Real-Aug: Realistic Scene Synthesis for LiDAR Augmentation in 3D Object Detection

Jinglin Zhan[1], Tiejun Liu[1], Rengang Li[1], Jingwei Zhang[1], Zhaoxiang Zhang[3], Yuntao Chen[2*]

[1]Inspur Electronic Information Industry Co.,Ltd.
[2]Centre for Artificial Intelligence and Robotics, HKISI, CAS
[3]Institute of Automation, CAS

Abstract

Data and model are the undoubtable two supporting pillars for LiDAR object detection.However, data-centric works have fallen far behind compared with the ever-growing list of fancy new models. In this work, we systematically study the synthesis-based LiDAR data augmentation approach (so-called GT-Aug) which offers maxium controllability over generated data samples. We pinpoint the main shortcoming of existing works is introducing unrealistic LiDAR scan patterns during GT-Aug. In light of this finding, we propose Real-Aug, a synthesis-based augmentation method which prioritizes on generating realistic LiDAR scans. Our method consists a reality-conforming scene composition module which handles the details of the composition and a real-synthesis mixing up training strategy which gradually adapts the data distribution from synthetic data to the real one. To verify the effectiveness of our methods, we conduct extensive ablation studies and validate the proposed Real-Aug on a wide combination of detectors and datasets. We achieve a state-of-the-art 0.744 NDS and 0.702 mAP on nuScenes test *set. The code shall be released soon.*

1. Introduction

The field of autonomous driving has seen a surge of interest in LiDAR 3D object detection due to its ability to overcome the limitations of image-based methods and improve overall system reliability. There has been an ever-growing list of novel point cloud feature extractors [44, 19, 38, 10, 5, 6, 31] and new detection paradigms [19, 31, 40, 25] in this area. In the meantime, data-centric works in this field fall far behind model-centric ones, though data and model are widely recognized as two fundamental components in perception tasks. The quantity and quality of point cloud data play key roles in achieving a performant detector and data augmentation has always been an integral part of this. However, existing works of LiDAR data augmentation either focus more on data under special weather condition [14, 13] or fall short at verifying their effectiveness [24, 11, 43, 15] on large-scale real-world datasets [2, 32]. In this work, we systematically study the synthesis-based approach for LiDAR data augmentation, which indicates the produce of placing a set of object point clouds into scene point clouds [38, 11]. Compared with scan-based(e.g. flip, scale, rotate) and object-based LiDAR data augmentation [7], synthesis-based ones generate diverse LiDAR scans and offer fine-grain controllability over the synthesized scenes like over-sampling objects from rare classes.

However, simply applying the vanilla synthesis-based LiDAR data augmentation(so-called GT-Aug) does not lead to satisfactory results on modern large-scale datasets like nuScenes and Waymo [33]. To explain the above phenomenon, we take the bicycle class in nuScenes as an example and plot its PR-Curve for a CenterPoint [40] detector trained with and without GT-Aug. Fig. 1 shows that after applying GT-Aug, the detector is able to recall more objects at the cost of generating more false positives. The PR-curve clearly reveals a downside of vanilla GT-Aug - introducing non-existing LiDAR scans pattern into the original dataset.

Real-Aug, which prioritizes on the realisticness of newly synthesized scenes, is proposed in this paper to overcome the limitation of vanilla synthesis-based LiDAR augmentation methods. It consists a reality-conforming scene composition module for handling intricate technical details throughout scene composition and a real-synthesis mixing up training strategy which gradually aligns the distribution from synthetic data to the real one. The effectiveness of Real-Aug are validated across multiple LiDAR object detection datasets for different detectors. We achieve a 4.7% 3D mAP improvement on KITTI 3D object detection benchmarks (2.1%, 4.0%, 8.1% for car, pedestrian

*Corresponding author: chenyuntao08@gmail.com

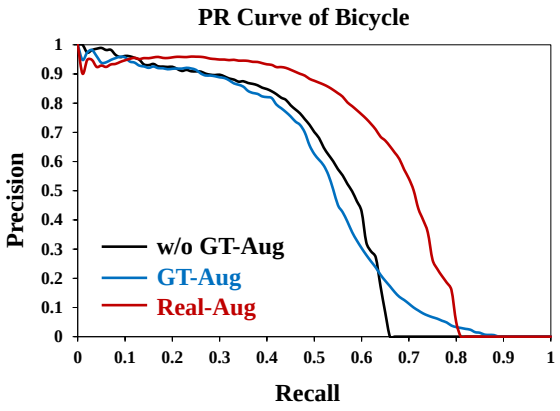

Figure 1. Pr-curve of the bicycle at different scene synthesis steps.

and cyclist respectively) for a baseline SECOND [38] detector. Notably, we achieve a 74.4% NDS and a 70.2% mAP on the $test$ set of nuScenes 3D object detection benchmark[1]. There is a significant 6.1% NDS and a 8.7% mAP improvement over our baseline CenterPoint [40] detector solely through data augmentation.

Our contributions can be summarized as follows:

(1) We reveal the realisticness issue of vanilla synthesis-based LiDAR data augmentation.

(2) We present a well-designed Real-Aug scheme, which features a reality-conforming scene composition module and a real-synthesis mixing up strategy.

(3) We highlight the effectiveness of Real-Aug by applying it cross a wide range of detectors for multiple datasets and achieve state-of-the-art results.

(4) We validate technical advantages of Real-Aug, including its robustness to hyper parameter choices and the improvement of data utilization.

2. Related Work

2.1. Non-synthesis Data Augmentation Methods

Data augmentaion methods are widely applied to artificially diversify the dataset and help promote the detectors' capacity. Commonly-used strategies include random flip, rotation, scale, and translation at both scene- and instance-level. Some physically valid simulation methods were deployed to deal with the detection challenges under foggy or snowy weather [14, 13]. PointPainting augmented point clouds with image semantics. It appended the predicted class score from image semantic segmentation network to each point [36]. Inspired by PointPainting, PointAugmenting decorated point clouds with corresponding point-wise CNN features extracted from 2D image detectors [37].

2.2. Synthesis-based Data Augmentation Methods

Mix3D devised a "mixing" technique to create new scenes by combining two augmented ones while ensuring sufficient overlap [27]. A sensor-centric approach was applied for maintaining the data structure of synthesized scenes consistent with the lidar sensor capacities [15]. One of the most popular synthesis-based data augmentation methods, Ground-Truth augmentation (GT-Aug), was presented by Yan et al. [38] in 2018 and applied in multiple LiDAR detection tasks [19, 40, 45, 39, 6, 17, 18, 1]. On top of GT-Aug, many techniques were proposed for diversifying the ground-truth database. Part-aware and shape-aware gt sampling divided objects into partitions and stochastically applied augmentation methods to each local region [8, 43]. Pattern-aware gt sampling downsampled the points of objects to create a new one with farther distance [16]. PointMixup utilized an interpolation method to compose new objects [4]. PointCutMix replaced part of the sample with shape-preserved subsets from another one [20]. Fang et al. proposed a rendering-based method for inserting visual objects simulated by CAD into the real background [11].

Placing instances at semantically plausible positions was proved to be essential to guarantee the improved performance for 2D object detectors [9, 35, 42]. In LiDAR-based 3D object detection, collision problem is commonly seen as a physical placement issue in GT-Aug. Yan et al. performed a collision test after ground-truth sampling and removed any sampled objects that collided with others [38]. Competition strategy, which remains the points closer to the sensor, was employed to generate a more physical synthesized scene [15]. LiDAR-Aug leveraged a "ValidMap" to generate poses for achieving more reasonable obstacle placements [11]. It divided point clouds into pillars and filtered out valid pillars according to the height distribution. Although some researchers have noticed the placement issue in GT-Aug, the systematical studies about unrealistic LiDAR scan patterns in synthesized scenes is still woefully insufficient. Particularly, the deviations of data distribution from synthetic data to the real one are rarely discussed. As a result, existing synthesis-based LiDAR augmentations only achieved limited success, especially in large-scale datasets like nuScenes and Waymo.

