. It also has a wide active community, and is integrated with other products. It is flexible and scalable, which makes it a very suitable solution for research and development purposes. LGSVL can be used by forking its repository, but the project is no longer maintained since January 2022, and therefore it is used much less than CARLA, and its developers' community is less numerous. Moreover, CARLA's original paper [25] is much more cited than LGSVL's [35] (3,527 vs 250 total citations).

D. Public Synthetic Datasets

As it was previously mentioned, we need a virtualization of the MMS used to capture the SemanticKITTI dataset to generate the synthetic data inside CARLA. This task was performed in [27], where seven labeled sequences (named towns) of 5,000 scans each, were obtained with a replica of the SemanticKITTI's MMS. This synthetic dataset, therefore, can be used as the simulated data for this study.

We adapted this synthetic data to the format of SemanticKITTI. As a result, the SemanticKITTI dataset is augmented with the seven towns of [27], each one corresponding to a new sequence. The augmented SemanticKITTI dataset is composed as depicted in Fig. 3.

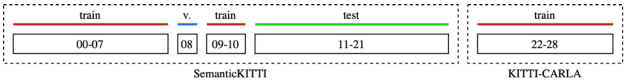

Fig. 3. Composition of original and synthetic sequences in augmented SemanticKITTI dataset for this use case. Sequences 00 to 21 correspond to SemanticKITTI and 22 to 28 to KITTI-CARLA. Sequences 00 to 07, 09, 10 and 22 to 28 form the training split, sequences 11 to 21 form the test split, and sequence 08 is used for validation.

E. Semantic Segmentation Model

Regarding the semantic segmentation model, we decided to use PVKD [34] for evaluation purposes, since its performance results (shown in Table II) obtains good values.

TABLE II. ACCURACIES OF CYLINDER3D AND PVKD ON SEMANTICKITTI

Method	mIoU	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign
Cylinder3D	68.9	97.1	67.6	63.8	50.8	58.5	73.7	69.2	48.0	92.2	65.0	77.0	32.3	90.7	66.5	85.6	72.5	69.8	62.4	66.2
PVKD	71.2	97.0	67.9	69.3	53.5	60.2	75.1	73.5	50.5	91.8	70.9	77.5	41.0	92.4	69.4	86.5	73.8	71.9	64.9	65.8

F. Architecture

Finally, the proposed architecture is detailed in Fig. 4, given more information that the one depicted in Fig. 1. We used the KITTI-CARLA dataset for the synthetic branch, whereas we use SemanticKITTI for the real data branch. The training set of SemanticKITTI is augmented with the converted KITTI-CARLA synthetic dataset, and the test split of SemanticKITTI is used for inference and evaluation.

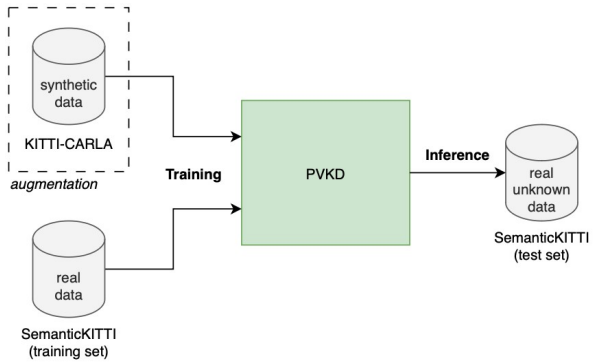

Fig. 4. System architecture with SemanticKITTI as the real dataset and KITTI-CARLA for augmentation.

IV. VALIDATION AND RESULTS

In order to test the viability of use of the synthetic dataset provided by [27], we conducted several experiments with different combinations of synthetic and real data sequences for the training split for the semantic segmentation pipeline. More specifically, we have conducted five experiments: one with 100% of real data and no synthetic data, and the other four with the 100% of synthetic data plus different amounts of real data. The cases with different amounts of synthetic data have not been considered, since the goal of this work is reducing the amount of real data used for training in favor of the synthetic data. The sequences selection is made under the criteria of the most balanced distribution of points between classes. Since KITTI-CARLA dataset does not have an intensity field, a statistical intensity model is introduced to infer the intensity values from the real data. This model is extensively explained in Section IV.C.

