Resolving Class Imbalance for LiDAR-based Object Detector by Dynamic Weight Average and Contextual Ground Truth Sampling

Daeun Lee[1] and Jinkyu Kim[2]
[1]Statistics and [2]Computer Science and Engineering, Korea University, Seoul 02841, Korea
{goodgpt, jinkyukim}@korea.ac.kr

Abstract

An autonomous driving system requires a 3D object detector, which must perceive all present road agents reliably to navigate an environment safely. However, real-world driving datasets often suffer from the problem of data imbalance, which causes difficulties in training a model that works well across all classes, resulting in an undesired imbalanced sub-optimal performance. In this work, we propose a method to address this data imbalance problem. Our method consists of two main components: (i) a LiDAR-based 3D object detector with per-class multiple detection heads where losses from each head are modified by dynamic weight average to be balanced. (ii) Contextual ground truth (GT) sampling, where we improve conventional GT sampling techniques by leveraging semantic information to augment point cloud with sampled ground truth GT objects. Our experiment with KITTI and nuScenes datasets confirms our proposed method's effectiveness in dealing with the data imbalance problem, producing better detection accuracy compared to existing approaches.

1. Introduction

LiDAR-based detectors have been widely adopted in the autonomous driving system for capturing 3D scene perception and understanding [15, 20, 5]. Such an autonomous driving system must detect all possible other road agents (or objects) to navigate an environment safely. Thus, a reliable LiDAR-based detector requires dealing equally with different road agents (or objects), e.g., cars, cyclists, barriers, or construction vehicles.

However, real-world driving datasets (e.g., KITTI [9] and nuScenes [1]) suffer from the problem of imbalance where a dataset contains unequal (or severely skewed) class distribution. As shown in Figure 1, objects such as cars (42.63%) have a higher percentage compared to the percentage of other classes, such as bicycles (1.03%), motorcycles (1.11%), or construction vehicles (1.39%). Similarly, in the KITTI dataset, cars (82.99%) have the majority of instances, while pedestrians (12.76%) or cyclists (4.24%) are underrepresented. Such data imbalance would cause difficulties in training a 3D object detector that reliably works well across all different classes, resulting in an undesired imbalanced quality.

Multi-task learning techniques have been applied to address this data imbalance problem by viewing multi-class joint detection as multi-task learning [17, 14]. In this work, we explore applying such multi-task learning techniques to address the data imbalance problem in the LiDAR-based 3D object detection task. Specifically, we focus on answering two key questions: (i) constructing multi-task network architecture and (ii) balancing feature sharing across different tasks. For (i), we use per-class multiple detection heads instead of a single head. Each detection head is encouraged to learn class-specific features while sharing a backbone, which is trained to extract universal features. For (ii), we explore applying existing multi-task loss balancing techniques to improve the overall performance of different detection heads. Specifically, we apply Dynamic Weight Average (DWA, [16]) that tunes gradients for different object categories based on the rate of loss changes for each head to learn average task weighting over time. We empirically observe that combining multi-headed architecture and gradient balancing techniques significantly improves detection accuracy.

Another story is data augmentation, which can make class distribution smoother by making the model sees rare classes more often during training. Conventionally, ground truth (GT) sampling [28] has been widely used. GT sampling collects all ground truth points inside the labeled bounding box into a database, and some of them are randomly introduced to the current training frame via concatenation. However, this does not consider where to place these objects. We, in fact, observe ground truth points are often introduced in a random position where that object is rarely observed in the real world. Thus, we propose *contextual* GT sampling that leverages semantic scene information to present ground truth points in a more natural position, e.g.,

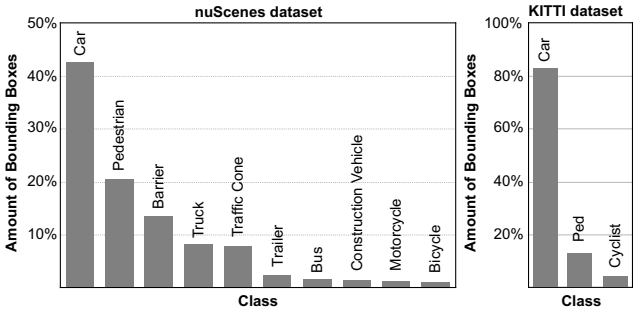

Figure 1. Class distributions for two 3D object detection datasets: nuScenes [1] (left) and KITTI [9] (right).

a sidewalk for pedestrians. Our experiment shows that our contextual GT sampling provides extra performance gain, especially for minor classes.

Our approach is mostly close to Zhu *et al*. [36] (CBGS) in that they also use multiple detection heads and data augmentation techniques, i.e., GT sampling [28]. However, our work differs from it as follows: (i) we explore using multi-task learning techniques, including multiple detection heads with loss balancing techniques, to improve overall detection performance across all categories. CBGS focused on utilizing multi-headed architecture with a uniform scaling, which minimizes a uniformly weighted sum and does not consider dynamically modifying weights like ours. (ii) we propose contextual GT sampling, which addresses issues with conventional GT sampling and results in better detection accuracy.

We summarize our contributions as follows:

- Inspired by multi-task learning, we propose a multi-headed LiDAR-based 3D object detector where losses for each head are balanced by dynamic weight average (DWA).
- Combined with multi-headed architecture, we propose contextual ground truth sampling, which improves conventional ground truth (GT) sampling by leveraging semantic scene information to introduce GT objects in a more realistic position.
- We conduct various experiments to demonstrate the effectiveness of our proposed approach with widely-used public datasets: KITTI and nuScenes. Our experiments show that multi-task learning techniques combined with our contextual GT sampling significantly improve the overall detection performance, especially for minor classes.