3. Realistic Scene Synthesis

Our method mainly consists of a reality-conforming scene composition module and a real-synthesis mixing up training strategy. We introduce a reality-conforming score in Sec. 3.1 to measure the realisticness of synthesized scans. The details of scene composition is described in Sec. 3.2. We elaborate the training strategy of how to blend synthe-

[1]Rank 1st among all LiDAR-only object detection methods by the time of submission

sized and real LiDAR scans to achieve the optimal performance in Sec. 3.3.

3.1. Reality-Conforming Score

Finding a proper metric to measure the realisticness is at the core of our method. The most direct approach is measuring how well a model trained on the vanilla *train* set perform on the augmented *val* set. If the newly synthesized scenes conform to the same data distribution of the train set, the vanilla model could recognize it without any performance degradation. Therefore, we define a reality-conforming score Re directly based on metric of the perception task at hand, which could seen as a generalized version of detection agreement score mentioned in LiDARSim [24]. Specifically, for LiDAR object detection, we define the the reality-conforming score $Re(\text{mAP})$ as the ratio between mAP tested on val set with and without ground-truth augmentation.

$$Re(\text{mAP}) = \frac{\text{mAP}_{\text{aug}}}{\text{mAP}_{\text{noaug}}} \tag{1}$$

The reality-conforming score could also be defined over other metrics, like mIoU for semantic segmentation and PQ for panoptic segmentation.

3.2. Reality-Conforming Scene Composition

We perform our scene synthesis solely via the object-scene composition approach, which could be formulated as sequentially place LiDAR points of one or more objects into a existing scene. We refer the set of all placeable objects as the *object bank* and all scenes as the *scene bank* accordingly.

The composition approach is widely adopt in both 2D and 3D object detection [41, 34, 23, 38, 40] but achieves far less success in large scale LiDAR object detection benchmark like nuScenes [2] and Waymo [32]. We find the realisticness is the key to the success of this kind of methods and elaborate the technical details affecting the realisticness of synthesized LiDAR scans in the following sections. Related ablation studies could be found in Tab. 5

3.2.1 Placeable Location Detection Module

It is obvious that not everywhere in a LiDAR scan is a suitable spot for placing an object. Accurate modeling where each kind of object could appear requires a ton of extra knowledge, we simplify this task by assuming most objects of interest are on the drivable surface. This simplification is not perfect in every case(e.g. a pedestrian could appear on the sidewalk) but it is a good approximation as objects appeared in the drivable area affect the behavior of ego vehicle most.

We adopt a light-weight coordinate MLP [26] as our placeability estimator. The input of the network is the coordinates and reflectivity of LiDAR points (x, y, z, r). We use a Fourier [26] encoder of order $L = 10$ to map the points into 64-dim embeddings. We use binary cross-entropy loss for training the estimator. The supervision could come from either model-based ground estimation method like PatchWorks [22, 21] or manually labelled LiDAR semantic segmentation.

An alternative way would be directly using ground estimator like PatchWorks but we choose the coordinated-MLP apporach for its denoising nature and low latency($<$ 1ms) on modern hardware.

3.2.2 Design Choices of Scene Composition

Object Position. Different from the vanilla GT-Aug which always place the object at the same position from where it been take, there are three factors in Real-Aug to consider when choosing physically reasonable position of a sampled object: the distance and observing angle from ego vehicle, as well as the predicted placeability.

Assuming the XOY location and heading of an object in its original scene is denoted as (x, y, θ) and its location and heading in the synthesized scene as (x', y', θ'), the distance constraint could be specified as,

$$x^2 + y^2 = x'^2 + y'^2 \pm \Delta \tag{2}$$

and observing angle constraint as

$$\theta + \arctan\frac{y}{x} = \theta' + \arctan\frac{y'}{x'}. \tag{3}$$

Here Δ is the error tolerance threshold as finding an exact match for distance constraint is almost impossible for a limited number of points in a single scan. By default Δ is set to the half length of object $L/2$.

The distance constraint ensures the realistic point density and the observing angle constraint guarantees the a realistic scan pattern. Finally for the placeable constraint, we simply reject all locations (x', y') with predicted placeability < 0.5 to avoid placing objects into unfavorable locations.

Object Heading. Taking a closer look at Eq. 2 and Eq. 3 we can find that the heading angle θ' of our object is a free variable, which opens up the possibility of choosing a natural heading angle for the object instead of placing it into the scene with some wired random headings. We select the heading angle θ' of our object to conform the heading distribution of objects from the same category in the scene$\{\theta_c\}$ via measuring the cosine similarity between headings,

$$\theta' = \arg\max_\theta \sum_c \cos(\theta - \theta_c) \tag{4}$$

If there is no object in the scene with the same category, we simply choose the most frequent heading for our object.

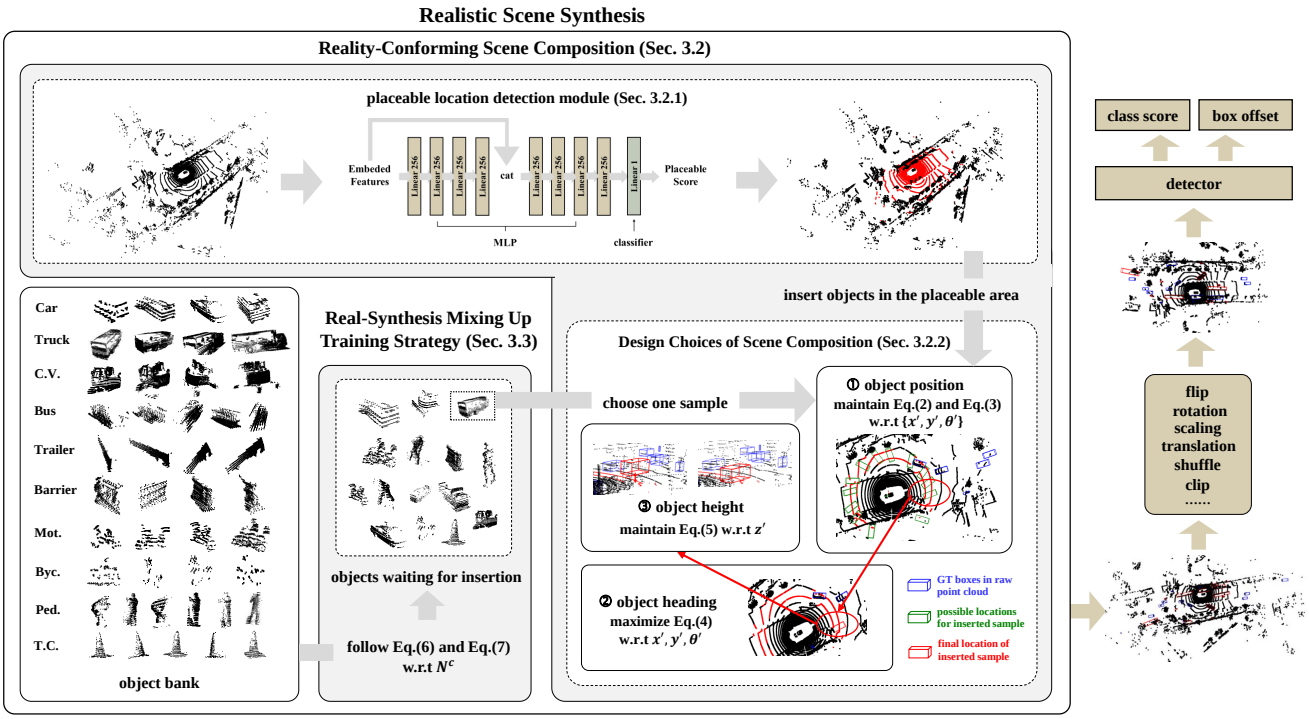

Figure 2. Overview of the proposed realistic scene synthesis for LiDAR augmentation (Real-Aug) in 3D object detection. (Points reflected from placeable area is painted with red. GT boxes in raw point cloud, possible locations for inserted sample and final location of inserted sample are presented by the bounding boxes with blue, green and red lines respectively.)