In order to avoid overfitting, the labeled SemanticKITTI sequences (00 to 10) are split like this: (i) sequences 00, 06, 07, 09 and 10 are used for training, (ii) sequence 08 is used for validation, and (iii)) sequences 01, 02, 03, 04 and 05 are used for testing. Since only five sequences are used for training the real part, the expected

outcome would be less accurate that the benchmark results shown in Table II:

- **Experiment 1:** 100% real data. In this experiment all the training sequences of SemanticKITTI are used as the training set (sequences 00, 06, 07, 09 and 10). No synthetic data is used.
- **Experiment 2:** 100% synthetic data. No SemanticKITTI sequences are used for training, only synthetic data (all the towns in KITTI-CARLA converted to SemanticKITTI format).
- **Experiment 3:** All Towns in KITTI-CARLA are used and the 25% of the SemanticKITTI training sequences are used for training (sequences 06 and 10).
- **Experiment 4:** All Towns in KITTI-CARLA are used and the 50% of the SemanticKITTI training sequences are used for training (sequences 06, 07, 09 and 10).
- **Experiment 5:** All Towns in KITTI-CARLA are used and the 75% of the SemanticKITTI training sequences are used for training (sequences 00, 06 and 09).

All these five experiments are executed twice: once with a fixed intensity value of 0.5 (SemanticKITTI's intensity is ranged between 0 and 1) and a second time with the statistical intensity model. The improvement in accuracy of this second set of experiments is discussed in Section IV.D.

A. Metrics and Label Matching

The standard metric used as performance indicator for this work's results is the Intersection over Union (IoU), also called Jaccard index. Let TP_c be the number of true positives, FP_c the number of false positives, FN_c the false negative predictions for each class c, and C the total number of classes. Then:

$$IoU_c = \frac{TP_c}{TP_c + FP_c + FN_c} \tag{11}$$

Therefore, the mean intersection over union (mIoU) is calculated as the mean of all the classes:

$$mIoU = \frac{1}{C}\sum_{c=1}^{C}\frac{TP_c}{TP_c + FP_c + FN_c} \tag{12}$$

The two datasets (real and synthetic) used for these experiments have different labels sets. Let L_r be the labels set of the real dataset and Ls the labels set of the synthetic data. In order to get to use a common labels set (let us call it L) both L_r and Ls must be mapped to match the final labels set L. Then, two mapping functions are needed, one for each dataset. Let MF_r the mapping function for the SemanticKITTI dataset and MF_s the mapping function for the synthetic dataset.

$$MF_r : L_r \mapsto L \tag{13}$$

$$MF_s : L_s \mapsto L \tag{14}$$

The mapping of both L_r and L_s into L is implemented with the criteria of the greatest geometrical similarity between classes. For example, SemanticKITTI's "car", "bicycle", "motorcycle", "truck" and "other-vehicle" map to a unified label "vehicle" of the set L. Similarly, the label "vehicles" of L_r maps to "vehicle" label of L. This process is implemented for all labels in both L_r and L_s, and the final set L has a total of 11 classes, namely: "vehicle", "person", "road", "sidewalk", "other-ground", "building", "fence", "vegetation", "terrain", "pole" and "traffic-sign".

B. Inferring Intensity by a Statistical Distribution Approach

As previously stated in Section IV.B, KITTI-CARLA dataset does not have an intensity field. This is, in fact, a great limitation to the overall performance of the system since this field provides significant information for many classes. To overcome this issue, a statistical distribution of intensities is computed per class, and then, in the conversion process, a random normal distribution of intensities is assigned to each class. In Table III and Fig. 5 the mean and standard deviation of intensity for each class in SemanticKITTI is shown.

To calculate these parameters, the intensity of each point is extracted into a per-class histogram. Although the intensity values are discrete, given the large amount of points in the whole dataset, these histograms can be approximated to a continuous distribution. The normal distribution was selected to represent the data because it best represents the intensity variations for each class, which correspond to variations in reflectivity in real-world objects due to heterogeneous surfaces, different colors, and noise.