2. Related Work

2.1. 3D Object Detection

A landmark work in the LiDAR-based 3D object detection is VoxelNet [35], an end-to-end trainable model that first voxelized a point cloud, and each equally spaced voxel is encoded as a descriptive volumetric representation. Given these features, conventional 2D convolutions are used to generate and regress its region proposals. Yan *et al*. [28] used sparse 3D convolutions to accelerate heavy computations of earlier LiDAR-based works. PointPillars [15] is another landmark work that speeds up the encoding of 3D volumetric representation by dividing the 3D space into pillars (instead of voxels). A more sophisticated architecture is also used to achieve better detection results. PointRCNN [22] used a two-stage architecture to refine the initial 3D bounding box proposals. Part-A2 [23] focuses on leveraging intra-object parts for better results. PV-RCNN [20] and PV-RCNN++ [21] simultaneously process coarse-grained voxels and the raw point cloud. Recently, CenterPoint [31] applied a key-point detector that predicts the geometric center of objects. Similarly, Voxel RCNN [5] used coarse voxel granularity to reduce the computation cost, retaining the overall detection performance. In this work, we focus on improving data imbalance problems in LiDAR-based object detection. Thus, we do not claim a novel 3D object detector; rather, we rely on existing landmark work PointPillars [15], PV-RCNN [20], and Voxel RCNN [5] to demonstrate the effectiveness of our proposed approach. Note that, ideally, our approach is applicable to others as well.

Lidar Points Augmentations. Data augmentation has been widely applied to LiDAR-based 3D object detection for various reasons: (i) improving point cloud quality by upsampling a low-density point cloud [32, 30] or by point cloud completion for occluded regions [33, 29, 2, 27]. (ii) Improving the robustness of object detection by global and local augmentations. Choi *et al*. [4] randomly augmented sub-partitions of GT objects (e.g., dropping points in a certain sub-partition) [4]. Zheng *et al*. [34] divided each ground truth object into six (inward facing) pyramids, then augmented them with random dropout, swap, and sparsifying operations. (iii) Improving generalization power by augmenting clear weather point clouds with adverse conditions via physical modelings, such as fog [13] or snowfall [12]. (iv) Augmenting LiDAR-based features with other modalities, such as images [26, 25]. (v) Smoothing class distribution by sampling ground truth objects from the (offline) database and introducing them to the current scene (GT sampling, [28]). In this work, similar to (v), we focus on smoothing the density of each class to address the data imbalance problem (i.e., improving the detection accuracy of rare objects while maintaining that of common objects). Thus, we start with GT sampling [28] as a baseline.

2.2. Gradient Balancing

Multitask learning has been widely explored to share features across different tasks while making task-specific mul-

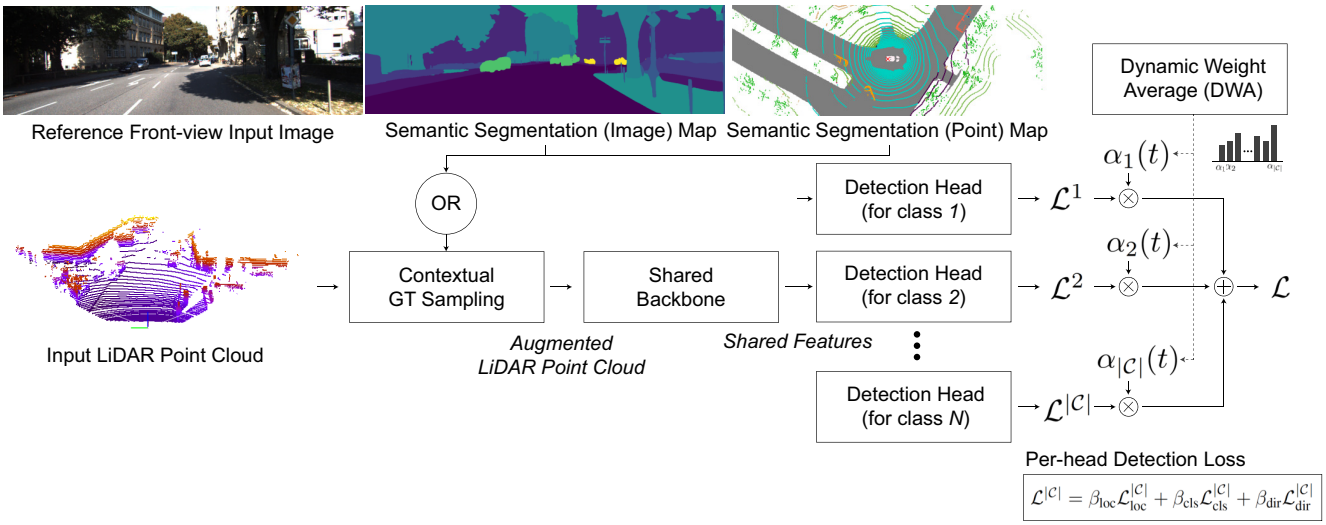

Figure 2. An overview of our proposed method to address the data imbalance problem in the LiDAR-based 3D object detection task. Our model consists of two main parts: (1) Per-class multi-headed architecture where detection losses $\mathcal{L}^c$ for $c \in \mathcal{C}$ for each head are balanced by dynamic weight average (DWA). (2) Contextual ground truth (GT) sampling, which is built upon conventional GT sampling and improves it by leveraging semantic scene information (either semantic segmentation image map or semantic point map) to place ground truth points in a more realistic position.