Object Height. Using the original height of an object could make it fly over or under the new scene's ground plane. We mitigate this by setting its bottom height in the new scene as the mean height of all ground points $\{z_g\}$ enclosed by the bounding box.

$$z' - H/2 = \text{avg}(z_g) \tag{5}$$

Collision Avoidance. In order to avoid collision, we use the same strategy in GT-Aug[38] and remove placed object if it overlaps with the existing ones.

3.3. Real-Synthesis Mixing Up Training Strategy

In this section, we elaborate a real-synthesis mixing up training strategy for gradually adapting the detector from synthetic data distribution to the real one. We introduce the real scene-category relation and scene-crowdedness relation alignments in Sec. 3.3.1 and Sec.3.3.2 to fulfill the full potential of Real-Aug.

3.3.1 Align to Realistic Scene-Category Relation

Existing large-scale autonomous driving datasets [3, 32, 2, 12] make great efforts at ensuring the diversity of video clips and generally contains scenarios from a wide range of weather, lighting and road conditions. However, the rich diversity of scenarios poses challenge for scene-synthesis. For instance, putting a bike rider on a closed cross-state highway or synthesizing a man holding an umbrella into a sunny afternoon country road both make the synthesized scenes highly unrealistic. So maintaining a reasonable scene-category relation is of great importance in our work.

We summarize the scene-category relation for three 20s video clips from nuScenes in Tab.1. It reveals strong connection between category distribution of objects and their

class	scene-0184 (residential)	scene-0234 (downtown)	scene-0399 (expressway)
Car	54	714	1155
Truck	0	217	115
C.V.	0	0	0
Bus	0	59	41
Trailer	0	80	0
Barrier	0	95	0
Mot.	0	0	13
Byc.	13	0	0
Ped.	0	1404	0
T.C.	0	83	59

Table 1. The category distributions of objects at scene-0184, scene-0234, scene-0399.

surrounding environments. Previous approaches like GT-Aug fail to realize the scene-category relation in driving scenarios, leading to detectors which hallucinate non-existing false positives as shown in Fig. 1.

In order to both enjoy the enhanced feature learning via scene augmentation and respect the original scene-category relation, we propose a mix up training strategy. The plain strategy is to place a preset number of objects for each category into the scene for each LiDAR scan. Using c to denote the object category, the number of inserted objects can be represented as N^c_{plain}. The plain strategy totally ignores the scene-category relation but generates diverse scans which is good for feature learning. Another strategy is to strictly respect the scene-category relation by inserting objects only from the existing categories in the scan. We use N^c_{exist} to denote the number of objects inserted by this strategy, where $N^c_{\text{exist}} = 0$ for categories not exist on this scan. We use a hyper-parameter $\alpha \in [0, 1]$ to balance above two strategies and obtain our final strategy N^c. We align the data distribution to the real scene-category relation by gradually annealing α from 1 to 0 towards the end of training.

$$N^c = N^c_{\text{plain}} \times \alpha + N^c_{\text{exist}} \times (1 - \alpha) \tag{6}$$

3.3.2 Align to Realistic Scene-Crowdedness Relation

class	fg/bg(w/o GT-Aug)	fg/bg(w/ GT-Aug)	Ratio
Car	0.791%	0.799%	1.01
Truck	0.193%	0.306%	1.59
C.V.	0.042%	0.621%	14.79
Bus	0.060%	0.394%	6.57
Trailer	0.095%	0.634%	6.67
Barrier	0.246%	0.336%	1.39
Mot.	0.021%	0.378%	18.00
Byc.	0.020%	0.384%	19.20
Ped.	0.344%	0.379%	1.10
T.C.	0.143%	0.231%	1.62

Table 2. Comparing the ratios of foreground voxels to background voxels with and without GT-Aug. The resolution used here for voxelization is [0.075m,0.075m,0.2m] and number of voxels are counted on a feature map of a total stride of 8.

While our composition-based augmentation greatly facilitate the feature leanring for LiDAR object detector, it inevitably distorting the crowdedness of the real scenes as we only inserting objects into scenes.

Tab. 2 demonstrates the increase of foreground voxels on the feature of a CenterPoint detector when applying GT-Aug for LiDAR augmentation. Among 10 categories defined in nuScenes detection task, fg/bg of bicycle, motorcycle and construction vehicle rank the top three with increased times of 19.2, 18.0, 14.8. The significant increase of foreground voxels encourages detectors to make more predictions than what there really are. To deal with this, we introduce another hyper-parameter β and also use a gradually annealing strategy to decrease its value for aligning the real scene-crowdedness relation.

$$N^c = (N^c_{\text{plain}} \times \alpha + N^c_{\text{exist}} \times (1 - \alpha)) \times \beta \tag{7}$$

4. Experiments

For verifying the effectiveness and generality of Real-Aug, we conduct extensive experiments on nuScenes and KITTI. Some brief descriptions of datasets are summarized in Sec. 4.1. The implement details are shown in Sec 4.2. We elaborate the evalutation results on the $test$ set of nuScenes and KITTI in Sec. 4.3. Exhaustive ablations are performed and discussed in Sec. 4.4.

4.1. Datasets and Metrics

nuScenes Dataset[2]. nuScenes dataset is a large-scale dataset designed for accelerating researches on multiple tasks in autonomous driving scenarios. It comprises over 1,000 scenes, which are divided into 700 scenes for training, 150 scenes for validation and 150 scenes for testing. The full dataset consists 390k 360-degree LiDAR sweeps, which are collected by Velodyne HDL-32E with 20Hz capture frequency. The nuScenes detection task requires detecting 10 object classes with 3D bounding boxes, attributes, and velocities. For evaluation, the official detection metrics, including NuScenes Detection Score (NDS) and mean Average Precision (mAP), are used.

KITTI Dataset[12]. KITTI dataset is a widely used benchmark dataset for 3D object detection. It contains 7481 frames for training and 7518 frames for testing. As Ref [19, 38], the training frames are further divided into *train* set with 3712 frames and *val* set with 3769 frames. The KITTI detection task requires detecting 3 object classes with 3 difficulty levels (Easy, Moderate, Hard). Detectors are evaluated by 3D Average Precision AP_{3D}, which is calculated with recall 40 positions (R40).