TABLE III. MEAN AND STANDARD DEVIATION (INTENSITY) OF EACH CLASS IN SEMANTICKITTI

Class	unlabeled	vehicle	person	road	sidewalk	other-ground	building	fence	vegetation	terrain	pole	traffic-sign
Mean	0.148	0.326	0.246	0.303	0.302	0.361	0.295	0.296	0.299	0.342	0.222	0.499
Standard deviation	0.195	0.183	0.104	0.087	0.096	0.163	0.141	0.142	0.161	0.1	0.19	0.367

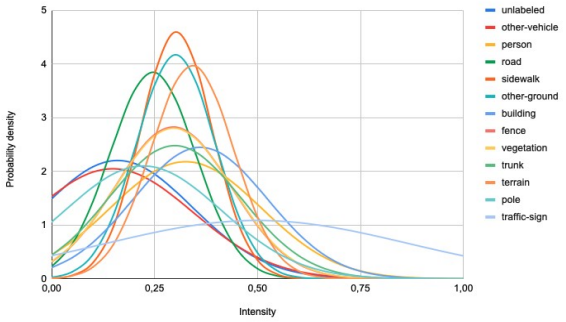

Fig. 5. Graphical representation of the normal distributions of intensities per class.

C. Results

Table IV shows the results of the experiments and Fig. 6 depicts a comparison of the obtained mIoU for each experiment with fixed intensity (0.5) vs the inferred intensity (by using a statistical distribution approach). According to the obtained results, the performance of the system is clearly sensible to the intensity distribution.

The experimental results show a significant improvement when the intensity is computed using the statistical distribution module with respect to when a fixed value is used. Also, no significant improvement is shown with percentages above 50% of the original training set for both the fixed and statistical intensity methods. This is because at this percentage, the limit of segmentation capabilities of this model with combined synthetic and real data is reached. Some classes vary significantly more than others in segmentation accuracy loss, specifically "person" and "other ground". This effect is a consequence of the variability on what these classes represent in the real world, which is very dependent on the scenario. The class "person" includes all types of pedestrians, cyclist and motorcyclists in many different poses, rendering a class that is hard to generalize. Also, "other ground" represents a highly variable class of surfaces that cannot be classified as roads or sidewalks, such as parking slots, non-drivable areas or emergency lanes, which can have very different geometries and reflectivities depending on the scenario.

A value worth noting is the IoU of the vehicle class, obtained using the 50% of real data with the statistic distribution of intensities. This value is higher than that obtained with 100% real data, and so the segmentation is more accurate for this class when using the synthetic data for augmentation.

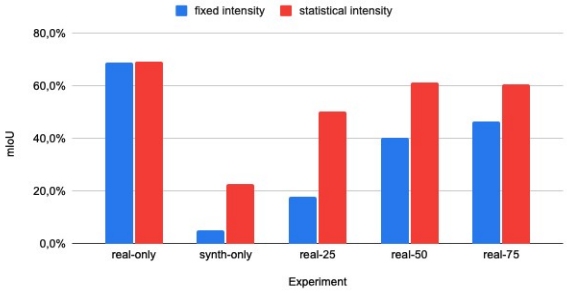

Fig. 6. Comparison of mIoU for each experiment with fixed intensity (0.5) vs statistical intensity.

D. Comparative Study

To generalize the usage of our method, we performed the same set of experiments using a different semantic segmentation model: Cylinder3D [33]. In Table V, the results of the experiments are shown. These experiments have been performed using the statistical intensity model. In Fig. 7, the mIoU of each experiment is presented.

