computer sciences and mathematics forum

MDPI

Proceeding Paper

Exploring 3D Object Detection for Autonomous Factory Driving: Advanced Research on Handling Limited Annotations with Ground Truth Sampling Augmentation †

Matthias Reuse [1,*], Karl Amende [1], Martin Simon [1] and Bernhard Sick [2]

[1] Valeo Schalter & Sensoren GmbH, Hummendorfer Str. 74, 96317 Kronach, Germany; karl.amende@valeo.com (K.A.); martin.simon@valeo.com (M.S.)
[2] Intelligent Embedded Systems, Universität Kassel, Wilhelmshöher Allee 73, 34121 Kassel, Germany; bsick@uni-kassel.de
* Correspondence: matthias.reuse@valeo.com
† Presented at the 2nd AAAI Workshop on Artificial Intelligence with Biased or Scarce Data (AIBSD), Vancouver, BC, Canada, 26 February 2024.

Abstract: Autonomously driving vehicles in car factories and parking spaces can represent a competitive advantage in the logistics industry. However, the real-world application is challenging in many ways. First of all, there are no publicly available datasets for this specific task. Therefore, we equipped two industrial production sites with up to 11 LiDAR sensors to collect and annotate our own data for infrastructural 3D object detection. These form the basis for extensive experiments. Due to the still limited amount of labeled data, the commonly used ground truth sampling augmentation is the core of research in this work. Several variations of this augmentation method are explored, revealing that in our case, the most commonly used is not necessarily the best. We show that an easy-to-create polygon can noticeably improve the detection results in this application scenario. By using these augmentation methods, it is even possible to achieve moderate detection results when only empty frames without any objects and a database with only a few labeled objects are used.

Keywords: 3D object detection; infrastructural LiDAR; data augmentation; autonomous driving

Citation: Reuse, M.; Amende, K.; Simon, M.; Sick, B. Exploring 3D Object Detection for Autonomous Factory Driving: Advanced Research on Handling Limited Annotations with Ground Truth Sampling Augmentation. *Comput. Sci. Math. Forum* **2024**, *9*, 5. https://doi.org/10.3390/cmsf2024009005

Academic Editors: Kuan-Chuan Peng, Abhishek Aich and Ziyan Wu

Published: 18 February 2024

1. Introduction

The vision of autonomous driving is becoming more and more of a reality, not only in individual transport, but also for industrial applications to automate processes, simplify workflows, and increase safety. For example, the transportation of goods through warehouses can be handled by autonomous vehicles. Or even newly built vehicles can drive through parts of production facilities without the help of human drivers. This application of autonomous factory driving will be investigated in this work. In our use case, the vehicles are to drive autonomously at low speed for part of the final production route through the plant premises. The entire system required to solve such a complex task can be divided into several parts. In this paper, we focus on the 3D object detection (3D OD) task, as it is an essential part of this pipeline, since the results of many other functions are based on its output.

Most of the publicly available datasets and benchmarks for 3D OD on LiDAR data are recorded from the point of view of a single ego vehicle [1–6]. Unfortunately, these are unsuitable for the stated problem, as an overall view of the production site that is as occlusion-free as possible is required to maneuver several vehicles safely at the same time. Therefore, an infrastructural LiDAR sensor setup is to be used. Using the ego-vehicle-based data for training would be a big domain shift to this setup. Although there are public datasets with infrastructural sensors [7–10], these are rather limited in availability, size and quality [11,12]. Furthermore, due to the domain shift, it is not possible to use these for

training and direct inference. Therefore, data had to be recorded and annotated accordingly. During the process of creating a suitable dataset, we encountered many sources of delay.

As a result, only a fraction of the pre-validated data could be labeled and prepared. This scarce data basis was addressed by using suitable methods such as heavy data augmentation. The most potent augmentation method for our case is ground truth (GT) sampling [13] as it touches the border to simulation. In GT sampling, a database of objects is created. During training time, objects are randomly selected from this database and inserted into the current frame. This work aims not only to provide a solution for the specific application but to investigate further the influence of GT sampling. Several experiments show its effectiveness even with very little data available. In addition, this work explores the idea that GT sampling can be better utilized in a fixed environment. For this scenario, a more sophisticated GT sampling method utilizing an easy-to-create polygon is proposed to compensate for the lack of data. Unfortunately, for legal reasons, the data can only be shown in parts within this publication and cannot be published in its entirety.

The rest of this paper is structured as follows: First, the related work that is relevant to this task will be reviewed. The specific sensor setup and the resulting dataset, the networks used for the experiments, the GT sampling augmentation variations, and the evaluation metric used will afterwards be discussed. The results of the experiments are then presented. Finally, a conclusion and an outlook on future extensions of the proposed methods will be given.

2. Related Work

In the following, the current state-of-the-art is discussed with regard to the main topics of this paper. Note that some officially unpublished papers on arXiv are also considered for the state-of-the-art research, as they provide additional insights.

2.1. 3D Object Detection with Infrastructural LiDAR Sensors

The current state-of-the-art in terms of 3D OD for autonomous driving is mainly divided by the input format of the LiDAR point cloud as well as the number of stages used for the detection network [14,15]. Most current approaches use the point cloud directly as input [16–19] or convert it into a discrete voxelgrid [20–28]. While one-stage detectors predict the objects directly [15,16,21,23,27,28], two stage detectors first produce proposals, which are refined to the final predictions in the second stage, resulting in better detection accuracy but slower run time [17,19,20,23,25–27]. The current state-of-the-art for 3D OD on infrastructural LiDAR uses only one-stage voxel-based detectors. The most commonly used network is PointPillars [10,21,29–31] or more or less strong extensions of it [32–34]. For example, the authors of [33] propose an extension of PointPillars with attention, a multi-task head, and other minor additions. Otherwise, Ref. [27] uses CenterPoint [12] and ref. [35] uses VoxelNet [28] for 3D OD. The surveys [36,37] provide an overview of the research field of infrastructural 3D OD.

