Towards Long-Tailed 3D Detection

Anonymous Author(s)
Affiliation
Address
email

Abstract: Contemporary autonomous vehicle (AV) benchmarks have advanced techniques for training 3D detectors, particularly on large-scale LiDAR data. Surprisingly, although semantic class labels naturally follow a long-tailed distribution, contemporary benchmarks focus on only a few `common` classes (e.g., `pedestrian` and `car`) and neglect many `rare` classes in-the-tail (e.g., `debris` and `stroller`). However, AVs must still detect `rare` classes to ensure safe operation. Moreover, semantic classes are often organized within a hierarchy, e.g., tail classes such as `child` and `construction-worker` are arguably subclasses of `pedestrian`. However, such hierarchical relationships are often ignored, which may lead to misleading estimates of performance and missed opportunities for algorithmic innovation. We address these challenges by formally studying the problem of *Long-Tailed 3D Detection* (LT3D), which evaluates on *all* classes, including those in-the-tail. We evaluate and innovate upon popular 3D detection codebases, such as CenterPoint and PointPillars, adapting them for LT3D. We develop hierarchical losses that promote feature sharing across common-vs-rare classes, as well as improved detection metrics that award partial credit to "reasonable" mistakes respecting the hierarchy (e.g., mistaking a `child` for an `adult`). Finally, we point out that fine-grained tail class accuracy is particularly improved via multimodal *fusion* of RGB images with LiDAR; simply put, small fine-grained classes are challenging to identify from sparse (LiDAR) geometry alone, suggesting that multimodal cues are crucial to long-tailed 3D detection. Our modifications improve accuracy by 5% AP on average for all classes, and dramatically improve AP for `rare` classes (e.g., `stroller` AP improves from 3.6 to 31.6).

Keywords: Autonomous Vehicles, Long-Tailed 3D Detection, Multimodal Fusion

1 Introduction

3D object detection is a key component in many robotic systems such as autonomous vehicles (AVs) [1, 2]. To facilitate research in this space, the AV industry has released large-scale 3D-annotated multimodal datasets [2, 3, 4]. However, these datasets benchmark on only a few `common` classes such as `pedestrian` and `car`. In practice, safe navigation [5, 6] requires AVs to reliably detect `rare`-class objects such as `child` and `stroller`. This motivates the problem of *Long-Tailed 3D Detection* (LT3D), which requires detecting objects from both `common` and `rare` classes.

Status Quo. Among contemporary AV datasets, nuScenes [2] has exhaustively annotated objects of various classes crucial to AVs (Fig. 1) and organized them with a semantic hierarchy (Fig. 3). However, the nuScenes benchmark ignores most `rare` classes, presumably because they have too few examples to train good detectors. As it focuses on only a few (`common`) classes, prior works miss opportunities to exploit this semantic hierarchy during training. We argue that these benchmarking protocols are flawed because detecting fine-grained classes is useful for downstream tasks such as motion planning. This motivates us to study LT3D by re-purposing *all* annotated classes in nuScenes.

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.