TABLE IV. QUANTITATIVE RESULTS OF CONDUCTED EXPERIMENTS ON THE SEMANTICKITTI DATASET

	Training set	mIoU (%)	vehicle	person	road	sidewalk	other-ground	building	fence	vegetation	terrain	pole	traffic-sign
Fixed intensity (0.5)	100% real, 0% synthetic	69.0	73.2	66.4	93.3	81.3	44.2	85.5	60.4	82.4	58.2	54.9	59.4
	0% real, 100% synthetic	5.2	0.0	0.1	0.1	3.0	0.1	16.0	0.0	32.7	0.1	4.9	0.0
	25% real, 100% synthetic	17.7	5.9	3.5	43.7	37.0	0.0	42.1	11.4	5.9	5.4	11.3	28.8
	50% real, 100% synthetic	40.3	21.5	4.3	**78.3**	**54.6**	**8.1**	62.0	41.1	56.4	**35.2**	34.9	46.4
	75% real, 100% synthetic	**46.3**	**39.0**	**27.2**	67.6	42.1	1.3	**74.3**	**47.0**	**76.5**	29.0	**49.7**	**56.0**
Statistical intensity	100% real, 0% synthetic	69.2	77.7	67.8	92.5	80.1	42.3	86.0	62.1	82.5	57.6	57.2	55.4
	0% real, 100% synthetic	22.7	32.5	1.7	27.8	35.3	0.1	33.4	15.6	57.5	32.4	7.5	5.6
	25% real, 100% synthetic	50.4	79.4	22.7	79.0	57.5	0.5	76.5	42.6	75.6	37.7	39.8	43.3
	50% real, 100% synthetic	**61.1**	**80.9**	**49.7**	**89.5**	**74.2**	**16.7**	**78.9**	**56.8**	**79.2**	50.8	49.3	46.4
	75% real, 100% synthetic	60.4	80.5	47.5	88.9	74.1	7.6	78.6	53.8	77.5	**51.8**	**52.1**	**51**.9

TABLE V. PVKD VS CYLINDER3D WITH STATISTICAL INTENSITY MODEL

	Training set	mIoU (%)	vehicle	person	road	sidewalk	other-ground	building	fence	vegetation	terrain	pole	traffic-sign
Cylinder3D	100% real, 0% synthetic	67.5	87.2	61.0	89.0	77.0	43.8	76.9	59.3	80.4	57.6	54.6	55.4
	0% real, 100% synthetic	20.4	18.6	4.0	45.7	25.4	0.7	32.2	20.1	33.0	24.4	7.8	12.5
	25% real, 100% synthetic	52.3	70.4	25.5	78.1	61.1	4.5	70.7	53.6	74.9	44.6	48.0	43.9
	50% real, 100% synthetic	**60.0**	**86.1**	**47.0**	84.4	69.4	18.1	**72.6**	**55.3**	76.2	48.4	**51.4**	51.4
	75% real, 100% synthetic	59.8	81.9	37.6	**87.8**	**72.4**	**32.8**	70.0	51.5	**77.2**	**49.4**	45.4	**52.1**
PVKD	100% real, 0% synthetic	69.2	77.7	67.8	92.5	80.1	42.3	86.0	62.1	82.5	57.6	57.2	55.4
	0% real, 100% synthetic	22.7	32.5	1.7	27.8	35.3	0.1	33.4	15.6	57.5	32.4	7.5	5.6
	25% real, 100% synthetic	50.4	79.4	22.7	79.0	57.5	0.5	76.5	42.6	75.6	37.7	39.8	43.3
	50% real, 100% synthetic	**61.1**	**80.9**	**49.7**	**89.5**	**74.2**	**16.7**	**78.9**	**56.8**	**79.2**	50.8	49.3	46.4
	75% real, 100% synthetic	60.4	80.5	47.5	88.9	74.1	7.6	78.6	53.8	77.5	**51.8**	**52.1**	**51**.9

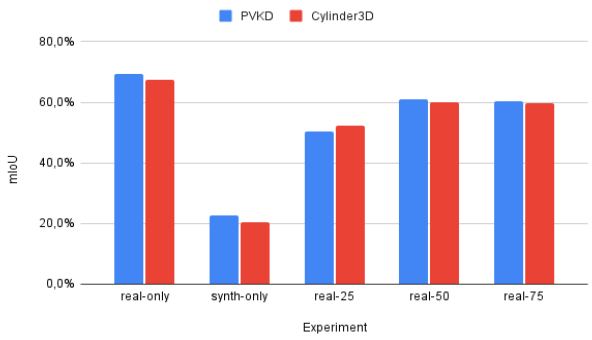

Fig. 7. Comparison of mIoU for each experiment with PVKD and Cylinder3D (statistical intensity model).