Unlike ego-vehicle-based detection, where a selection of datasets has become the standard [1–6], for the task of infrastructural detection many different smaller datasets exist. The authors of [35] use CARLA [38] to create a synthetic dataset of a T-junction and a roundabout. The authors conduct experiments related to the number of LiDAR sensors in the simulated setups and the stage of the fusion of the different point clouds. It is shown that early fusion of overlapping sensors is able to increase the detection results. In [30] also mainly simulation data is used to perform 3D OD. It is likewise shown that an early fusion of the point clouds increases the detection results. The authors of [32] also work with simulation data from Carla. They experiment with the placement of different LiDAR sensors in an infrastructure setup for the task of 3D OD and with varying fusion schemes. Their experiments show that a LiDAR setup that leads to higher uniformity and coverage of the objects of interest is beneficial for 3D OD. The authors of [33] perform 3D OD on the IPS300+ [8] and the A9 dataset [7] as well as semi-synthetic data. An early fusion of the point clouds is also performed here. In [34] a follow-up to [33] with the use of

synthetic data is presented. In [12] a semi-automated annotation pipeline for infrastructure LiDAR data is introduced. Their dataset features only one LiDAR sensor, so no fusion is required. The authors claim the release of their dataset named FLORIDA, but at the time of writing, it had not yet been published. The authors of [31] again use the A9 dataset. They utilize the three LiDAR sensors as well as the roadside cameras for 3D OD by fusing the results of conventional methods as well as deep learning approaches. In addition to the already mentioned datasets with real recordings for infrastructural LiDAR object detection IPS300+ [8] and A9 [7], there are other small datasets that are publicly available, such as the Baai-vanjee dataset [9], in which an intersection in Bejing was captured with two LiDAR sensors, or the datasets of intersections in Germany proposed in [39] or [40]. Infrastructural LiDAR data can further be extended with point clouds recorded by vehicles. There are several papers for this extended task [11,41,42], which work with simulated data or the DAIR-V2X dataset [10]. Further works in this direction have been published, but since this deviates from the task of this work, we will leave it at this point with this selection.

2.2. Data Augmentation

There are a variety of different augmentation methods for the task of 3D OD on LiDAR data. In the two papers [43,44] the most common methods for 3D OD were applied and experimented with different parameter sets, networks and datasets. These are simple transformations of the entire point cloud and the GT boxes such as rotation, scaling, and translation. These transformations can also be applied at object level. Here, only the GT boxes and their inner points are transformed. Other augmentation methods such as frustum-based deletion and noise of points [45], shifting different parts of one object [46], or mixup [47] also exist, but are applied much less frequently. Another very common method included in the standard catalog of augmentation methods is GT sampling, which was first introduced in [13]. GT boxes and their inner points are collected in a database, and during training, objects are drawn from this database and inserted into the current point cloud. This GT sampling method was further developed in several ways. The placement of the objects, which were originally inserted at the position where they were cut out, was a key area of research. Thus, the placement of the objects on the previously estimated ground plane has established itself as the standard in the community [44,48–50]. In [51,52], the semantic segmentation of corresponding camera images was used to find semantically meaningful positions, such as the placement of cars on the road and pedestrians on the sidewalk. The authors of [48] introduce a ValidMap for position generation. A grid is created from the number of points within a cell and their height information in relation to the ground. Additionally, the objects are inserted occlusion-aware, so that inserted objects cast a shadow in the point cloud. In [50] an estimation of roads and sidewalks is performed for better placement of objects. Occlusion handling is also used here. The authors of [26] propose a pattern-aware down sampling of objects from the database so that they can be realistically placed further distances away.

3. Methods

The datasets for infrastructural LiDAR 3D OD presented in the previous section do not meet the requirements for the task in this paper. The datasets IPS300+ [8] and Lumpi [40] come closest to the requirements. However, the first could not be download due to region lock, while no labels were available for the second at the time of writing. Furthermore, the public datasets are not released for commercial use. For this reason, own data were recorded at the relevant sites. Following related work, single-stage detectors were used and the point clouds were fused at an early stage. In contrast to previous work in this area, this paper focuses on compensating for the limited data available, mainly using the GT sampling augmentation method and applying it to the specific case of the fixed environment. In the following subsections, the acquired dataset, the object detectors, the augmentation methods and the evaluation metrics for the experiments are explained.

3.1. Sensor Setup & Data

For data collection, we equipped two factory sites with infrastructural LiDAR sensors and cameras so that all regions of interest are clearly visible. A mix of automotive-grade fisheye (1MPix, 190° FoV) and pinhole cameras (1MPix, 120° FoV) cover the near and long range, respectively. At the time of writing, two different facilities are supported. referred to as K and G. Site K has three HESAI XT32 LiDAR sensors, nine fisheye cameras, and four pinhole cameras. The fused point cloud of all LiDAR sensors can be seen in Figure 1. Site G is larger and more complex than site K. Therefore, eleven LiDAR sensors, nineteen fisheye, and fifteen pinhole cameras were required to cover the area.

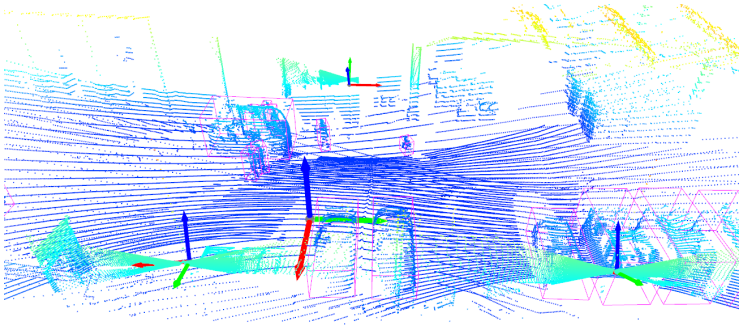

Figure 1. Excerpt of a fused LiDAR point cloud from the infrastructural setup of factory site K recorded with three LiDAR sensors. Labeled ground truth boxes are shown in pink. Point color decodes the height from blue as the lowest to yellow as the highest. LiDAR positions are shown with small coordinate axes each. Big coordinates axes mark the world coordinate origin.