4.2. Implement Details

Our implementation of LiDAR-based 3D object detection is based on open-sourced OpenPCDet [28] and the published code of CenterPoint [40]. For nuScenes, we choose CenterPoint-Voxel, CenterPoint-Pillar, SECOND-Multihead frameworks for analysis. For KITTI, we choose SECOND and PointPillar frameworks for anaylsis. Detectors are trained with a batch size of 32 on 8 A100 GPUs. We utilize adam optimizer with one-cycle learning rate policy. We use the same data augmentation methods (except GT-Aug) and network designs as prior works [40, 45, 38, 19]. The total training epochs for nuScenes and KITTI are set as

Methods	NDS	mAP	Car	Truck	C.V.	Bus	Trailer	Barrier	Mot.	Byc.	Ped	T.C.
CBGS [45]	0.633	0.528	0.811	0.485	0.105	0.549	0.429	0.657	0.515	0.223	0.801	0.709
PillarNet-34 † [30]	0.714	0.660	0.876	0.575	0.279	0.636	0.631	0.772	0.701	0.423	0.873	0.833
LidarMultiNet [39]	0.716	0.670	0.869	0.574	0.315	0.647	0.610	0.735	0.753	0.476	0.872	0.851
Transfusion_L † [1]	0.702	0.655	0.862	0.567	0.282	0.663	0.588	0.782	0.683	0.442	0.861	0.820
LargeKernel3D_L [6]	0.705	0.653	0.859	0.553	0.268	0.662	0.602	0.743	0.725	0.466	0.856	0.800
LargeKernel3D_L † [6]	0.728	0.688	0.873	0.591	0.302	0.685	0.656	0.750	0.778	0.535	0.883	0.824
AFDetV2 [17]	0.685	0.624	0.863	0.542	0.267	0.625	0.589	0.710	0.638	0.343	0.858	0.801
MDRNet_L [18]	0.705	0.652	0.865	0.545	0.257	0.638	0.589	0.748	0.731	0.452	0.866	0.829
MDRNet_L † [18]	0.720	0.672	0.873	0.577	0.283	0.665	0.622	0.752	0.744	0.485	0.876	0.843
CenterPoint [40]	0.655	0.580	0.846	0.510	0.175	0.602	0.532	0.709	0.537	0.287	0.834	0.767
CenterPoint+Real-Aug	**0.709**	**0.658**	**0.852**	**0.546**	**0.313**	**0.652**	**0.600**	**0.770**	**0.726**	**0.464**	**0.857**	**0.800**
CenterPoint † [40]	0.673	0.603	0.852	0.535	0.200	0.636	0.560	0.711	0.595	0.307	0.846	0.784
CenterPoint+Real-Aug †	**0.734**	**0.690**	**0.858**	**0.582**	**0.349**	**0.673**	**0.639**	**0.787**	**0.784**	**0.523**	**0.881**	**0.811**
SparseFishNet3D+Real-Aug †	**0.744**	**0.702**	**0.868**	**0.593**	**0.355**	**0.701**	**0.656**	**0.776**	**0.783**	**0.551**	**0.890**	**0.845**

Table 3. Comparison with state-of-the-art methods on *test* sets of nuScenes detection benchmarks.(†: test-time augmentation.)

Method	Car			Pedestrian			Cyclist			mAP
SECOND [29]	85.3%	76.6%	71.8%	43.0%	35.9%	33.6%	71.1%	55.6%	49.8%	58.1%
SECOND+Real-Aug	86.8%	78.4%	74.7%	47.2%	40.3%	37.2%	81.4%	63.2%	56.2%	62.8%

Table 4. mAP_{3D} difference of well-trained SECOND model (using GT-Aug or Real-Aug) on KITTI *test* sets.

20 and 80 respectively. We adopt weighted Non-Maxima Suppression (NMS) [10] during inference. The placeability estimator described in Sec. 3.2.1 is supervised by the ground labels generated from PatchWorks [22, 21].

4.3. Evaluation on nuScenes and KITTI *test* set

nuScenes. As shown in Tab. 3, the CenterPoint detector trained with Real-Aug outperforms other state-of-the-art LiDAR-only methods on the nuScenes *test* set. Comparing to the work reported by Yin et al. [40], Real-Aug promotes the NDS and mAP of CenterPoint from 0.673, 0.603 to 0.734, 0.690. Notably, our methods bring significant mAP improvement for bicycle, motorcycle and construction vehicle by 21.6%, 18.9%, 14.9% over the baseline. Combining with the SparseFishNet3D backbone which is described in Sec.4.4.6, we achieve 0.744 NDS and 0.702 mAP on nuScenes *test* set.

KITTI. The evaluation metric of KITTI changes from AP_{3D} R11 to AP_{3D} R40. For an unbiased comparison, we take the submitted results which are achieved based on the reimplement of OpenPCDet [29] as a reference. The results shown in Tab. 4 authenticate the effectiveness of Real-Aug in KITTI dataset. There is an average boost of 4.7% AP_{3D} for all classes with different difficulties (2.1%, 4.0%, 8.1% for car, pedestrian and cyclist respectively).

4.4. Ablations And Analysis

Real-Aug is investigated with extensive ablation experiments on *val* set of nuScenes. In Sec. 4.4.1, we discuss the realisticness of synthesized scenes. The advantages of Real-Aug, including its effectiveness, robustness and its role in promoting the data utilization, are analyzed from Sec. 4.4.2 to Sec. 4.4.5. The optimized backbone, which is called SparseFishNet3D, is described in Sec. 4.4.6 for achieving better detection performance.

4.4.1 Reality-Conforming Score

Methods	Position	Heading	Height	Category	Re(mAP)
w/o GT-Aug					1.000
GT-Aug					0.744
Real-Aug	✓	✓	✓	✓	0.933
Real-Aug		✓	✓	✓	0.870
Real-Aug	✓		✓	✓	0.880
Real-Aug	✓	✓		✓	0.906
Real-Aug	✓	✓	✓		0.795

Table 5. Re(mAP) of synthesized scenes generated by different methods.

The reality-conforming score Re(mAP) defined in Sec. 3.1 is deployed for comparing realisticness between

	Methods	NDS	mAP	Car	Truck	C.V.	Bus	Trailer	Barrier	Mot.	Byc.	Ped	T.C.
0	GT-Aug	0.666	0.595	0.849	0.586	0.184	0.696	0.389	0.687	0.587	0.427	0.850	0.696
1	w/o GT-Aug	0.659	0.594	0.837	0.571	0.170	0.677	0.372	0.678	0.648	0.449	0.847	0.696
2	+ Real Composition	0.678	0.611	0.851	0.593	0.206	0.722	0.425	0.696	0.611	0.458	0.849	0.700
3	+ MixUp Training	**0.694**	**0.641**	0.850	**0.605**	**0.247**	0.713	**0.435**	0.689	**0.704**	**0.597**	0.849	**0.717**

Table 6. Effectiveness of reality-conforming scene composition and real-synthesis mixing up training strategy. (Evaluation dataset: nuScenes *val* set, model: CenterPoint-Voxel, voxel size: [0.075,0.075,0.2])

GT-Aug and Real-Aug. Simultaneously, we ablate the contribution of each component in Real-Aug. The detector, which possess a framework of CenterPoint-Voxel and a voxel size of [0.075,0.075,0.2], is trained on the vanilla nuScenes *train* set. The inference results on the augmented nuScenes *val* set are compared and shown in Tab. 5. In contrast to GT-Aug, the Re(mAP) of Real-Aug increases from 0.744 to 0.933, which means our proposed reality-conforming scene composition approach and real-synthesis mixing up training strategy can effectively shrinkage the gap between the synthesized scenes and the real one. Each component, including the physically reasonable object position, heading, height and real scene-category relation, matters for realizing realistic scene synthesis for LiDAR augmentation in 3D object detection. The alignment of real scene-crowdedness relation finally regress to the raw point clouds without any synthesis-based augmentations.

4.4.2 Effectiveness of Real-Aug on nuScenes *val* set

The effectiveness of Real-Aug, which contains a reality-conforming scene composition module and a real-synthesis mixing up training strategy, is validated on nuScenes *val* set. Our proposed reality-conforming scene composition module boosts NDS from 0.666 to 0.678 and mAP from 0.595 to 0.611. Combining the real-synthesis mixing up training strategy, the performance of CenterPoint-Voxel can be further optimized and finally reaches 0.694 NDS and 0.641 mAP.