tiple predictions, such as for multi-domain image classification [19], post estimation and action recognition [10], or depth estimation and semantic segmentation [7, 17]. Key questions in multitask learning are (i) constructing multi-task network architectures and (ii) balancing feature sharing across different tasks. To address the latter, studies reported that multitask loss balancing techniques improve the overall performance for different tasks [17, 14]. Kendall *et al*. [14] modified the loss function based on task uncertainty, and GradNorm [3] dynamically tuned gradient magnitudes and showed it improves accuracy and reduces overfitting across different tasks. Dynamic Task Prioritisation [11] prioritized difficult tasks based on performance metrics. Dynamic Weight Average (DWA, [16]) is also proposed to use the rate of loss changes for each task to learn average task weighting over time. In this work, we explore applying recent multitask learning techniques to tune gradients for different object categories and reduce the data imbalance problem.

3. Method

3.1. Gradient Balanced Per-class Detection Heads

Multi-heads Architecture. We first adopt a multi-task learning (MTL) strategy, which aims to learn multiple tasks jointly by leveraging the shared knowledge of all the tasks at hand. MTL is effective in reducing the data sparsity problem where the number of labeled examples for each task is insufficient to optimize a model. This is because MTL can aggregate all labeled data and utilize more data from different tasks to obtain a more accurate learner with generalizable representations for multiple tasks. As reported in existing works [36], MTL is also effective in improving the performance of multi-category joint detectors.

As shown in Figure 2, we apply the above-mentioned multi-task learning strategy by utilizing multiple object detection heads for each category with a shared encoder (i.e., backbone). This multi-headed architecture prevents the risk of overfitting in a particular dominant task (e.g., a single-head model would more overfit in detecting common objects than the rare). It also reduces the data imbalance problem when combined with data augmentation techniques (which we will explain in Section 3.2).

Point Cloud Encoder. Our model is built upon a seminar work, PointPillars [15], though our model is easily applicable to other LiDAR-based 3D object detectors. Following PointPillars, we encode a point set $\mathcal{P}$ into an evenly-spaced grid of M pillars with x-y coordinates. Points in each pillars are augmented with a tuple $(x_c, y_c, z_c, x_p, y_p)$, where x_c, y_c, and z_c are distances to the mean of all points in the pillar, and x_p and y_p are offsets from the pillar center. We then apply a simplified PointNet [18] architecture to encode each point and aggregate features into a single feature vector per pillar by a max operation. The resulting $M \times N$ feature map is then processed through a backbone, reshaping it to $W \times H$. Detection heads then share this resulting feature map for the final verdict.

Detection Heads. Following PointPillars [15], we use the Single Shot Detector (SSD) setup as the detection head, matching predictions to the ground truth using 2D Intersec-

Table 1. Object categories and their associated semantic labels (based on KITTI [9] and nuScenes [1] dataset) for contextual GT sampling. *Abbr.* C.V.: Construction Vehicle, T.C.: Traffic Cone.

Dataset	Object Category	Associated Semantic Labels
NuScenes [1]	Pedestrians	Sidewalk
	Car, Truck, Bus, Trailer, C.V.	Drivable Surface
	Motorcycle, Bicycle, Barrier, T.C.	Sidewalk, Drivable Surface
KITTI [9]	Pedestrian	Sidewalk
	Car	road
	Cyclist	sidewalk, road

tion of Union (IoU). Each detection head is trained with the following three types of losses: (i) localization loss $\mathcal{L}_{\text{loc}}$, (ii) object classification loss $\mathcal{L}_{\text{cls}}$, and (iii) heading loss $\mathcal{L}_{\text{dir}}$. The total loss is as follows:

$$\mathcal{L} = \frac{1}{N_{\text{pos}}} \sum_{c \in \mathcal{C}} \alpha_c(t)(\beta_{\text{loc}}\mathcal{L}^c_{\text{loc}} + \beta_{\text{cls}}\mathcal{L}^c_{\text{cls}} + \beta_{\text{dir}}\mathcal{L}^c_{\text{dir}}) \quad (1)$$

we accumulate all losses from $|\mathcal{C}|$ detection heads where $\mathcal{C}$ is a set of categories. The number of positive anchors is denoted by N_{pos} and hyperparameters β_{loc}, β_{loc}, and β_{loc} are set by default as 2, 1, and 0.2, respectively. Note that we use $\alpha_c(t)$ as a loss weight determined at each timestep t to tune the loss and fix data imbalances in gradient norms, which we will explain in detail in the next section. In Figure 2, we describe our LiDAR-based object detector with per-class detection heads with gradient balancing.

Balancing Gradients. To determine the balancing weight $\alpha_c(t)$ at each timestep t for a detection head c, we use a technique called dynamic weight average (DWA, [16]). We use DWA to compute weights for each head c as follows:

$$\alpha_c(t) = \frac{|\mathcal{C}|\exp(w_c(t-1)/T)}{\sum_c \exp(w_c(t-1)/T)} \quad (2)$$

where $w_c(t)$ is defined as the relative descending rate at timestep (or iteration) t and defined as follows: $w_c(t-1) = \mathcal{L}_c(t-1)/\mathcal{L}_c(t-2)$ where $\mathcal{L}_c(t)$ is the averaged loss value from a detection head c. We use a temperature T to control the strength of gradient balancing. As reported in [16], we use the loss $\mathcal{L}_c(t)$ averaged over several iterations to reduce unstable training due to uncertainties from stochastic gradient descent and random training data selection.