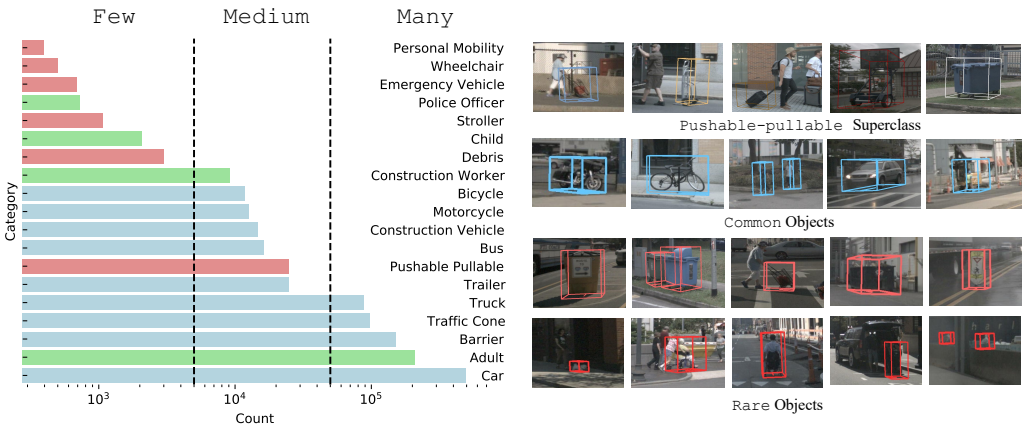

Figure 1: According to the histogram of per-class object counts (on the **left**), the nuScenes benchmark focuses on the common classes in cyan (e.g., car and barrier) but ignores rare ones in red (e.g., stroller and wheelchair). In fact, the benchmark creates a superclass pedestrian by grouping multiple classes in green, including the common class adult and several rare classes (e.g., child and police-officer); this complicates the analysis of detection performance as pedestrian performance is dominated by adult. Moreover, the ignored superclass pushable-pullable also contains diverse objects such as shopping-cart, dolly, luggage and trash-can as shown in the top row (on the **right**). We argue that AVs should also detect rare classes as they can affect AV behaviors. Following [7], we report performance with three groups of classes based on their cardinality (split by dotted lines): Many, Medium, and Few.

Protocol. LT3D requires 3D localization and recognition of objects from each of the common (e.g., adult and car) and rare classes (e.g, child and stroller). Moreover, for safety-critical robots such as autonomous vehicles, we believe detecting but mis-classifying rare objects (e.g., mis-classifying a child as an adult) is preferable to failing to detect them at all. Therefore, we propose a new metric to quantify the severity of classification mistakes in LT3D that exploits inter-class relationships to award partial credit (Fig. 3). We use both the standard and proposed metrics to evaluate 3D detectors on all classes.

Technical Insights. To address LT3D, we first retrain state-of-the-art LiDAR-based 3D detectors on *all* classes. Naively retraining detectors produces poor performance on rare classes (e.g., yielding 0.1 AP on child and 3.6 AP on stroller). We propose several algorithmic innovations to improve these results. First, to encourage feature sharing across common-vs-rare classes, we learn a single feature trunk, adding in hierarchical coarse classes that ensure features will be useful for both common and rare classes. Second, we find that LiDAR data is simply too impoverished for even humans to recognize certain tail objects that tend to be small, such as strollers. We explore multimodal-fusion detectors, and introduce a simple approach that post-processes single-modal 3D detections from LiDAR and RGB inputs, filtering away detections that are inconsistent across modalities. Our innovations significantly improve performance by 5 AP on average, and dramatically boost performance when allowing for partial credit (e.g., achieving 16.9 / 38.8 AP for child / stroller).

Contributions. We make three major contributions. First, we formulate the problem LT3D, emphasizing detection of both common and rare classes in safety-critical AVs. Second, we design LT3D's benchmarking protocol and develop a supplemental metric that awards partial credit depending on the severity of misclassifications (e.g., misclassifying child-vs-adult is less problematic than misclassifying child-vs-car). Third, we propose several architecture-agnostic approaches to LT3D, including a simple multimodal fusion technique that uses RGB to filter out false-positive LiDAR-based detections, which significantly improves detection precision for rare classes.

2 Related Works

3D Object Detection has been widely studied in the context of autonomous vehicle (AV) research. Contemporary benchmarks favor LiDAR-based detectors, emphasizing common classes and ignoring

`rare` ones. Approaches to 3D detection usually adopt an anchor-based model architecture that defines per-class shapes to guide class-aware object detection [8, 9, 10, 11, 12]. A recent *anchor-free* model, CenterPoint [13] achieves the state-of-the-art for LiDAR-based 3D object detection. Specifically, it learns to predict an object's center and estimates the 3D shape for each detected object's center. Existing LiDAR-based 3D detectors exclusively focus on data from `common` classes [8, 9, 13] and do not study how to detect `rare` classes. RBG-based 3D detectors underperform LiDAR-based methods because the RGB input does not provide reliable 3D measures (unlike LiDAR). As a result, RGB-based 3D detectors are not widely adopted. However, in exploring LT3D we find that RGB-detectors shine for detecting objects of `rare` classes. Importantly, multimodal fusion significantly improves LT3D.