The information of objects in nuScenes *trainval* set are summarized in Appendix A.1 to help analyze the phenonmena shown in Tab. 6. In nuScenes dataset, car and pedestrian are two categories with most abundant data. Car exists in 97.73% frames and pedestrian exists in 79.19% frames. As a result, most detectors perform well on them. Although truck exists in 70.26% frames, which ranks the 3rd place, the corresponding performance of detector is still worse than expectation. Trucks' unsatisfactory detection accuracy should be attributed to the rare low points density inside their bounding boxes. In each voxel with the size of [0.075,0.075,0.2], the average points number of truck is 0.044, which is mucher lower than that of barrier (0.550) and traffic cone (0.534). The above points-density-related issues are more serious in construction vehicle and trailer (with only 0.018 and 0.017 points per voxel). As a result, it is hard for detectors to distinguish them from background points. The abnormal mAP decline of motorcycle and bicycle when introducing GT-Aug also attract our attention, which may own to their complex morphology. As shown in Tab. 6, the effectivness of GT-Aug severely suffer from the dramatical mAP degradation of motorcycle and bicycle.

The proposed Real-Aug minimizes the misleading from non-existing LiDAR scan patterns introduced by GT-Aug. It excels at dealing with the complex-morphology and low-points-density issues, which is beneficial for unleashing the full power of detectors. Replacing GT-Aug with Real-Aug achieves a boost of 6.3%, 4.6%, 11.7%, 17.0% AP for construction vehicle, trailer, motorcycle and bicycle respectively.

4.4.3 Robustness of Real-Aug

Model	Method	voxel size	NDS	mAP
SECOND	GT-Aug	[0.1,0.1,0.2]	0.620	0.505
SECOND	Real-Aug	[0.1,0.1,0.2]	0.651	0.552
CenterPP	GT-Aug	[0.1,0.1,8.0]	0.608	0.503
CenterPP	Real-Aug	[0.1,0.1,8.0]	0.639	0.558
CenterVoxel	GT-Aug	[0.075,0.075,0.2]	0.666	0.595
CenterVoxel	Real-Aug	[0.075,0.075,0.2]	0.694	0.641
CenterVoxel	GT-Aug	[0.15,0.15,0.2]	0.637	0.558
CenterVoxel	Real-Aug	[0.15,0.15,0.2]	0.658	0.596

Table 7. Model Robustness analysis. The results are evaluated on nuScenes *val* set

Robust to different detectors. The generality of Real-Aug is validated in multiple detectors with various voxel sizes. Evaluation results on nuScenes *val* set are shown in Tab. 7. In center-based models (including CenterPoint-Voxel and CenterPoint-Pillar), Real-Aug bring significant improvements (approximate 3% NDS and 5% mAP) over the baseline. The optimized performance is also valid when decreasing the voxel resolution. We test Real-Aug on SECOND-Multihead [45], which is a typical anchor-based detector, for further exploring its versatility. The proposed

realistic scene sysnthesis method for LiDAR augmentation also yields extra performance gain in anchor-based frameworks. It enhances SECOND-Multihead with a considerable increase of 3.1% NDS and 4.7% mAP.

Model	Method	Car	Pedestrian	Cyclist
SECOND	w/o GT-Aug	77.8%	44.2%	56.5%
SECOND	GT-Aug	81.4%	52.4%	65.3%
SECOND	Real-Aug	81.7%	54.0%	68.2%
PointPillar	w/o GT-Aug	75.4%	42.1%	42.9%
PointPillar	GT-Aug	77.9%	47.6%	63.2%
PointPillar	Real-Aug	78.9%	51.5%	64.1%

Table 8. Model Robustness analysis. The results are evaluated on KITTI *val* set.

Robust to different datasets. We test the adaptability of Real-Aug on KITTI dataset, in which objects and their distributions are highly divergent from that in nuScenes. Thanks to the extensive expansion of training samples' diversity, detectors trained on KITTI can greatly benefit from GT-Aug and achieve better performance. Even so, Real-Aug yields extra performance gain. For the baseline of SECOND, GT-Aug increases mAP_{3D} of moderate car, pedestrian and cyclist from 77.8%, 44.2%, 56.5% to 81.4%, 52.4%, 65.3%. Replacing GT-Aug with Real-Aug, mAP_{3D} can be further optimized to 81.7%, 54.0%, 68.2%. Similar experimental phenomena are obtained when transforming detector from SECOND to PointPillar. The mAP_{3D} of all categories with various difficulties increases from 53.5% to 62.9% if GT-Aug is used. Real-Aug further enhances the performance of PointPillar to get a mAP_{3D} of 64.8%.

Method	sampled num	NDS	mAP
GT-Aug	ref*1	0.666	0.595
GT-Aug	ref*3	0.650	0.568
GT-Aug	fix=15	0.651	0.568
Real-Aug	ref*1	0.694	0.641
Real-Aug	ref*3	0.693	0.639
Real-Aug	fix=15	0.690	0.634

Table 9. Parameter robustness analysis. ref correspond to {car:2, truck:3, construction vehicle.:7, bus:4, trailer:6, barrier:2, motorcycl':6, bicycle:6, pedestrian:2, traffic-cone:2}. (Evaluation dataset: nuScenes *val* set, model: CenterPoint-Voxel, voxel size: [0.075,0.075,0.2])

Robust to different hyper parameter choices. In GT-Aug, different magnitudes are used for sampling ground-truth objects with different categories. For each category, the magnitude means the number of objects that will be placed into a training point cloud frame. The reference design is proposed by Zhu et al. [45] and widely used in the latter published works [40, 6, 39, 1]. We simply choose three magnitude design and compare their influence on the model trained with GT-Aug and Real-Aug. In Tab. 9, ref*1 denotes the same setting as previous optimized design [45], ref*3 means we multiply the magnitude for each category in [45] by three times, fix=15 indicates we use the same setting 15 for all categories. In GT-Aug, detectors' performance degrades greatly when magnitudes for different categories deviate from optimal solution. While Real-Aug is robust to different hyper parameter choices. According to our experience, only if there is enough inserted objects at the beginning stage of the model training and follow the real-synthesis mixing up training strategy given in Sec. 3.3, the final results can almost reach the optimized level achieved by multiple attempts of hyper parameters for 10 catogories.

4.4.4 Improvement of Data Utilization

Methods	N_{train}/N_{total}	NDS	mAP
w/o GT-Aug	1	0.659	0.594
GT-Aug	1	0.666	0.595
Real-Aug	1	**0.694**	**0.641**
Real-Aug	1/2	0.681	0.622
Real-Aug	1/4	0.667	0.600
Real-Aug	1/8	0.641	0.555

Table 10. Compare the performance of detectors trained with different proportions of data. (Evaluation dataset: nuScenes *val* set, model: CenterPoint-Voxel, voxel size: [0.075,0.075,0.2]).

We compare the performance of detectors trained with different proportions of data and list them in Tab. 10. Real-Aug promotes the utilization of data to a great extent (with an approximate increase of 4 times). The detector trained with 25% data and Real-Aug performs comparably to the one trained with 100% data and GT-Aug.