3.2. Context-aware LiDAR Points Augmentation

Data imbalance problem is often observed in various perception datasets for autonomous driving. Common objects, such as cars, generally are more numerous (by a large margin) than rare object classes, such as construction vehicles or traffic cones. Such data imbalance significantly limits balancing the network's final performance per class. One way to solve this problem is via a data augmentation approach, which can make class distribution smoother by making the model see rare classes more often during training.

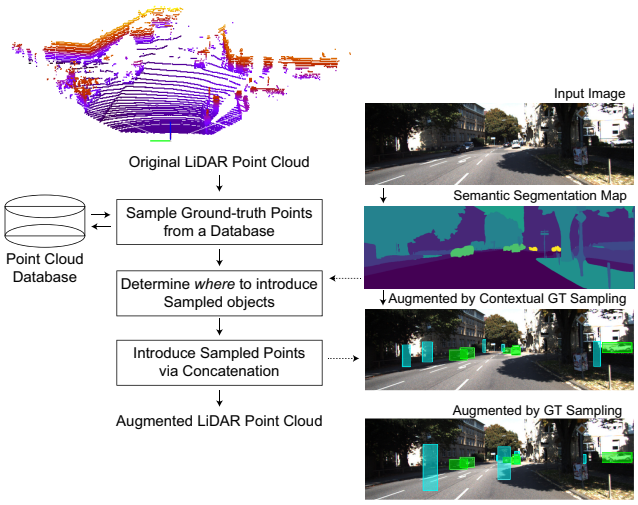

Figure 3. An overview of our proposed contextual GT Sampling. We first sample some of the ground truth LiDAR points from a database. Based on a semantic segmentation map, we then identify potential regions where objects can be plausibly observed (e.g., pedestrians on the sidewalk). Conventional GT sampling does not consider this.

Sampling Ground Truths from Database (GT Sampling). A common practice in LiDAR-based data augmentation is sampling ground truths from the database called GT sampling [28]. All ground truth points inside the labeled bounding box (with their labels) are collected in an offline database. Some ground truth points are randomly chosen from this database during training and placed into the current frame of point clouds via concatenation, simulating objects from different frames or environments. Thus, the average density of rare classes can be improved.

Contextual GT Sampling. Though a simple filtering rule is used to ensure sampled objects do not collide with other objects, it does not consider *where* to place these objects. For example, as shown in Figure 4, ground truth points of pedestrians are introduced to a random position where pedestrians are rarely observed (e.g., on the road intended for vehicles). To address this, we advocate using prior semantic information to introduce objects in a more realistic position (e.g., on a sidewalk for pedestrians).

Formerly, given a 2D semantic segmentation map, we first identify potential image regions where objects are plausibly or commonly observed. In Table 1, we summarize objects and their associated semantic labels (to be placed). Based on a given camera's known intrinsic and extrinsic

parameters, the 3D geometric center points of a given object bounding box are first projected into the ground (set z as 0) and then projected into a 2D image plane to determine semantic information of the region. Finally, objects introduced to non-associated regions (e.g., pedestrians on a roadway) are removed by a filtering criterion. We explain an overview of our contextual GT sampling process in Figure 3. Note that the contextual GT sampling can be applied with LiDAR-based segmentation maps where we collect top-k closest points and their semantic labels followed by a k-NN classifier.

4. Experiments

4.1. Setup

Implementation Details. Our model is built upon PointPillars [15] architecture, and we follow its default setting for training our model. Our implementation is based on an open source project for LiDAR-based 3D object detection called OpenPCDet [24], which supports multiple LiDAR-based 3D perception models, including PointPillars [15], PV-RCNN [20], and Voxel RCNN [5]. Thus, we believe that our proposed regularizing components could be easily applied to other LiDAR-based perception models and ensure reproduction. Our model is trained end-to-end using Adam optimizer with the learning rate 0.003. The whole model is trained for 80 epochs on 4 NVIDIA GeForce RTX 3090 GPUs. Since our data augmentation strategy and gradient balancing technique are turned off during inference, the inference time would remain the same or slightly larger than that of PointPillars (due to utilizing multi-headed architecture).

Evaluation Details. For evaluation, we use the widely-used publicly available KITTI [9] 3D Object Detection dataset, which provides 7,481 training images and 7,518 test images along with LiDAR point clouds. Overall, 80,256 objects are labeled, and (as typically chosen) we focus on three types of objects: cars, pedestrians, and cyclists. Note that our model is based only on LiDAR point clouds during inference, and we use images for better qualitative analysis. Further, we use a large-scale dataset called nuScenes [1], which provides over 1500 hours of driving data collected from four different major cities. Our model evaluation is done on ten classes: i.e., car, truck, construction vehicle (CV), bus, trailer, barrier, motorcycle, bicycle, pedestrian, and traffic cone (TC).