Multimodal 3D Detection. Conventional wisdom suggests that fusing multimodal cues, particularly using LiDAR and RGB, can improve 3D detection. Intuitively, LiDAR faithfully measures the 3D world (although it has notoriously sparse point returns), and RGB is high-resolution (but lacks 3D information). Multimodal fusion for 3D detection is an active field of exploration. Existing methods suggest different ways to fuse the two modalities. Proposed methods encode separate modalities and fuse object proposals [14, 15, 16, 17, 18], augment LiDAR points with either RGB features [19], augment RGB images with LiDAR points [20] or add semantic information obtained by processing RGB inputs [21, 22]. Others propose stage-wise methods that first detect boxes over images and localize in 3D with LiDAR [23] and fuse detections from single-modal detectors [24, 25]. While the above methods have not been tested for LT3D, our work shows that RGB is a key modality for LT3D.

Long-Tailed Perception (LTP). Real-world data tends to follow long-tailed class distributions [26], i.e., a few classes are dominant in the data, while many others are rarely seen. LTP is a long-standing problem in the literature [7]. It has been widely studied through the lens of image classification and requires training on class-imbalanced data, aiming for high accuracy averaged across imbalanced classes [7, 27, 28]. Existing methods propose reweighting losses [29, 30, 31, 32, 33, 34], rebalancing data sampling [35, 36, 37], balancing gradients computed from imbalanced classes [38], and balancing network weights [28]. Others study LTP through the lens of 2D object detection over RGB images [39]. To the best of our knowledge, long-tailed 3D detection (LT3D) has not been explored yet. In LT3D, we find a unique challenge, rare classes are not only infrequent, but are also difficult to distinguish using LiDAR alone. This motivates us to use RGB to complement LiDAR. We find using both RGB (for better recognition) and LiDAR (for better 3D localization) helps detect `rare` classes.

3 LT3D: Methodology

To approach LT3D, we first retrain SOTA 3D detectors on *all* classes, including LiDAR-based detectors (PointPillars [8], CBGS [9], and CenterPoint [13]), an RGB-based detector FCOS3D [40], and multimodal detectors (MVP [22] and TransFusion [17]). We further introduce several modifications to these models that consistently improve LT3D.

Grouping-Free Detector Head. Extending existing 3D detectors to train with more classes is surprisingly challenging. Many contemporary networks use a multi-head architecture that groups classes of similar size and shape to facilitate efficient feature sharing. For example, CenterPoint groups `pedestrian` and `traffic-cone` since these objects are both tall and skinny. However, multi-headed grouping strategies may not work for diverse classes like `pushable-pullable` and `debris`. Therefore, we first consider making each class its own group to avoid hand-crafted grouping heuristics. However, heads of rare classes overfit and added heads use considerably more GPU memory. Our final solution is to merge all classes into a single group with a proportionally heavier detector head to simplify training. Adding a new class is as simple as adding a single convolutional channel to the detector output. Our grouping-free detector head achieves unchanged accuracy over grouping-based methods.

Training with Semantic Hierarchies. nuScenes defines a semantic hierarchy (Fig. 3) for all classes, grouping semantically similar classes under coarse-grained categories. We leverage this hierarchy during training. Specifically, we train detectors to predict three labels for each object: its fine-grained

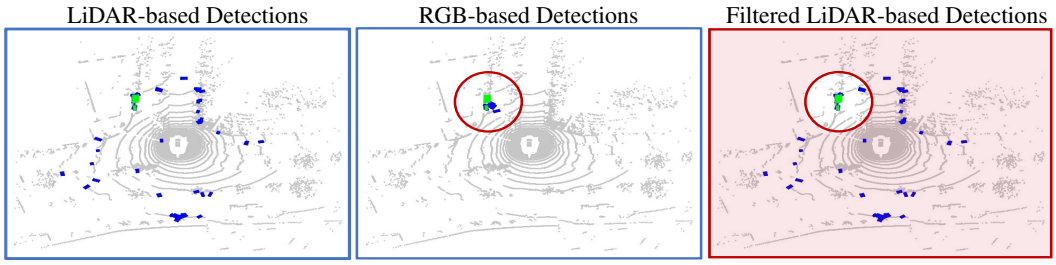

Figure 2: Multimodal filtering effectively removes high-scoring false-positive LiDAR detections. The green boxes are ground-truth `strollers`, while the blue boxes are `stroller` detections from our best performing models, liDAR-based detector CenterPoint [13] (**left**) and RGB-based detector FCOS3D [40] (**mid**). The final filtered result removes LiDAR detections not within m meters of any RGB detection (**right**).