4.4.5 Choice of different real-synthesis mixing up training strategy

we define two hyper parameters α and β in Sec. 3.3 to align real scene-category and scene-crowdedness relation. The divisity of synthesized scenes matters at the beginning of model training. The disturbance introduced by object insertion can be further modified by invoking our proposed real-synthesis mixing up training strategy. The wake-up signal is defined by the start iteration percentage shown in the 1st and 2nd column of Tab.11. After receiving the signal, we linearly reduce α from 1.0 to 0.0 and use the step strategy to decrease β. As shown in Tab. 11, real scene-category alignment leads to an increase of 1.0% NDS and 1.8% mAP. Combining the alignment to real scene-crowdedness relation, the highest NDS and mAP of CenterPoint-Voxel can

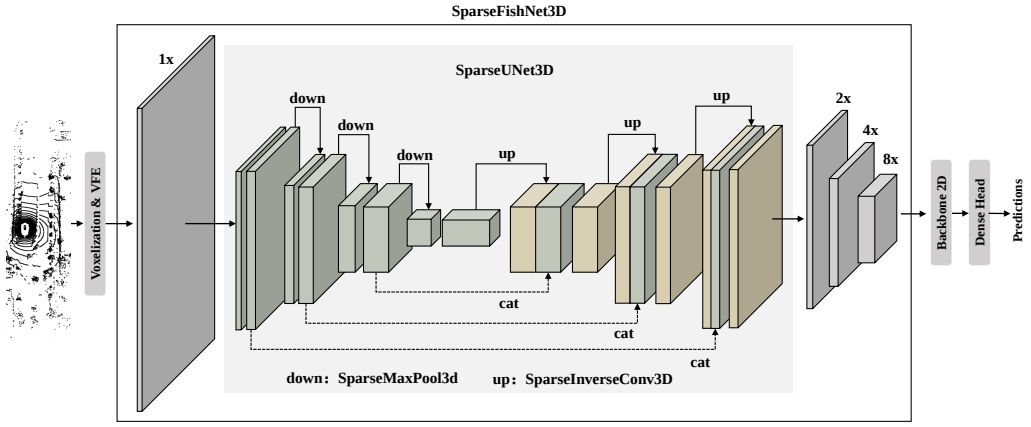

Figure 3. The framework of SparseFishNet3D. Based on the baseline backbone (SECOND), we apply a SparseUNet3D on the 2x downsample 3D feature map for stronger feature representation.

α start pct	β div steps	β div factor	NDS	mAP
-	-	-	0.678	0.611
0.75	-	-	0.688	0.629
0.75	[0.75,0.85]	2	**0.695**	**0.643**
0.75	[0.75,0.85]	4	0.694	0.641
0.75	[0.75,0.85]	8	0.693	0.640
0.80	[0.80,0.90]	2	0.693	0.639
0.80	[0.80,0.90]	4	0.695	0.640
0.80	[0.80,0.90]	8	0.695	0.642

Table 11. Choice of different real-synthesis mixing-up training strategy. (Evaluation dataset: nuScenes *val* set, model: CenterPoint-Voxel, voxel size: [0.075,0.075,0.2].)

reach 0.695 and 0.643.

4.4.6 Backbone Optimization

Backbone	Real-Aug	NDS	mAP
SECOND(Baseline)	✓	0.694	0.641
+ SC-Conv	✓	0.699	0.645
+ IoU Pred	✓	0.704	0.655
+ SparseFishNet3D	✓	0.710	0.661

Table 12. Ablations on backbone optimization. The results are evaluated on nuScenes *val* set

For enhancing the performance of detector, We use the self-calibrated convolution block (SC-Conv block) in 2D backbone and add an IoU prediction branch in the multi-task head as AFDetv2 [17]. The optimized detector achieves 1.6% NDS and 2.0% mAP increasement. We also apply a SparseUNet3D on the 2x downsample 3D feature map for stronger feature representation with larger receptive fields. Based on the above optimization methods, we achieve 0.710 NDS and 0.661 mAP on nuScenes *val* set (without any test time augmentations).

5. Conclusion

A novel data-oriented approach, Real-Aug, is proposed for LiDAR-based 3D Object Detection. It consists of a reality-conforming scene composition module and a real-synthesis mixing up training strategy. We conduct extensive experiments to verify the effectiveness of Real-Aug and achieve a state-of-the-art 0.744 NDS and 0.702 mAP on nuScenes *test* set.

References

[1] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1090–1099, 2022. 2, 6, 8

[2] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, and Qiang Xu. nuscenes: A multimodal dataset for autonomous driving. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11621–11631, 2020. 1, 3, 4, 5

[3] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Sławomir Bąk, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3d tracking and forecasting with rich maps. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8748–8757, 2019. 4

[4] Yunlu Chen, Vincent Tao Hu, Efstratios Gavves, Thomas Mensink, Pascal Mettes, Pengwan Yang, and Cees G. M.

Snoek. Pointmixup: Augmentation for point clouds. *Proceedings of the European conference on computer vision (ECCV)*, pages 330–345, 2020. 2

[5] Yukang Chen, Yanwei Li, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Focal sparse convolutional networks for 3d object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5428–5437, 2022. 1

[6] Yukang Chen, Jianhui Liu, Xiaojuan Qi, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Scaling up kernels in 3d cnns. *arXiv preprint arXiv:2206.10555*, 2022. 1, 2, 6, 8

[7] Shuyang Cheng, Zhaoqi Leng, Ekin Dogus Cubuk, Barret Zoph, Chunyan Bai, Jiquan Ngiam, Yang Song, Benjamin Caine, Vijay Vasudevan, Congcong Li, Quoc V. Le, Jonathon Shlens, and Dragomir Anguelov. Improving 3d object detection through progressive population based augmentation. *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 279–294, 2020. 1

[8] Jaeseok Choi, Yeji Song, and Nojun Kwak. Part-aware data augmentation for 3d object detection in point cloud. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3391–3397, 2021. 2

[9] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Modeling visual context is key to augmenting object detection datasets. *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 364–380, 2018. 2

[10] Lue Fan, Xuan Xiong, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Rangedet: In defense of range view for lidar-based 3d object detection. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2918–2927, 2021. 1, 6

[11] Jin Fang, Xinxin Zuo, Dingfu Zhou, Shengze Jin, Sen Wang, and Liangjun Zhang. Lidar-aug: A general rendering-based augmentation framework for 3d object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4710–4720, 2021. 1, 2

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012. 4, 5

[13] Martin Hahner, Christos Sakaridis, Mario Bijelic, Felix Heide, Fisher Yu, Dengxin Dai, and Luc Van Gool. Lidar snowfall simulation for robust 3d object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16364–16374, 2022. 1, 2

[14] Martin Hahner, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Fog simulation on real lidar point clouds for 3d object detection in adverse weather. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15283–15292, 2021. 1, 2

[15] Frederik Hasecke, Martin Alsfasser, and Anton Kummert. What can be seen is what you get: Structure aware point cloud augmentation. *IEEE Intelligent Vehicles Symposium (IV)*, pages 594–599, 2022. 1, 2

[16] Jordan S. K. Hu and Steven L. Waslander. Pattern-aware data augmentation for lidar 3d object detection. *IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 2703–2710, 2021. 2

[17] Yihan Hu, Zhuangzhuang Ding, Runzhou Ge, Wenxin Shao, Li Huang, Kun Li, and Qiang Liu. Afdetv2: Rethinking the necessity of the second stage for object detection from point clouds. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 36(1):969–979, 2022. 2, 6, 9

[18] Dihe Huang, Ying Chen, Yikang Ding, Jinli Liao, Jianlin Liu, Kai Wu, Qiang Nie, Yong Liu, Chengjie Wang, and Zhiheng Li. Rethinking dimensionality reduction in grid-based 3d object detection. *arXiv preprint arXiv:2209.09464*, 2022. 2, 6

[19] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, and Jiong Yang. Pointpillars: Fast encoders for object detection from point clouds. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12697–12705, 2019. 1, 2, 5

[20] Dogyoon Lee, Jaeha Lee, Junhyeop Lee, Hyeongmin Lee, Minhyeok Lee, Sungmin Woo, and Sangyoun Lee. Regularization strategy for point cloud via rigidly mixed sample. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15900–15909, 2021. 2

[21] Seungjae Lee, Hyungtae Lim, and Hyun Myung. Patchwork++: Fast and robust ground segmentation solving partial under-segmentation using 3d point cloud. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13276–13283, 2022. 3, 6

[22] Hyungtae Lim, Oh Minho, and Hyun Myung. Patchwork: Concentric zone-based region-wise ground segmentation with ground likelihood estimation using a 3d lidar sensor. *IEEE Robotics and Automation Letters*, 6(4):6458–6465, 2021. 3, 6