The results show that both models behave in similar ways, with PVKD slightly outperforming Cylinder3D. The comparison between both semantic segmentation models shows the generalization capabilities of the statistical intensity model.

E. Limitations and Challenges

So far, we have commented the viability of the KITTI-CARLA dataset to SemanticKITTI for data augmentation, but its applicability to other datasets and scenarios can be challenging. First, the labels of both real and synthetic datasets must be, if not fully coincident, at least matchable. In this case, the mapping functions defined in Eqs. (13) and (14) should be reimplemented according to the labels sets of both datasets.

Another difficulty arises from the diverse sensors used to capture different datasets, which have different parameters like range or field of view, making the applicability of synthetic data not straightforward. To target this sensor domain generalization problem, a dataset similar to KITTI-CARLA has been published [28], which simulates a Velodyne HDL-32 inside a SemanticKITTI scene. Even in the case of datasets captured using the same model of the LiDAR sensor, with the same range and field of view, still some variations in intensity may occur, due to the differences in intensity calibration parameters in the manufacturing stage. In this case, the sensor domain generalization can be addressed via an equalization process for intensity distributions between sensors.

In practical applications, the variability of sensors and configurations may be even bigger. Moreover, LiDAR manufacturers introduce new models to the market continuously as technologies advance, rendering the generalization problem more challenging with time. This impacts the viability of using synthetic data for augmentation right away.

However, our statistical intensity modelling method is still applicable to a wide range of synthetic datasets, as far as they are labeled, since it effectively replicates the intensity distributions from real datasets to synthetic ones. The main challenges in real-world applications are: 1) getting large amounts of real data labeled, and 2) getting a customized synthetic dataset that accurately represents a virtual geometric replica of the real-world data.

V. Conclusions

We have introduced a methodology to implement augmentation on SemanticKITTI using the synthetic dataset KITTI-CARLA. For this to be possible, we have proposed a mechanism to map the different labels in both datasets. Besides, and since PVKD is proved to be sensible to the intensity values on the scene, we propose a novel statistic distribution module to infer the real intensities of the different classes in SemanticKITTI. Our experimental results show that the synthetic dataset KITTI-CARLA can be effectively used to significantly reduce the amount of real data needed to achieve reliable results.

The key feature of the proposed method is its simple yet effective implementation. Other augmentation experiments based on synthetic datasets are forced to either ignore the intensity information [36, 37] or generate it using complex and computationally expensive methods [38]. Our method provides a very efficient way to generate the intensity field in a significant manner that effectively contributes to improve the segmentation results.

The main weakness of the simulation-based data generation approach is the breach in realism between synthetic and real-world data. Noise, reflections, light scattering or return losses are examples of artifacts that are hard to replicate in a simulated scenario. As a future development, research in realistic LiDAR simulation may help reduce the leap between synthetic and real data, and thus reducing even more the need for real data in favor of synthetic data. Alternatively, recent developments in Latent Diffusion Models (LDM) applied to LiDAR data show promising results when generating synthetic data in both conditioned and unconditioned ways [39, 40]. This approach optimizes even more the generation process by eliminating the need for a simulation environment, along with the workload of creating scenarios and running simulations on it. Also, LDM enable the introduction of the mentioned artifacts by learning them from the real-world data, rendering the generated data more realistic. Moreover, LDM permit the implementation of other tasks related to conditioned generation, such as “image to LiDAR” or “text to LiDAR”. Although further research is still to come, LDM present a promising and highly flexible alternative for LiDAR synthetic data generation at a low cost.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

R.D. conducted the research; P.A. conducted the experiments; P.A. implemented the code; R.D. and P.A. wrote the paper; all authors had approved the final version.