4.2. Quantitative Analysis

Evaluation with KITTI Dataset. We first analyze our proposed model with the publicly available KITTI [9] 3D object detection dataset. As shown in Table 2, starting from the baseline (we use PointPillars [15]), we compare the 3D object detection performance (in mAP) with and without our two main components: (i) per-class multiple detection heads along with gradient balancing techniques and (ii) contextual GT sampling. We compare our model with CBGS (cross-balanced grouping and sampling, [36]), which similarly aims to reduce the negative effect of class imbalance problems in the perception task.

We observe in Table 2 that our model generally provides a performance improvement in all classes and metrics and generally outperforms the alternative approach. Such an improvement is significant in detecting cyclists, which rarely appear in the dataset (734 out of 17,298 labeled training objects). This may confirm that our proposed components are effective in improving the detection performance of rarely observed objects in such an imbalanced dataset. We also observe that our gradient-balanced multi-headed architecture and contextual GT sampling significantly improves the overall object detection performance, especially for rarely-observed object classes (compare cyclist vs. car and pedestrians)

Evaluation with a large-scale nuScenes Dataset. To further demonstrate the effectiveness of our proposed approach, we evaluate with a large-scale nuScenes dataset, which is more challenging for perception mainly due to its volume and data imbalances across different classes. In Table 3, we compare 3D object detection performance (in 3D and BEV mAPs) for all ten different object classes: (in the sorted order by their numbers) car, truck, bus, trailer, construction vehicle, pedestrian, motorcycle, bicycle, traffic cone, and barrier. Similarly, we compare our model with CBGS [36]. As we observe in Table 3, our model generally outperforms alternatives, and the performance boost is notable in minor classes, such as buses, trailers, construction vehicles, etc. This further confirms that our proposed model is effective in dealing with rarely-observable objects, and such improvement is larger than the existing approach, CBGS [36].

4.3. Ablation Study

Effect of Contextual GT Sampling for LiDAR Data Augmentation. We further evaluate the effect of using the contextual sampling with existing LiDAR-based 3D object detection models: PointPillars [15], PV-RCNN [20], and Voxel RCNN [5]. As we observe in Table 4, perception performance (in terms of 3D mAP and BEV mAP) generally improves by changing conventional GT sampling with our contextual GT sampling. A similar pattern of improvements is observed in all models, and a larger effect is obtained in minor classes, such as pedestrians and cyclists. Note that we only consider cars for Voxel RCNN [5] as its original architecture is focusing only on detecting cars.

In Figure 4, we provide an example of conventional GT sampling and our proposed contextual GT sampling for

Table 2. 3D object detection performance (in mAP) on the publicly available KITTI [9] validation dataset. We also report counts (in %) for each class to determine the data imbalance amount. In higher IoU threshold setting: Car (0.7), Pedestrian (0.5), and Cyclist (0.5), In lower IoU threshold setting: Car (0.5), Pedestrian (0.25), and Cyclist (0.25).

Model	Higher IoU threshold setting								Lower IoU threshold setting							
	Car (83.00%)		Pedestrian (12.76%)		Cyclist (4.24%)		Avg. (100.00%)		Car (83.00%)		Pedestrian (12.76%)		Cyclist) (4.24%)		Avg. (100.00%)	
	3D↑	BEV↑	3D↑	BEV↑	3D↑	BEV↑	3D↑	BEV↑	3D↑	BEV↑	3D↑	BEV↑	3D↑	BEV↑	3D↑	BEV↑
A. PointPillars [15]	78.04	87.49	49.40	55.78	63.95	68.97	63.80	70.75	**95.62**	94.49	69.68	69.86	73.78	73.78	79.69	79.38
B. A + CBGS [36]	77.78	87.47	**51.06**	**57.40**	65.53	69.40	64.79	71.42	94.33	94.51	72.20	72.57	72.54	72.56	79.69	79.88
C. A + Ours	78.37	**89.28**	**51.06**	56.92	**68.79**	**72.44**	**66.07**	**72.88**	94.88	**96.29**	**73.24**	**73.36**	**76.58**	**76.58**	**81.57**	**82.08**
D. C w/o contextual GT sampling	**78.77**	88.28	50.50	55.86	65.31	69.24	64.86	71.13	94.42	94.61	70.64	70.88	72.45	72.81	79.17	79.43

Table 3. 3D object detection performance (in mAP) on the nuScenes [8] validation set. We also report counts (in %) for each class to determine the data imbalance amount. *Abbr.* C.V.: Construction Vehicle, Ped: Pedestrian, Moto: Motorcycle, T.C.: Traffic Cone.

Model	Car	Ped	Barrier	Truck	T.C.	Trailer	Bus	C.V.	Motor	Bicycle	Avg.
	(42.64%)	(20.31%)	(13.49%)	(8.19%)	(7.90%)	(2.41%)	(1.54%)	(1.39%)	(1.11%)	(1.03%)	(100.00%)
A. PointPillars + CBGS	80.8	71.9	47.8	49.2	46.9	34.2	62.4	12.1	30.9	4.8	44.1
B. PointPillars + Ours	**82.1**	71.9	**54.5**	**53.8**	**50.1**	**39.1**	**67.0**	**16.3**	**40.2**	**9.4**	**48.4**
	(1.3↑)	(0.0)	(6.7↑)	(4.6↑)	(3.2↑)	(4.9↑)	(4.6↑)	(4.2↑)	(9.3↑)	(4.6↑)	(4.3↑)
C. B + w/o contextual GT sampling	81.0	**72.3**	50.2	49.0	45.2	34.3	63.4	10.7	32.9	6.9	44.6

pedestrians. A probability occupancy grid is computed for augmented LiDAR points to be copied (from a GT LiDAR point cloud database) and pasted (into a scene), given the semantic information. For example, ground-truth pedestrian points are sampled from a database and placed into the scene based on the probability occupancy grid (compare how points are augmented by GT sampling (cyan) and our contextual GT sampling (red)). Note that a differently color-coded semantic segmentation map overlays all images.