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021. 3

[24] Sivabalan Manivasagam, Shenlong Wang, Kelvin Wong, Wenyuan Zeng, Mikita Sazanovich, Shuhan Tan, Bin Yang, Wei-Chiu Ma, and Raquel Urtasun. Lidarsim: Realistic lidar simulation by leveraging the real world. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11167–11176, 2020. 1, 3

[25] Jiageng Mao, Yujing Xue, Minzhe Niu, Haoyue Bai, Jiashi Feng, Xiaodan Liang, Hang Xu, and Chunjing Xu. Voxel transformer for 3d object detection. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3164–3173, 2021. 1

[26] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 3

[27] Alexey Nekrasov, Jonas Schult, Or Litany, Bastian Leibe, and Francis Engelmann. Mix3d: Out-of-context data augmentation for 3d scenes. *IEEE International Conference on 3D Vision (3DV)*, pages 116–125, 2021. 2

[28] https://github.com/open-mmlab/OpenPCDet. 5

[29] https://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d. 6

[30] Guangsheng Shi, Ruifeng Li, and Chao Ma. Pillarnet: Real-time and high-performance pillar-based 3d object detection. *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 35–52, 2022. 6

[31] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. *International Journal of Computer Vision*, 131(2):531–551, 2023. 1

[32] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurélien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Sheng Zhao, Shuyang Cheng, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2446–2454, 2020. 1, 3, 4

[33] Tianweiy and Abyssaledge. Discussion about database sampler effectiveness in the official implement of centerpoint. https://github.com/tianweiy/CenterPoint/issues/250, 2022. 1

[34] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *International Conference on Machine Learning (ICML)*, pages 10347–10357, 2021. 3

[35] Shashank Tripathi, Siddhartha Chandra, Amit Agrawal, Ambrish Tyagi, James M. Rehg, and Visesh Chari. Learning to generate synthetic data via compositing. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 461–470, 2019. 2

[36] Sourabh Vora, Alex H. Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4604–4612, 2020. 2

[37] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. Pointaugmenting: Cross-modal augmentation for 3d object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11794–11803, 2021. 2

[38] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 1, 2, 3, 4, 5

[39] Dongqiangzi Ye, Zixiang Zhou, Weijia Chen, Yufei Xie, Yu Wang, Panqu Wang, and Hassan Foroosh. Lidarmultinet: Towards a unified multi-task network for lidar perception. *arXiv preprint arXiv:2209.09385*, 2022. 2, 6, 8

[40] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11784–11793, 2021. 1, 2, 3, 5, 6, 8, 12

[41] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6023–6032, 2019. 3

[42] Lingzhi Zhang, Tarmily Wen, Jie Min, Jiancong Wang, David Han, and Jianbo Shi. Learning object placement by inpainting for compositional data augmentation. *Proceedings of the European conference on computer vision (ECCV)*, pages 566–581, 2020. 2

[43] Wu Zheng, Weiliang Tang, Li Jiang, and Chi-Wing Fu. Se-ssd: Self-ensembling single-stage object detector from point cloud. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14494–14503, 2021. 1, 2

[44] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4490–4499, 2018. 1

[45] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint arXiv:1908.09492*, 2019. 2, 5, 6, 7, 8

A. Object Information

In order to help analyze the difficulty of detection task in nuScenes and KITTI dataset, we summarize the basic information of objects with different categories. The results are shown in Tab.13 and Tab.14.

For a specific category, $\bar{l}$, $\bar{w}$, $\bar{h}$ denotes the average length, width and height of objects. $\bar{D}_{pts}$ denotes the average LiDAR points number inside each voxel of the object's 3D bounding box. It can be calculated by Eq. 8:

$$\bar{\mathcal{D}}_{pts} = \frac{\bar{N}_{pts}}{\bar{N}_{voxel}} = \frac{\bar{N}_{pts}}{(\bar{l}/v_D) \times (\bar{w}/v_W) \times (\bar{h}/v_H)} \tag{8}$$

where $\bar{N}_{pts}$, $\bar{N}_{voxel}$ denotes the average number of points, voxels inside object's 3D bounding box. v_D, v_W, v_H are the voxel size defined in detectors. In Tab. 13, v_D, v_W, v_H are set as 0.075, 0.075, 0.2. $\bar{N}_{pts}$ is calculated according to the densified point cloud with 10 LiDAR sweeps. In Tab. 14, v_D, v_W, v_H are set as 0.05, 0.05, 0.1. $\mathcal{R}_{frame}$ denotes the percentage of frames that contain the corresponding objects. The sum of $\mathcal{R}_{frame}$ for all classes is not equal to 1.0 because one frame may contain objects with different categories. $\mathcal{R}_{obj}$ denotes the percentage of objects throughout the whole *object bank*.

A.1. nuScenes

	$\bar{l}$	$\bar{w}$	$\bar{h}$	$\bar{\mathcal{D}}_{pts}$	$\mathcal{R}_{frame}$	$\mathcal{R}_{obj}$
Car	4.634	1.954	1.734	0.065	97.73%	42.43%
Truck	6.992	2.517	2.870	0.044	70.26%	8.29%
C.V.	6.454	2.857	3.216	0.018	23.33%	1.41%
Bus	11.090	2.933	3.464	0.029	31.90%	1.60%
Trailer	12.283	2.904	3.875	0.017	24.67%	2.40%
Barrier	0.503	2.524	0.983	0.550	31.71%	13.65%
Mot.	2.102	0.769	1.472	0.172	21.60%	1.16%
Byc.	1.706	0.601	1.294	0.116	20.85%	1.07%
Ped.	0.728	0.668	1.770	0.127	79.19%	20.11%
T.C.	0.415	0.408	1.069	0.534	39.63%	7.88%

Table 13. Information of objects in nuScenes *trainval* set.

According to the performance of CenterPoint-Voxel trained with GT-Aug, which is shown in the 2nd line of Tab. 6, we divide all the ten categories defined in nuScenes detection task into four groups: (1) car, pedestrian (with AP above 0.8); (2) bus, barrier, traffic cone (with AP ranging from 0.6 to 0.8); (3) truck, motorcycle, bicycle (with AP ranging from 0.4 to 0.6); (4) trailer, construction vehicle (with AP lower than 0.4).

Car and pedestrian are two categories with most abundant data in nuScenes dataset. Car exists in 97.73% frames and pedestrian exists in 79.19% frames. Their $\mathcal{R}_{obj}$ also ranks top two throughout the whole *object band*. As a result, detectors perform well on car and pedestrian. Bus, barrier, traffic cone are three categories with clear and simple structural characteristics. The morphology consistency in different scenarios reduce the difficulty for detector to distinguish them from other objects and background points. Truck, motorcycle and bicycle, whose data is not as rich as pedestrian and meanwhile with more complex and volatile morphology than bus, barrier and traffic cone, are more difficult to be detected. For trailer and construction vehicle, the lowest points density in each voxel introduces much confusion for detectors to recognize them from background points.

Real-Aug, which contains a a reality-conforming scene composition module to handle the details of the composition and a real-synthesis mixing up training strategy to gradually adapt the data distribution from synthetic data to real one, can greatly alleviate negative effects of existing synthesis-based augmentation methods. Detectors trained with Real-Aug present remarkable performance optimization, especially on motorcycle, bicycle, trailer and construction vehicle.

A.2. KITTI

	$\bar{l}$	$\bar{w}$	$\bar{h}$	$\bar{\mathcal{D}}_{pts}$	$\mathcal{R}_{frame}$	$\mathcal{R}_{obj}$
Car	3.884	1.629	1.526	0.011	89.35%	82.46%
Ped.	0.842	0.660	1.761	0.048	23.78%	12.87%
Cyc.	1.764	0.597	1.737	0.023	15.25%	4.67%

Table 14. Information of objects in KITTI *trainval* set.