The bounding boxes are labeled on the fused point cloud of all available LiDAR sensors. Spatial registration was done by extrinsic calibration, whereby all point clouds were transferred to a world system. Temporal alignment was done using PTP synchronization using the GPS time for all sensors and then aligning the point clouds based on minimal timestamp difference. The box parameters are the center position in 3D, the dimensions and the heading angle. Although more classes are labeled, cars and pedestrians are most relevant for the task. Therefore, only these two are considered in the following experiments. A total of 175,932 cars and 17,377 pedestrians were labeled. For site K there are 39 sequences and 1411 frames. For site G there are 90 sequences and 3119 frames. For the experiments, these two were considered as one dataset and split into training (60%), validation (20%), and test (20%). For this purpose, all recorded sequences were divided into subsets, whereby care was taken to ensure that the number of objects belonging to the car and pedestrian classes roughly corresponded to the defined ratios. The result was a training set with 95,270 cars and 10,426 pedestrians, 39,743 cars and 3455 pedestrians for the validation, and 40,917 cars and 3494 pedestrians for the test split.

3.2. 3D Object Detectors

For our experiments, two different 3D object detectors were chosen according to their usability for the described task and the aforementioned state-of-the-art. The implementations of both networks are based on the OpenPCDet framework [53]. Consequently, most of the hyperparameters are taken from the configuration files provided by OpenPCDet, unless otherwise stated. The two networks are briefly presented below.

PointPillars is a lightweight one-stage detector [21]. The input point cloud is converted into a voxelgrid, where the voxels have an infinite height and thus form the namesake pillars. A feature vector is calculated for each pillar, and the resulting feature pseudo image is further processed by 2D convolutions. An anchor-based detection head generates the final box predictions. Since PointPillars is an older model, an updated version of [54] is used as it has high performance with comparatively low memory consumption and low inference time, which is crucial for a real-time application.

CenterPoint is another fast voxelization approach [27]. The input point cloud is first converted to a voxelgrid, and again a pseudo-image feature map is created and further processed by 2D convolutions. Unlike PointPillars, the predictions are not made via an anchor head, but are based only on the prediction of object centers. The authors of [27] also propose a second stage extension for CenterPoint, which refines the box predictions. The one-stage variation is used for the experiments.

3.3. Ground Truth Sampling

GT sampling is one of the most commonly used augmentation methods for 3D OD on LiDAR data. Objects from the training data are gathered in a database and inserted into the current LiDAR frame during training time. Usually the insertion is done at the same position and orientation as the original GT object. A common addition is to adjust the insertion height to the previously estimated ground plane of the current frame to prevent objects from floating above or below the ground. Before insertion, a collision check is carried out based on the bounding boxes to prevent possible overlaps with other objects. However, this can still lead to unrealistic placement of objects within unlabeled point clusters such as walls. Furthermore, the original position of an object in the extraction frame is not necessarily within the region of interest in the current frame. To counteract this behavior, Ref. [48] propose a ValidMap based on the number of points and the height of the points in relation to the estimated ground plane. Objects are only inserted at valid positions on the map. Inspired by this approach, we also limit the insertion area. Unlike for ego-vehicle-based data, the environment for our task is fixed. This makes it possible to determine the regions in which objects are to be placed beforehand. For both sites K and G, a polygon is drawn around the areas where objects should be inserted. The polygons can be seen in Figure 2.

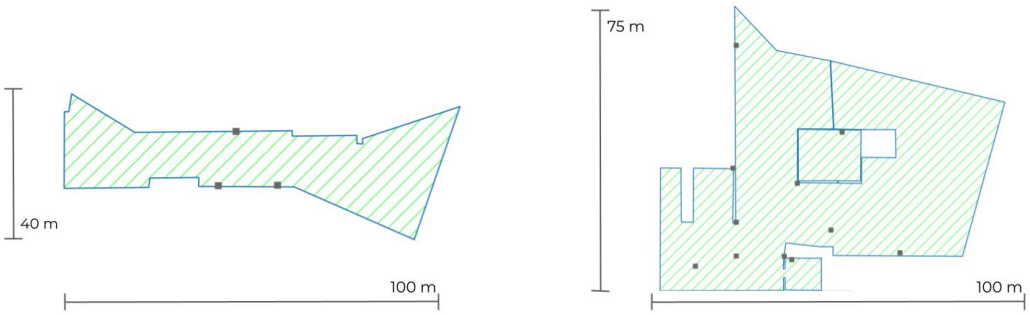

Figure 2. The two polygons drawn to restrict the area into which the objects are inserted by the GT sampling method are shown. **Left**, the polygon for site K is depicted, **right** for site G. Hatched area represents the valid space. LiDAR sensors are depicted as black squares. The axes indicated the scale in meter.

Various methods for inserting the objects are also examined. A sketch for these variations can be seen in Figure 3.

This means that not only the position and orientation of the original GT object is used. A random selection of position and orientation from a uniform distribution is also experimented with. In addition, a polar coordinate-based placement is considered, where the polar distance of the object is kept within a perimeter of two meters around the original distance. The relative angle to the world origin is used for orientation. Based on the selected position, the orientation is calculated in such a way that the relative angle to the world origin is always the same as the original. From a human perspective, this should increase the realism of the augmentation method in the case of a single sensor. In our multi sensor setup, the effects of these different insertion methods need to be investigated. The combination of all these methods result in eight different variants of GT sampling, since the GT orientation and the relative orientation for GT positioning are the same.