Funding

This work was supported by the Automotive Technology Centre of Galicia (CTAG), by the grant

PID2020113795RB-C33 funded by MICIU/AEI/10.13039/501100011033 (COMPROMISE project) and the grant PID2023-148716OB-C31 funded by MCIU/AEI/10.13039/501100011033 (DISCOVERY project). Additionally, this work has also received financial support by the Galician Regional Government under project ED431B 2024/41 (GPC).

ACKNOWLEDGMENT

We would like to thank the Centro de Supercomputación de Galicia (CESGA) for their support and the resources for this research.

REFERENCES

[1] B. Gao, Y. Pan, C. Li, S. Geng, and H. Zhao, "Are we hungry for 3D LiDAR data for semantic segmentation? A survey of datasets and methods," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 6063–6081, 2021.

[2] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019.

[3] H. Caesar *et al.*, "nuScenes: A multimodal dataset for autonomous driving," in *Proc. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11618–11628.

[4] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3354–3361.

[5] P. Sun, H. Kretzschmar, X. Dotiwalla *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," arXiv preprint, arXiv:1912.04838, 2020.

[6] W. Tan, N. Qin, L Ma, Y. Li, J. Du, G. Cai, K. Yang, and J. Li, "Toronto-3D: A large-scale mobile LiDAR dataset for semantic segmentation of urban roadways," in *Proc. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020.

[7] X. Roynard, J.-E. Deschaud, and F. Goulette, "Paris-lille-3D: A large and high-quality ground truth urban point cloud dataset for automatic segmentation and classification," *The International Journal of Robotics Research*, vol. 37, no. 6, pp. 545–557, 2018.

[8] Q. Hu, B. Yang, S. Khalid, W. Xiao, N. Trigoni, and A. Markham, "Towards semantic segmentation of urban-scale 3D point clouds: A dataset, benchmarks and challenges," in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4977–4987.

[9] Y. Pan, B. Gao, J. Mei, S. Geng, C. Li, and H. Zhao, "Semanticposs: A point cloud dataset with large quantity of dynamic instances," arXiv preprint, arXiv:2002.09147, 2020.

[10] J. Geyer, Y. Kassahun, M. Mahmudi *et al.*, "A2d2: Audi autonomous driving dataset," arXiv preprint, arXiv:2004.06320, 2020.

[11] B. Wilson, W. Qi, T. Agarwal *et al.*, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," in *Proc. the Neural Information Processing Systems Track on Datasets and Benchmarks*, Vanschoren, J. and Yeung, S., eds. 2021.

[12] J. Houston, G. Zuidhof, L. Bergamini, Y. Ye, L. Chen, A. Jain, S. Omari, V. Iglovikov, and P. Ondruska, "One thousand and one hours: Self-driving motion prediction dataset," in *Proc. Conference on Robot Learning*, 2020.

[13] G. Tong, Y. Li, D. Chen, Q. Sun, W. Cao, and G. Xiang, "Cspc-dataset: New lidar point cloud dataset and benchmark for large-scale scene semantic segmentation," *IEEE Access*, vol. 8, pp. 87695–87718, 2020.

[14] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys, "SEMANTIC3D.NET: A new large-scale point cloud classification benchmark," in *Proc. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2017, pp. 91–98.

[15] S. M. I. Zolanvari, S. Ruano, A Rana, A. Cummins, R. E. da Silva, M. Rahbar, and A. Smolic, "Dublincity: Annotated lidar point cloud and its applications," arXiv preprint, arXiv:1909.03613, 2019.

[16] N. Varney, V. K. Asari, and Q. Graehling, "Dales: A large-scale aerial lidar data set for semantic segmentation," in *Proc. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 717–726.

[17] W. Gao, L. Nan, B. Boom, and H. Ledoux, "SUM: A benchmark dataset of semantic urban meshes," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 179, pp. 108–120, Sep. 2021.