According to the performance of SECOND, which is shown in Tab. 8, we divide the three categories defined in KITTI detection task into two groups: (1) car; (2) pedestrian and cyclist. Comparing to car, pedestrian and cyclist are lack in data quantity and meanwhile possess high morphology complexity. As a result, the mAP_{3D} of car are much higher than that of the other two categories. The effectiveness of Real-Aug is also validated in KITTI dataset. According to Tab. 8, the mAP_{3D} of moderate objects that is inferenced by SECOND can be further increased from 66.4% to 68.0% when replacing GT-Aug with Real-Aug.

B. Test Time Augmentation

Throughout the inference process of CenterPoint-Voxel on nuScenes dataset, we use two test time augmentation (TTA), including double flip and point-cloud rotation along the yaw axis, to improve the detector's final detection performance. The yaw angles are set the same as that in [40], which are $[0°, \pm 6.25°, \pm 12.5°, \pm 25°]$. The inference re-

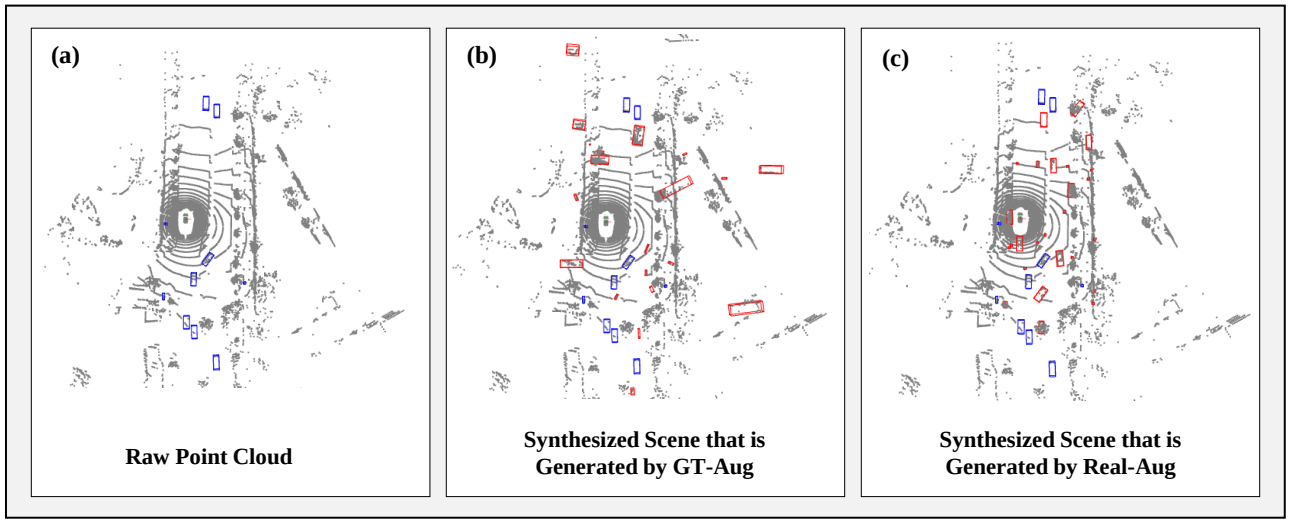

Figure 4. Visualization of synthesized scene generated by GT-Aug (b) and Real-Aug (c). (Boxes with blue lines: the original boxes in raw point clouds; Boxes with red lines: the inserted boxes.)

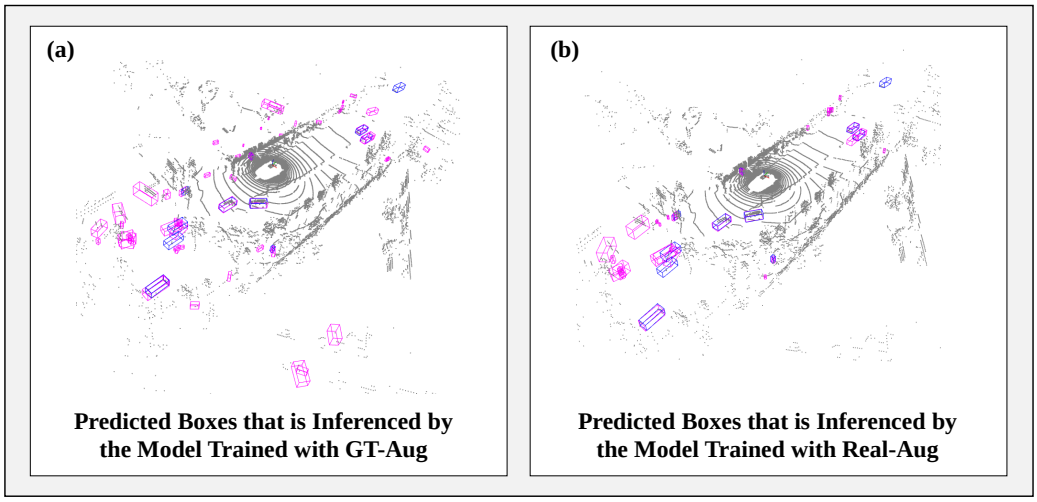

Figure 5. Visualization of predicted boxes that is inferenced by the model trained with GT-Aug (a) and Real-Aug (b). (Boxes with blue lines: ground-truth boxes; Boxes with pink lines: predicted boxes.)

Model	double flip	rotation	NDS	mAP
CenterVoxel			0.694	0.641
CenterVoxel	✓	✓	0.715	0.667
SparseFishNet3D			0.710	0.661
SparseFishNet3D	✓	✓	0.731	0.689

Table 15. Effects of TTA (double flip and rotation). (Evaluation dataset: nuScenes *val* set, model:CenterPoint-Voxel, voxel size: [0.075,0.075,0.2]).

sults of CenterPoint-Voxel on nuScenes *val* set are shown in Tab. 15. With TTA, the NDS and mAP of the baseline model (CenterPoint-Voxel) raise from 0.694 and 0.641 to 0.715 and 0.667 respectively. The SparseFishNet3D which is decribed in Sec. 4.4.6 further enhances the detection performance with a considerable 0.731 NDS and 0.689 mAP.

In KITTI dataset, objects outside the front view are not annotated. Throughout the inference process of SECOND, y-flip test time augmentation is applied. The evaluated results of SECOND on KITTI *val* set are shown in Tab. 16. There is an increase of 0.1%, 3.3%, 0.4% AP_{3D} for moder-

Model	y-flip	Car	Pedestrian	Cyclist
SECOND		81.7%	54.0%	68.2%
SECOND	✓	81.8%	57.3%	68.6%

Table 16. Effects of TTA (y-flip). (Evaluation dataset: KITTI *val* set, model:SECOND, voxel size: [0.05,0.05,0.1]).

ate car, pedestrian and cyclist.

C. Visualization

The augmented point clouds generated by GT-Aug and Real-Aug are visualized in Fig. 4. GT-Aug introduces many non-existing LiDAR scans patterns into the point clouds. The inserted objects, which locate at physically unreasonable place and move towards inappropriate direction, will hinder detectors from learning effective features. In this paper, a reality-conforming scene composition module is proposed to deal with the above mentioned problems. It handles the details of synthesis operation and maintains the authenticity of the composite scene as much as possible. The real-synthesis mixing up training strategy can further alleviate the negative influence introduced by synthesis-based LiDAR augmentation. The predicted boxes that is inferenced by the model trained with GT-Aug and Real-Aug are visualized in Fig. 5. The predicted boxes with scores lower than 0.1 are filtered out. It is clear that replacing GT-Aug with Real-Aug can effectively reduce false positives.

D. Limitations and Future Work

In addition to applying our Real-Aug on nuScenes and KITTI dataset, we will also extend our methods on Waymo dataset in the future.