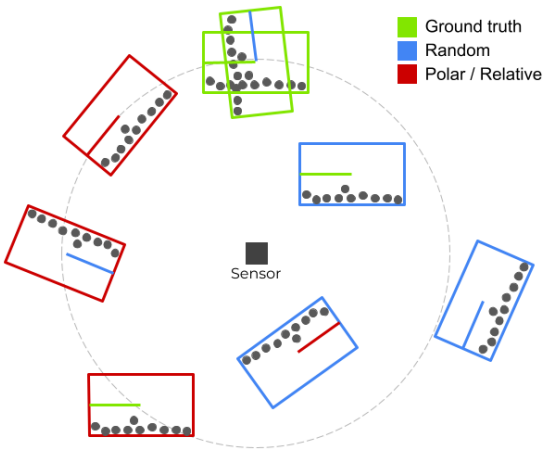

Figure 3. A sketch is shown for all eight ground truth sampling variations. LiDAR sensor is depicted as a black square. Boxes and points refer to an exemplary object placed with all eight variations. The lines inside the boxes indicated the orientation. The colors of the boxes represent the position variation, colors of their middle lines are the orientation variant. The dashed circle shows the polar distance of the ground truth.

3.4. Evaluation Metrics

We use the common mean average precision (mAP) metric for evaluation. The method used is similar to that in COCO [55] and is a non-interpolated version of the mAP as opposed to, for example, KITTI [3]. The implementation is based on the one in MMDET [56] and has been adapted to support IoU matching thresholds per class. The IoU thresholds used are the same as in KITTI, 0.7 for cars and 0.5 for pedestrians. Two filters were applied to the GT during training and evaluation. First, only boxes with at least 5 points in them were used, and second, all boxes outside of the defined grid range were discarded.

4. Experiments

In this section, the experimental results for in total three different waves of experiments are reported and discussed. In the first experiment, the effect of the different GT sampling variants as well as the effect of the polygon will be investigated. The second experiment will investigate how well GT sampling is suited for training with little available data. In a third experiment, this is examined for the special case that only one empty frame without any object occurrences is available for each site. Here, the usable training data is generated only by GT sampling. Unless otherwise noted, all experiments are conducted using the dataset presented previously for the car and pedestrian classes, and results are reported for the test split. The validation split was used for hyperparameter tuning such as learning rates, amount of epochs, and augmentation parameters. For all experiments, global rotation drawn from $U(-\pi, \pi)$, scaling drawn from $U(0.95, 1.05)$, and flip around both ground axes with a probability of 0.5 for each axes were applied. If GT sampling was performed, 10 cars and 10 pedestrians were tried to be inserted if no collision with another labeled object occurred. Ground planes are utilized for the height placement. More advanced augmentation methods were not applied to keep the interpretation of the experiments as simple as possible and to avoid further obscuring the results. The networks were trained for 100 epochs with an Adam-One-Cycle optimizer [57] with a learning rate of 0.001 for PointPillars and 0.003 for CenterPoint, respectively. Each training was repeated six times to counteract random effects during the training, and the median of the six results is reported.

4.1. Ground Truth Sampling Methods

In the first experimental wave, all eight meaningful combinations of the different positioning methods (ground truth, random, polar) and orientation methods (ground truth, random, relative) are evaluated with and without using the polygons. The results can be seen in Table 1. Intuitively, before looking at the results in more detail, one would expect

that using the polygon would improve the results in all cases. With all positioning methods, placement of objects outside of reasonable boundaries is possible. These are prevented by placing them inside the polygon, allowing the network to focus more on the actual region of interest. Regarding the GT sampling methods, based on experiments with single sensor setups one would expect the combination of polar positioning and relative orientation to produce the best results. It is unclear whether this also applies to our multi sensor setup.

Table 1. The mean average precision (car and pedestrian) for PointPillars and CenterPoint on test split with different combinations of GT sampling methods is shown. The left values for each cell are without polygon. The right values are with polygon. Values next to network names are baseline results without GT sampling. Reported values are the median of six trainings each. Higher values are better. The best results are marked in bold for PointPillars and CenterPoint, respectively.

Orientation / Position	mAP of Median in % ↑					
	PointPillars (62.82)			CenterPoint (68.58)		
	Ground Truth	Random	Relative	Ground Truth	Random	Relative
Ground truth	63.31/63.54	63.92/64.26	—/—	71.78/70.71	72.04/72.01	—/—
Random	63.92/**66.61**	63.22/65.49	64.35/65.70	72.14/72.95	72.51/72.96	72.46/72.64
Polar	64.16/65.16	63.84/64.81	63.63/65.72	72.32/**73.50**	72.27/72.02	72.16/71.58

Looking first at the results for PointPillars, it can be seen that the usage of the polygon increases the mAP of the medians in all variations of GT sampling. The highest mAP is achieved using the polygon and random positioning and GT orientation with 66.61%. The lowest mAP is reached without polygon and random positioning and random orientation. The two GT positioning variations benefit the least from the polygon, with 0.23 and 0.34 percentage points for GT and random orientation, respectively. Thus, the assumptions made previously are only partially accurate. Although the polygon improves the mAP in all cases, the impact on GT positioning is relatively small. This could be due to the restriction on the number of objects inserted, as all objects outside the polygons are discarded. Contrary to expectations, polar positioning and relative orientation do not perform best but are second best. Due to the multi sensor setup, this variant is not necessarily the most realistic in terms of scan pattern and distribution of points. The random positioning with the GT orientation, which performed best, is not realistic as well. Due to the random positioning, the orientation of the objects is not correct in most cases. Consequently, the random choice of position and orientation should also give very good results. In fact, however, this variant has only the fourth-highest mAP of 65.49%. Restricting the orientation angles to existing angles in the dataset could make the difference here.

The results can be roughly seen again for CenterPoint. The polygon also generally improves the mAP, except for the two GT position variations. The mAP deteriorates by 1.06 percentage points from 71.78% to 70.71%, and by 0.03 percentage points from 72.04% to 72.01% for GT positioning with GT orientation and random orientation, respectively. The attempted explanation in the case of PointPillars remains valid here as well. Objects outside the polygons are never added during the training. The best mAP is obtained for CenterPoint from polar positioning and GT orientation using the polygon. Thus, the mAP in this case reaches a value of 73.50%. The lowest mAP this time shows GT positioning and GT orientation with polygon with 70.71%. The best GT sampling variation for CenterPoint may differ from PointPillars, but still, the same argumentation applies for the orientation. The GT orientation narrows the possible rotation angles to those occurring in the dataset with corresponding distribution. Based on the results for both networks, it seems that random or polar positioning with guidance of the polygons are beneficial compared to GT positioning. One explanation could be the increased variety of object positions and, consequently, the scenes created after augmentation.