Effect of Gradient Balancing. In Table 5, we further provide our ablation study to verify the effect of balancing gradients across multiple per-class detection heads. Given the PointPillars [15] as a baseline, we first modify the network architecture with per-class multiple detection heads (Model B). Then, we apply the following three multitask learning techniques: GradCosine [6], GradNorm [3], and Dynamic Weight Averaging (DWA, [16]). We observe in Table 5 that (i) applying per-class multiple detection heads generally improves the overall detection accuracy. Also, we observe that (ii) using Dynamic Weight Averaging outperforms other gradient balancing techniques, and DWA provides a performance gain potentially due to the balanced losses across different heads. Interestingly, the other two techniques (GradCosine and GradNorm) generally degrade the overall detection performance. This is probably due to a data imbalance problem and indicates that a contextual GT sampling technique needs to be used together.

4.4. Qualitative Analysis

Analysis with nuScenes dataset. In Figure 5 (a-f), we provide a qualitative comparison of predictions between our baseline (PointPillars+CBGS) and ours (PointPillars with our proposed per-class multi-heads with gradient balancing and contextual GT sampling). We provide six examples sampled from the nuScenes validation dataset with different color-coded bounding boxes (see caption). We observe that ours generally predict fewer false positives, especially for minor classes (see cyan boxes). This is probably because our per-class multiple detection heads encourage the model to learn more class-specific features, resulting in better robustness in detection. Further, we also observe that our model predicts objects in a more reasonable location. As we use contextual GT sampling, which considers more realistic places to augment objects, we observe that ours generally predicts objects in a more right place. For example, our baseline model produces prediction outputs of trucks in some counter-intuitive places.

Effect of Modifying Weights by Dynamic Weight Average. We also analyze weight ($\alpha_c(t)$) changes that balance

Table 4. The effect of contextual sampling on the 3D object detection performance (in mAP) with three existing object detection models: PointPillars [15], PV-RCNN [20], and Voxel RCNN [5]. We use the publicly available KITTI [9] validation set.

Model	Car (0.5)		Pedestrian (0.25)		Cyclist (0.25)		Avg.	
	3D↑	BEV↑	3D↑	BEV↑	3D↑	BEV↑	3D↑	BEV↑
PointPillars [15] + GT Sampling	94.50	94.66	70.40	70.72	71.03	71.03	78.64	78.80
PointPillars [15] + *Contextual* GT Sampling	94.99 (0.45↑)	95.09 (0.40↑)	70.93 (0.53↑)	71.23 (0.51↑)	72.80 (1.77↑)	72.80 (1.77↑)	79.57 (0.93↑)	79.71 (0.91↑)
PV-RCNN [20] + GT Sampling	94.42	96.20	75.32	75.60	79.15	79.15	82.96	83.65
PV-RCNN [20] + *Contextual* GT Sampling	94.74 (0.32↑)	96.50 (0.30↑)	75.50 (0.18↑)	75.92 (0.32↑)	83.25 (4.10↑)	83.25 (4.10↑)	84.50 (1.54↑)	85.22 (1.57↑)
Voxel RCNN [5] + GT Sampling	94.91	96.66	-	-	-	-	94.91	96.66
Voxel RCNN [5] + *Contextual* GT Sampling	97.08 (2.17↑)	97.31 (0.65↑)	-	-	-	-	97.08 (2.17↑)	97.31 (0.65↑)

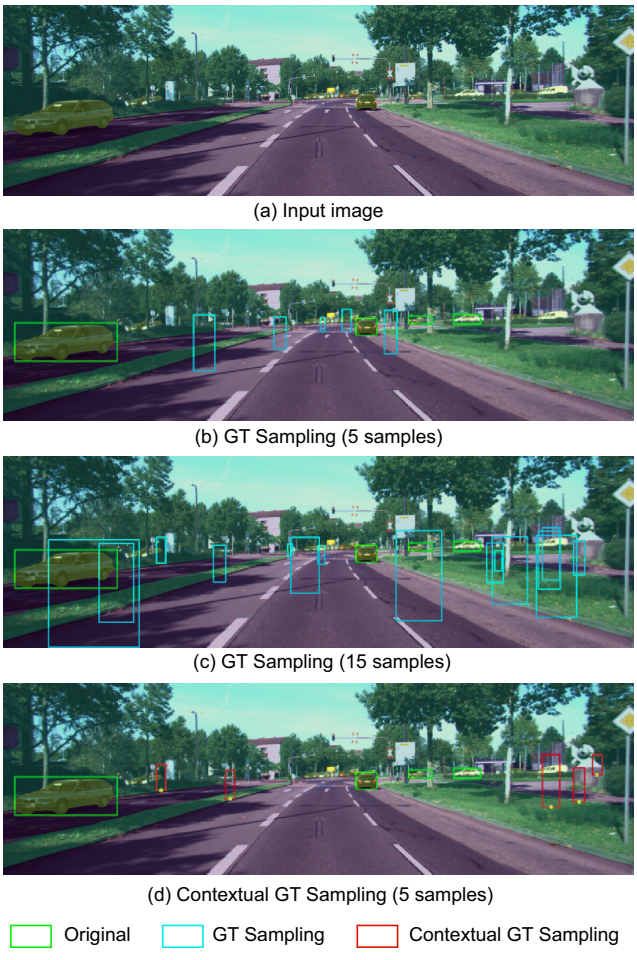

Figure 4. A comparison between existing ground-truth (GT) sampling and our proposed contextual ground-truth (GT) sampling. (a) A front-view image overlaid by a semantic segmentation map. An image with original bounding boxes (green) and augmented bounding boxes (cyan) from a ground-truth bounding box database either by (b-c) GT sampling method or (d) our proposed contextual GT sampling method.