[18] G. Can, D. Mantegazza, G. Abbate, S. Chappuis, and A. Giusti, "Semantic segmentation on swiss3dcities: A benchmark study on aerial photogrammetric 3d pointcloud dataset," *Pattern Recognit. Lett.*, vol. 150, pp. 108–114, 2020.

[19] L. Lin, Y. Liu, Y. Hu, X. Yan, K. Xie, and H. Huang, "Capturing, reconstructing, and simulating: the urbanscene3d dataset," in *Proc. ECCV*, 2022, pp. 93–109.

[20] IPG Automotive GmbH. Carmaker. [Online]. Available: https://ipg-automotive.com/products-services/simulation-software/carmaker/

[21] The MathWorks, Inc. Roadrunner. [Online]. Available: https://www.mathworks.com/products/roadrunner.html

[22] Anyverse. (2024). [Online]. Available: https://anyverse.ai

[23] Synkrotron. [Online]. Available: https://www.synkrotron.ai

[24] National Instruments. Monodrive. [Online]. Available: https://monodrive.readthedocs.io/en/latest/

[25] A. Dosovitskiy, G. Ros, F. Codevilla, A. M. L´opez, and V. Koltun, "Carla: An open urban driving simulator," in *Proc. Conference on Robot Learning*, 2017.

[26] J.-E. Deschaud, D. Duque, J. P. Richa, S. Velasco-Forero, B. Marcotegui, and F. Goulette, "Pariscarla-3D: A real and synthetic outdoor point cloud dataset for challenging tasks in 3D mapping," *Remote Sensing*, vol. 13, no. 22, 2021.

[27] J.-E. Deschaud, "KITTI-CARLA: A KITTI-like dataset generated by CARLA simulator," arXiv preprint, arXiv:2109.00892, 2021.

[28] J. P. Richa, J.-E. Deschaud, F. Goulette, and N. Dalmasso, "Adasplats: Adaptive splatting of point clouds for accurate 3D modeling and real-time high-fidelity lidar simulation," *Remote Sensing*, vol. 14, no. 24, 2022.

[29] C. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in Neural Information Processing Systems*, 30, 2017.

[30] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z Wang, A. Trigoni, and A. Markham, "Randla-net: Efficient semantic segmentation of large-scale point clouds," in *Proc. 2020 IEEE/CVF Conference on Computer* Vision *and Pattern Recognition (CVPR)*, 2019, pp. 11105–11114.

[31] H. Thomas, C. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proc. 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6410–6419.

[32] T. Cortinhal, G Tzelepis, and E. E. Aksoy, "Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds," in *Proc. International Symposium on Visual Computing*, 2020.

[33] H. Zhou, X. Zhu, X. Song, Y. Ma, Z. Wang, H. Li, and D. Lin, "Cylinder3D: An effective 3d framework for driving-scene lidar semantic segmentation," arXiv preprint, arXiv:2008.01550, 2020.

[34] Y. Hou, X. Zhu, Y. Ma, C. C. Loy, and Y. Li, "Point-to-voxel knowledge distillation for lidar semantic segmentation," are semantic segmentation of lidar point clouds," in *Proc. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 8469–8478.

[35] G. Rong, B. H. Shin, H. Tabatabaee *et al.*, "LGSVL simulator: A high fidelity simulator for autonomous driving," in *Proc. 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6.

[36] Z. Qian *et al*., "Synthcity: A benchmark framework for diverse use cases of tabular synthetic data," *Neural Information Processing Systems*, 2023.

[37] B. Hurl *et al*., "Precise synthetic image and LiDAR (PreSIL) dataset for autonomous vehicle perception," in *Proc. 2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 2522–2529.

[38] A. Xiao *et al*., "SynLiDAR: Learning from synthetic LiDAR sequential point cloud for semantic segmentation," arXiv preprint, arXiv:2107.05399, 2021.
[39] H. Ran *et al*., "Towards realistic scene generation with LiDAR diffusion models," in *Proc. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 14738–14748.
[40] V. Zyrianov, H. Che, Z. Liu, and S. Wang, "Lidardm: Generative lidar simulation in a generated world," arXiv preprint, arXiv:2404.02903, 2024.