On the basis of these observations, the polygons are also used in the following experiments. Since it is not possible to use the GT information for some of the upcoming experiments, the random position and orientation are used in the following. On average, these have the highest mAP value of the GT variants.

4.2. Reduction of Dataset Size

The amount of training data is one of the most critical factors for good results of a deep learning approach. Therefore, the next wave of experiments will examine the ability of the GT sampling augmentation to compensate for the lack of data. Thus, the size of the training dataset is limited in this experiment to roughly 75%, 50%, and 25% of the original size, respectively. These subdivisions of the training dataset were created in the same way as train, valid, and test split. Thus, subsets of all training sequences according to the desired object ratios were created. The iterations per epoch are set to the initial 100% training set in all cases to allow a fair comparison. This is done by randomly reusing samples during training until the wanted number of iterations per epoch is reached. The database for the GT augmentation is adjusted to the current dataset size accordingly, such that only objects available in the current train set are present in the object database. The results can be seen in Figure 4. Before looking at the results, two things can be expected intuitively and based on the related work. First, the GT sampling augmentation can be expected to enhance the results in all cases for both networks. And secondly, it can be expected that the mAP decreases the less data is used.

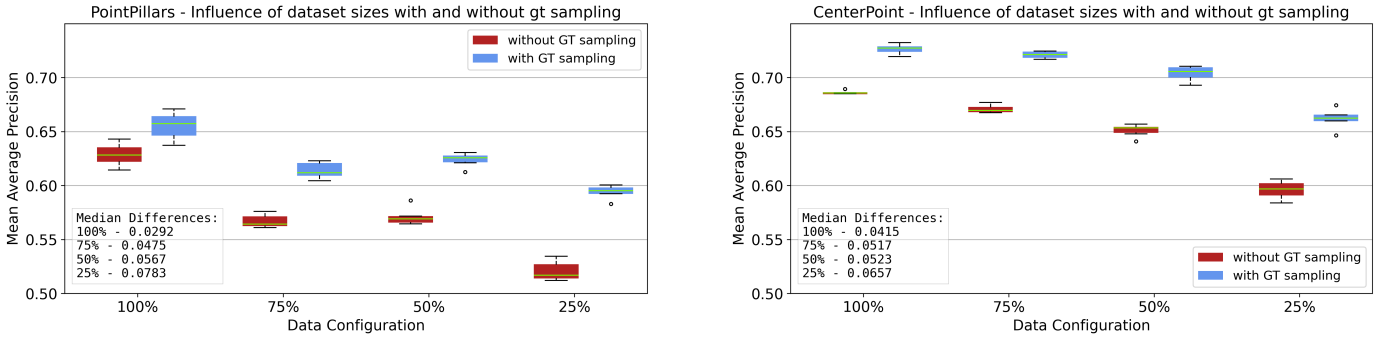

Figure 4. The mean average precision (car and pedestrian) on the test set for different sizes of the training split for PointPillars and CenterPoint without (red) and with (blue) GT sampling is shown as a boxplot. The differences in the medians between without and with GT sampling for each dataset size are shown in the bottom left corner. Best seen zoomed-in and in color.

Indeed, these two observations can be made for most of the results. The mAP decreases the less data is used with an exception at 75% to 50% dataset size. This is the only irregularity of this kind. GT sampling increases the results for all four dataset sizes. The gains with GT sampling amount to 2.92, 4.75, 5.67, and 7.83 percentage points, respectively. It can be observed that the median differences increase the lesser data used for training. Thus, it can be concluded that the GT sampling augmentation is not only able to increase the quality of the training results but is also able to cushion the effect of fewer training data by providing more variations of the available data. To reinforce this thesis, the experiments are repeated for the CenterPoint object detector. Again, with the utilization of GT sampling augmentation the median mAPs increase in all four cases. The differences between the medians with and without GT sampling amount to 4.15, 5.18, 5.23, and 6.58 percentage points, respectively. Therefore, the same observation as for PointPillars can be made. The effects of the smaller dataset are mitigated by GT sampling.

The question arises about how this cushion effect of GT sampling applies to ego-vehicle-based data. Therefore, the same experiment was performed on the KITTI dataset. The initial 100% split is the common one for KITTI. The other splits are again created by selecting whole sequences. Only CenterPoint is used for this experiment because its results

are more stable and have lower variance. The GT sampling variation is set to GT position and GT orientation. The common validation split is used for validation but not the official KITTI benchmark evaluation. Instead, the evaluation described before is used here. Thus, the results are only comparable to themselves, not with other publications regarding the KITTI benchmark for 3D OD. Table 2 shows the results.

Table 2. Results for KITTI (car and pedestrian) on validation split for CenterPoint with different training set sizes are shown. Reported are the results for the median of six training runs.

Training Data	mAP of Median in % ↑			
	100%	**75%**	**50%**	**25%**
KITTI + GT Sampling	51.90	51.08	47.35	45.25
KITTI − GT Sampling	48.18	46.95	42.45	34.48
Difference	3.73	4.13	4.90	10.78

The observations found on the infrastructural data can also be seen for the ego-vehicle-based KITTI data. The mAP of the medians is higher for all four sizes of the training set with GT sampling than without. Thus, the gain induced by utilizing the GT sampling method amounts to 3.73 percentage points for 100% dataset size and 10.78 percentage points for 25% dataset size. In the case of the KITTI dataset, the cushion effect of the GT sampling method is even more substantial compared to the infrastructural data used for the rest of this work. Therefore, the described effect of this augmentation method is not exclusive to a fixed environment.