Table 5. We compare 3D object detection accuracy with three different multitask learning techniques: GradCosine [6], GradNorm [3], and Dynamic Weight Average DWA [16]. We report scores on the KITTI [9] validation set with a set of higher IoU threshold.

Model	3D↑	BEV↑
A. PointPillars [15]	63.80	70.75
B. A + Per-Class Multiple Detection Heads	64.07	71.03
C. B + GradCosine [6]	63.60	71.09
D. B + GranNorm [3]	63.65	69.72
E. B + DWA [16]	**64.86**	**71.13**

losses from each detection head to reduce the data imbalance problem. Without contextual GT sampling (dashed), we observe that the model provides more weights on minor classes (compare green (cyclists) vs. red (cars)) to balance between two heads. This trend continues even with contextual GT sampling, but their weight gap is reduced. This is because our contextual GT sampling provides more examples for minor classes to balance the number of examples across classes.

Social Impact. We believe our effort to reduce the data imbalance problem is also in the mainstream of ethical AI, which focuses on removing implicit biases potentially from training data that might include biased data collection or reflect historical or social inequalities. We aim to make the model pay more attention to unrepresentative classes, resulting in lower error rates for minor classes while retaining the same or lower error rates for others. Also, as our model is for building autonomous driving systems, our work would inherit its social impact.

5. Conclusion

In this work, we introduced a method to address the data imbalance problem in the LiDAR-based 3D object detec-