4.3. Empty Case Experiments

In this experiment, the size of the training data is reduced to only one frame for both sites G and K, respectively, with no objects present. Thus, all meaningful training data is produced by the GT sampling augmentation. The GT database is taken from the 100%, 75%, 50%, and 25% trainings split, respectively, to investigate the performance for different amounts of objects. Once more, the experiments are performed with and without using the polygons to further look into its impact in this particular case. The results are depicted in Figure 5.

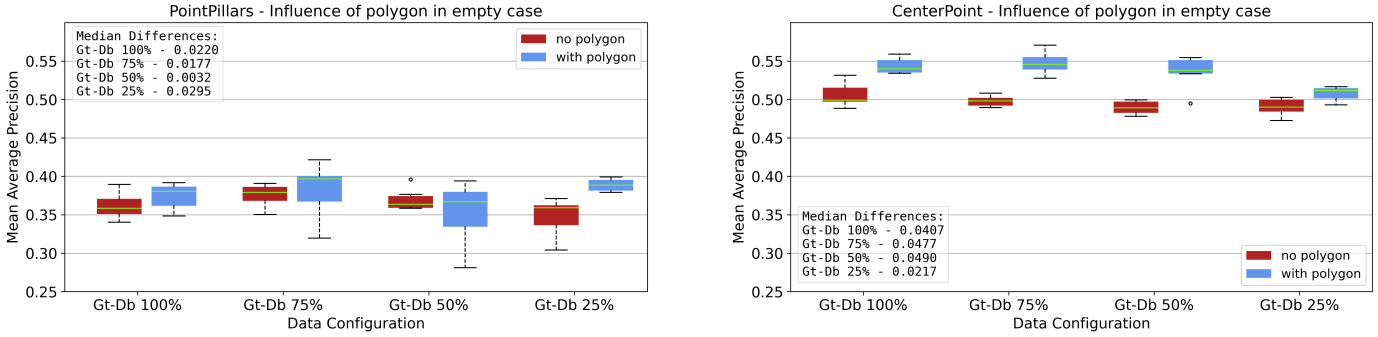

Figure 5. The mean average precision (car and pedestrian) on test set for different sizes of the GT database for PointPillars and CenterPoint without (red) and with (blue) polygon is shown as boxplot. The differences of the medians between without and with usage of polygon for each database size are shown inside the box. Best seen zoomed in and in color.

Based on the previous experiments, it can be expected that the polygon can increase the mAP. Furthermore, one could expect intuitively that the mAP decreases with smaller GT sampling database. Nothing can be said about the size of the mAP, as such an experiment has not yet been carried out in a similar form. Looking at the results, one of the previous assumptions is directly refuted. Other than expected, the results are comparably stable for the different sizes of the GT database. The largest difference between the four database

variations is only 2.06 percentage points. Considering the amount of data samples dropped, that is surprisingly small. Note that unexpectedly the mAP is highest for a database size of 75%. With utilization of the polygon the mAP is higher in all cases. Here, too, the results for the different database sizes are surprisingly close together. Once more, it can be observed that the mAP is highest for 75% database size. Looking at the results for CenterPoint, the same observations can be made. The polygon increases the result in all cases by up to 4.90 percentage points, which is a higher gain as shown in Table 1. The polygon gains even more value in the case of empty frames.

The number of objects is not as important as originally assumed. This might be caused due to a low overall variance of objects in the data. Due to the drop of mAP compared to the experiments regarding the dataset size, the exact positions and other physical effects, such as occlusion and sampling patterns, have an even stronger influence than previously expected. The better results of the 75% database compared to the 100% database indicate that the sheer number of objects is not the most relevant factor.

5. Conclusions & Further Work

In this work, we investigated 3D OD on an infrastructural LiDAR setup for autonomous factory driving. By using the GT sampling method, we were able to improve performance while compensating for the lack of labeling. Results were generally improved when a polygon was used to constrain the placement of objects. This restricts the placement of objects and is easy to create for a fixed environment. Moreover, the most commonly used variant of GT sampling, where objects are inserted in their original position and orientation, does not perform best. It has been shown that the GT sampling method can also mitigate the negative effect of less labeled data. This was demonstrated not only for the infrastructural setup, but also with an ego-vehicle based dataset. Finally, the possibility of training only with a database of objects inserted in the fixed environment was investigated. Surprisingly, the size of the database was found to have a smaller impact than expected.

This last experiment could be the starting point for future work. A possible continuation is the enrichment of the database for GT sampling with objects from other data sets. This could further reduce the labeling effort. The insertion of object models through ray casting is also an interesting extension that reaches the limits of simulation. Furthermore, an additional consideration of occlusion could be implemented, or a placement of the objects based on a probability map.

Author Contributions: Conceptualization, writing—review and editing, M.R., K.A., M.S. and B.S.; methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, visualization, M.R., K.A. and M.S.; supervision, M.S. and B.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data mainly used in this work is confidential and can not be provided.

Conflicts of Interest: M.R., K.A. and M.S. are employed by Valeo. All authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as potential conflicts of interest.

References

1. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuScenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11621–11631.

2. Chang, M.F.; Lambert, J.; Sangkloy, P.; Singh, J.; Bak, S.; Hartnett, A.; Wang, D.; Carr, P.; Lucey, S.; Ramanan, D.; et al. Argoverse: 3D tracking and forecasting with rich maps. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 8740–8749. [CrossRef]
3. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
4. Geyer, J.; Kassahun, Y.; Mahmudi, M.; Ricou, X.; Durgesh, R.; Chung, A.S.; Hauswald, L.; Pham, V.H.; Mühlegg, M.; Dorn, S.; et al. A2d2: Audi autonomous driving dataset. *arXiv* **2020**, arXiv:2004.06320.
5. Houston, J.; Zuidhof, G.; Bergamini, L.; Ye, Y.; Jain, A.; Omari, S.; Iglovikov, V.; Ondruska, P. One Thousand and One Hours: Self-driving Motion Prediction Dataset. In Proceedings of the Conference on Robot Learning (CoRL), Cambridge, MA, USA, 16–18 November 2020; pp. 1–10.
6. Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. Scalability in perception for autonomous driving: Waymo open dataset. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 2446–2454.
7. Creß, C.; Zimmer, W.; Strand, L.; Lakshminarasimhan, V.; Fortkord, M.; Dai, S.; Knoll, A. A9-Dataset: Multi-Sensor Infrastructure-Based Dataset for Mobility Research. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Aachen, Germany, 4–9 June 2022; pp. 965–970.
8. Wang, H.; Zhang, X.; Li, J.; Li, Z.; Yang, L.; Pan, S.; Deng, Y. IPS300+: A Challenging Multimodal Dataset for Intersection Perception System. *arXiv* **2021**, arXiv:2106.02781.
9. Yongqiang, D.; Dengjiang, W.; Gang, C.; Bing, M.; Xijia, G.; Yajun, W.; Jianchao, L.; Yanming, F.; Juanjuan, L. BAAI-VANJEE Roadside Dataset: Towards the Connected Automated Vehicle Highway technologies in Challenging Environments of China. *arXiv* **2021**, arXiv:2105.14370.
10. Yu, H.; Luo, Y.; Shu, M.; Huo, Y.; Yang, Z.; Shi, Y.; Guo, Z.; Li, H.; Hu, X.; Yuan, J.; et al. DAIR-V2X: A Large-Scale Dataset for Vehicle-Infrastructure Cooperative 3D Object Detection. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 21361–21370.
11. Kloeker, L.; Liu, C.; Wei, C.; Eckstein, L. Framework for Quality Evaluation of Smart Roadside Infrastructure Sensors for Automated Driving Applications. *arXiv* **2023**, arXiv:2304.07745.
12. Wu, A.; He, P.; Li, X.; Chen, K.; Ranka, S.; Rangarajan, A. An Efficient Semi-Automated Scheme for Infrastructure LiDAR Annotation. *arXiv* **2023**, arXiv:2301.10732.
13. Yan, Y.; Mao, Y.; Li, B. SECOND: Sparsely Embedded Convolutional Detection. *Sensors* **2018**, *18*, 3337. [CrossRef]
14. Mao, J.; Shi, S.; Wang, X.; Li, H. 3D Object Detection for Autonomous Driving: A Comprehensive Survey. *Int. J. Comput. Vis.* **2023**, pp. 1–55. [CrossRef]
15. Qian, R.; Lai, X.; Li, X. 3D Object Detection for Autonomous Driving: A Survey. *Pattern Recognit.* **2022**, *130*. [CrossRef]
16. Pan, X.; Xia, Z.; Song, S.; Li, L.E.; Huang, G. 3D Object Detection with Pointformer. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 7463–7472.
17. Shi, S.; Wang, X.; Li, H. PointRCNN: 3D object proposal generation and detection from point cloud. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 770–779. [CrossRef]
18. Yang, Z.; Sun, Y.; Liu, S.; Jia, J. 3DSSD: Point-based 3D Single Stage Object Detector. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11040–11048.
19. Yang, Z.; Sun, Y.; Liu, S.; Shen, X.; Jia, J. STD: Sparse-to-Dense 3D Object Detector for Point Cloud. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 1951–1960.
20. Deng, J.; Shi, S.; Li, P.; Zhou, W.; Zhang, Y.; Li, H. Voxel R-CNN: Towards High Performance Voxel-based 3D Object Detection. In Proceedings of the Conference on Artificial Intelligence (AAAI), virtual, 2–9 February 2021; pp. 1201–1209.
21. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. Pointpillars: Fast encoders for object detection from point clouds. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; Volume 2019-June, pp. 12689–12697. [CrossRef]
22. Liu, Z.; Zhao, X.; Huang, T.; Hu, R.; Zhou, Y.; Bai, X. TANet: Robust 3D Object Detection from Point Clouds with Triple Attention. In Proceedings of the Conference on Artificial Intelligence (AAAI), New York, NY, USA, 7–12 February 2020.
23. Mao, J.; Xue, Y.; Niu, M.; Bai, H.; Feng, J.; Liang, X.; Xu, H.; Xu, C. Voxel Transformer for 3D Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 3164–3173.
24. Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; Li, H. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10529–10538.
25. Shi, S.; Wang, Z.; Shi, J.; Wang, X.; Li, H. From Points to Parts: 3D Object Detection from Point Cloud with Part-aware and Part-aggregation Network. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 2647–2664. [CrossRef] [PubMed]
26. Wu, H.; Wen, C.; Li, W.; Li, X.; Yang, R.; Wang, C. Transformation-Equivariant 3D Object Detection for Autonomous Driving. *arXiv* **2022**, arXiv:2211.11962.