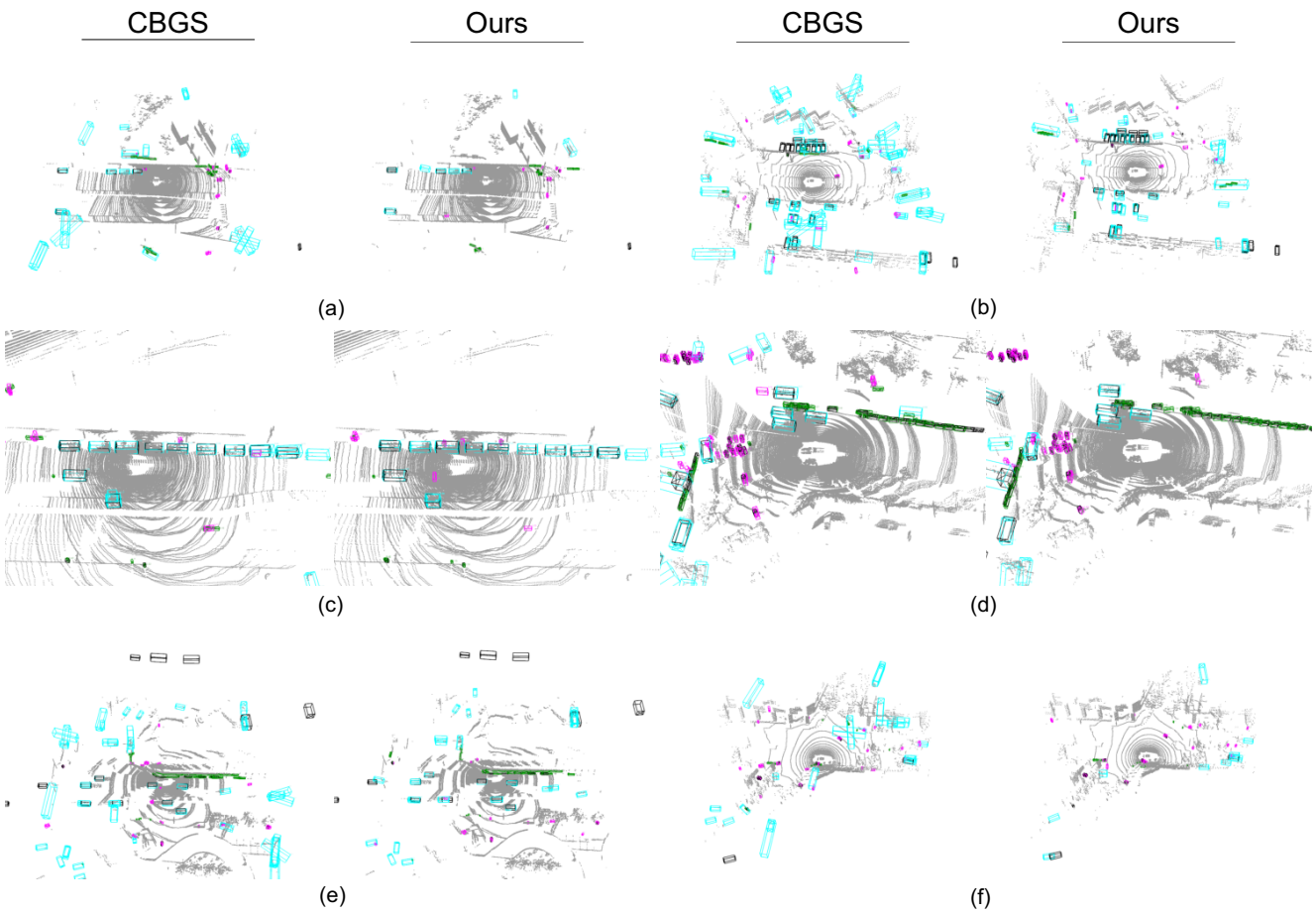

Figure 5. Comparison between CBGS [36] (based on PointPillars [15]) vs. Ours (PointPillars [15] + Per-Class Multiple Detection Heads with Gradient Balancing + Contextual GT Sampling) on nuScenes [8] validation set. Ground truth boxes are color-coded as black, while other predicted boxes are color-coded as cyan (Car, Truck, Construction Vehicle, Bus, Trailer), pink (Pedestrian, Bicycle, Motocycle), and green (Barrier, Traffic Cone).

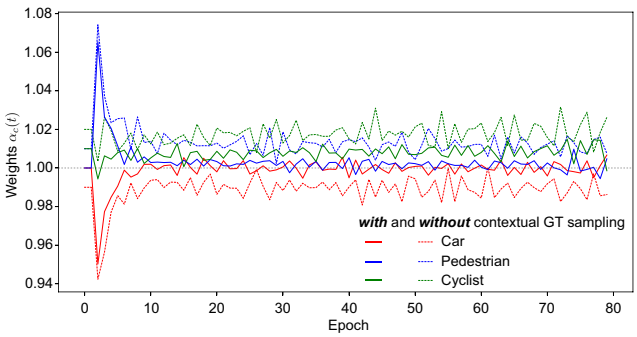

Figure 6. Changes in balancing weights $\alpha_c(t)$ (measured at the end of each epoch) for different classes: cars (red), pedestrians (blue), and cyclists (green). We also compare weights with (solid) and without (dashed) contextual GT sampling. Data: KITTI [9].

tion task. We proposed two main components: (1) multi-task learning-inspired per-class multi-headed LiDAR-based 3D object detector where losses from each head are modified to be balanced. (2) Contextual ground truth (GT) sampling, which improves the conventional GT sampling by leveraging semantic scene information to introduce objects in a more realistic location, resulting in a better quality of data augmentation. We conducted various experiments with a large-scale nuScenes dataset and a widely-used KITTI dataset. We demonstrated the effectiveness of our proposed method by showing improved accuracies of minor classes.

Acknowledgements. This work was supported by the grant from Autonomous Driving Center at Hyundai Motor Company's R&D Division and the grant by ITRC(Information Technology Research Center) support program (IITP-2022-RS-2022-00156295). We thank Jaewoo Cho, Nokyung Park, and Jongwon Park for their helpful comments.

References

[1] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[2] Xuelin Chen, Baoquan Chen, and Niloy J Mitra. Unpaired point cloud completion on real scans using adversarial training. *arXiv preprint arXiv:1904.00069*, 2019.

[3] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pages 794–803. PMLR, 2018.

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

[5] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1201–1209, 2021.

[6] Yunshu Du, Wojciech M Czarnecki, Siddhant M Jayakumar, Mehrdad Farajtabar, Razvan Pascanu, and Balaji Lakshminarayanan. Adapting auxiliary losses using gradient similarity. *arXiv preprint arXiv:1812.02224*, 2018.

[7] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015.

[8] Whye Kit Fong, Rohit Mohan, Juana Valeria Hurtado, Lubing Zhou, Holger Caesar, Oscar Beijbom, and Abhinav Valada. Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking. *arXiv preprint arXiv:2109.03805*, 2021.

[9] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[10] Georgia Gkioxari, Bharath Hariharan, Ross Girshick, and Jitendra Malik. R-cnns for pose estimation and action detection. *arXiv preprint arXiv:1406.5212*, 2014.

[11] Michelle Guo, Albert Haque, De-An Huang, Serena Yeung, and Li Fei-Fei. Dynamic task prioritization for multitask learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 270–287, 2018.

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

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

[14] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[15] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019.

[16] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1871–1880, 2019.

[17] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3994–4003, 2016.

[18] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[19] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. *Advances in neural information processing systems*, 30, 2017.

[20] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020.

[21] 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. *arXiv preprint arXiv:2102.00463*, 2021.

[22] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[23] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE transactions on pattern analysis and machine intelligence*, 43(8):2647–2664, 2020.

[24] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. `https://github.com/open-mmlab/OpenPCDet`, 2020.

[25] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4604–4612, 2020.

[26] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. Pointaugmenting: Cross-modal augmentation for 3d object

detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11794–11803, 2021.

[27] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. Grnet: Gridding residual network for dense point cloud completion. In *European Conference on Computer Vision*, pages 365–381. Springer, 2020.

[28] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

[29] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 206–215, 2018.

[30] Wang Yifan, Shihao Wu, Hui Huang, Daniel Cohen-Or, and Olga Sorkine-Hornung. Patch-based progressive 3d point set upsampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5958–5967, 2019.

[31] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021.

[32] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-net: Point cloud upsampling network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2790–2799, 2018.

[33] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 International Conference on 3D Vision (3DV)*, pages 728–737. IEEE, 2018.

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

[35] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018.

[36] 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.