27. Yin, T.; Zhou, X.; Krähenbühl, P. Center-based 3D Object Detection and Tracking. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 11784–11793.
28. Zhou, Y.; Tuzel, O. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 4490–4499. [CrossRef]
29. Chen, H.; Liu, B.; Zhang, X.; Qian, F.; Mao, Z.M.; Feng, Y.; Author, C. A Cooperative Perception Environment for Traffic Operations and Control. *arXiv* **2022**, arXiv:2208.02792.
30. Kloeker, L.; Geller, C.; Kloeker, A.; Eckstein, L. High-Precision Digital Traffic Recording with Multi-LiDAR Infrastructure Sensor Setups. In Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020; pp. 1–8.
31. Zimmer, W.; Birkner, J.; Brucker, M.; Nguyen, H.T.; Petrovski, S.; Wang, B.; Knoll, A.C. InfraDet3D: Multi-Modal 3D Object Detection based on Roadside Infrastructure Camera and LiDAR Sensors. *arXiv* **2023**, arXiv:2305.00314.
32. Cai, X.; Jiang, W.; Xu, R.; Zhao, W.; Ma, J.; Liu, S.; Li, Y. Analyzing Infrastructure LiDAR Placement with Realistic LiDAR Simulation Library. *arXiv* **2022**, arXiv:2211.15975.
33. Zimmer, W.; Wu, J.; Zhou, X.; Knoll, A.C. Real-Time And Robust 3D Object Detection with Roadside LiDARs. In Proceedings of the 12th International Scientific Conference on Mobility and Transport: Mobility Innovations for Growing Megacities, Singapore, 5–7 April 2022; pp. 199–219.
34. Zimmer, W.; Grabler, M.; Knoll, A. Real-Time and Robust 3D Object Detection within Road-Side LiDARs Using Domain Adaptation. *arXiv* **2022** arXiv:2204.00132.
35. Arnold, E.; Dianati, M.; de Temple, R.; Fallah, S. Cooperative Perception for 3D Object Detection in Driving Scenarios using Infrastructure Sensors. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 1852–1864. [CrossRef]
36. Bai, Z.; Wu, G.; Qi, X.; Liu, Y.; Oguchi, K.; Barth, M.J. Infrastructure-Based Object Detection and Tracking for Cooperative Driving Automation: A Survey. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Aachen, Germany, 5–9 June 2022; pp. 1366–1373.
37. Sun, P.; Sun, C.; Wang, R.; Zhao, X. Object Detection Based on Roadside LiDAR for Cooperative Driving Automation: A Review. *Sensors* **2022**, *22*, 9316. [CrossRef]
38. Dosovitskiy, A.; Ros, G.; Codevilla, F.; López, A.; Koltun, V. CARLA: An open urban driving simulator. In Proceedings of the Conference on Robot Learning (CoRL), Mountain View, CA, USA, 13–15 November 2017.
39. Strigel, E.; Meissner, D.; Seeliger, F.; Wilking, B.; Dietmayer, K. The Ko-PER Intersection Laserscanner and Video Dataset. In Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 8–11 October 2014; pp. 1900–1901.
40. Busch, S.; Koetsier, C.; Axmann, J.; Brenner, C. LUMPI: The Leibniz University Multi-Perspective Intersection Dataset. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Aachen, Germany, 5–9 June 2022; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2022; Volume 2022-June, pp. 1127–1134. [CrossRef]
41. Bai, Z.; Wu, G.; Barth, M.J.; Liu, Y.; Sisbot, E.A.; Oguchi, K. PillarGrid: Deep Learning-based Cooperative Perception for 3D Object Detection from Onboard-Roadside LiDAR. In Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC), Macau, China, 24–28 September 2022; pp. 1743–1749. [CrossRef]
42. Bai, Z.; Wu, G.; Qi, X.; Liu, Y.; Oguchi, K.; Barth, M.J. Cyber Mobility Mirror for Enabling Cooperative Driving Automation in Mixed Traffic: A Co-Simulation Platform. *arXiv* **2022**, arXiv:2201.09463. [CrossRef]
43. Hahner, M.; Dai, D.; Liniger, A.; Gool, L.V. Quantifying Data Augmentation for LiDAR based 3D Object Detection. *arXiv* **2020**, arXiv:2004.01643.
44. Reuse, M.; Simon, M.; Sick, B. About the Ambiguity of Data Augmentation for 3D Object Detection in Autonomous Driving. In Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 979–987.
45. Cheng, S.; Leng, Z.; Cubuk, E.D.; Zoph, B.; Bai, C.; Ngiam, J.; Song, Y.; Caine, B.; Vasudevan, V.; Li, C.; et al. Improving 3D Object Detection through Progressive Population Based Augmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 279–294.
46. Choi, J.; Song, Y.; Kwak, N. Part-Aware Data Augmentation for 3D Object Detection in Point Cloud. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 3391–3397.
47. Xiao, A.; Huang, J.; Guan, D.; Cui, K.; Lu, S.; Shao, L. PolarMix: A General Data Augmentation Technique for LiDAR Point Clouds. *arXiv* **2022**, arXiv:2208.00223.
48. Fang, J.; Zuo, X.; Zhou, D.; Jin, S.; Wang, S.; Zhang, L. LiDAR-Aug: A General Rendering-based Augmentation Framework for 3D Object Detection. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 4710–4720.
49. Zhu, B.; Jiang, Z.; Zhou, X.; Li, Z.; Yu, G. Class-balanced Grouping and Sampling for Point Cloud 3D Object Detection. *arXiv* **2019**, arXiv:1908.09492.
50. Šebek, P.; Pokorný, Š.; Vacek, P.; Svoboda, T. Real3D-Aug: Point Cloud Augmentation by Placing Real Objects with Occlusion Handling for 3D Detection and Segmentation. *arXiv* **2022**, arXiv:2206.07634.

51. Lee, D.; Park, J.; Kim, J. Resolving Class Imbalance for LiDAR-based Object Detector by Dynamic Weight Average and Contextual Ground Truth Sampling. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Waikola, HI, USA, 2–7 January 2023; pp. 682–691.
52. Shi, P.; Qi, H.; Liu, Z.; Yang, A. Context-guided ground truth sampling for multi-modality data augmentation in autonomous driving. In Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC), Bilbao, Spain, 24–28 September 2023; John Wiley and Sons Inc.: Hoboken, NJ, USA, 2023; Volume 17, pp. 463–473. [CrossRef]
53. OpenPCDet Development Team. OpenPCDet: An Open-Source Toolbox for 3D Object Detection from Point Clouds. 2020. Available online: https://github.com/open-mmlab/OpenPCDet (accessed on 17 November 2023).
54. Vora, S.; Lang, A.H.; Helou, B.; Beijbom, O. PointPainting: Sequential Fusion for 3D Object Detection. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 4604–4612.
55. Lin, T.Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common Objects in Context. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
56. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv* **2019**, arXiv:1906.07155.
57. Smith, L.N. A disciplined approach to neural network hyper-parameters: Part 1—Learning rate, batch size, momentum, and weight decay. *arXiv* **2018**, arXiv:1803.09820.