label (e.g., `child`, its coarse class (e.g., `pedestrian`), and the root class `object`. Given that we adopt a grouping-free detector head that outputs separate "multitask" heatmaps for each class, we use a per-class logit-loss rather than multi-class softmax loss, essentially treating each class to be detected as a separate task (that shares the same feature trunk). Given this architectural modification, it is "trivial" to add additional coarse classes. Crucially, because we do not employ softmax losses, adding a `vehicle` heatmap does not directly interfere with the `car` heatmap (as they would with a multi-class softmax loss). However, this might produce repeated detections on the same test object. We address that by simply ignoring coarse detections at test time. Perhaps surprisingly, this training method improves detection performance not only for `rare` classes, but also for `common` classes. This is presumably because it regularizes the learned features to generalize better.

Augmentation Schedule. Class-balanced resampling is a common technique in learning with long-tailed classes. This augmentation strategy increases the number of `rare` objects seen in training but skews the class distribution and leads to more false positives for `rare` classes in inference. Prior works [21, 17] suggest disabling class-balanced resampling for the last few training epochs to better match the real class distribution, reducing false positives. We validate this approach in training 3D detectors and find that it often improves performance for `rare` classes at the cost of `common` classes.

Multimodal Fusion by Filtering. Small fine-grained classes are challenging to identify from sparse (LiDAR) geometry alone, suggesting that multimodal cues can improve long-tailed detection. We evaluate several multimodal fusion algorithms, but find a simple strategy of post-processing filtering to work remarkably well. Although LiDAR-based detectors are widely adopted for 3D detection, we find that they produce many high-scoring false positives (FPs) for rare classes due to misclassification. We focus on removing such FPs. To this end, we use an RGB-based detector to filter out high-scoring false-positive LiDAR detections by leveraging two insights: (1) LiDAR-based 3D-detectors are accurate w.r.t 3D localization and yield high recall (though classification is poor), and (2) RGB-based 3D-detections are accurate w.r.t recognition (though 3D localization is poor). Fig. 2 demonstrates this filtering strategy. For each RGB-based detection, we search in a radius of m meters for a LiDAR-based detection, keep them and remove all the others (that are not close to any RGB-based detections). We use FCOS3D [40] as the RGB-based detector in this work.

4 LT3D: Evaluation Protocol

Conceptually, LT3D extends the traditional 3D detection problem, which focuses on identifying objects from K `common` classes, by further requiring detection of N `rare` classes. Recall that as LT3D is motivated by safety concerns in AVs, we further propose a complementary hierarchical AP metric to better diagnose detector performance by analyzing cross-category mistakes.

4.1 Evaluation Metrics

As LT3D emphasizes detection performance on *all* classes, we report the metrics for three groups of classes based on their cardinality (Fig. 1-left): *many* (>50k objects per class), *medium* (5k∼50k), and *few* (<5k). We describe the metrics below.

Standard Detection Metrics. Mean average precision (mAP) is an established metric for object detection [41, 1, 42]. For 3D detection on LiDAR sweeps, a true positive (TP) is defined as a detection that has a center distance within a distance threshold on the ground-plane to a ground-truth annotation [2]. mAP computes the mean of AP over classes, where per-class AP is the area under the precision-recall curve, and distance thresholds of [0.5, 1, 2, 4] meters.

Hierarchical Mean Average Precision (mAP$_H$). For safety critical applications, although correctly localizing and classifying an object is ideal, detecting and misclassifying *some* object is more desirable than a missed detection (e.g., detect but misclassify a `child` as `adult` versus not detecting this `child`). Therefore, we introduce hierarchical AP (AP$_H$) which considers such semantic relationships across classes to award partial credit. We report results for *both* standard detection metrics and our proposed metric.

To encode these relationships between classes, we leverage the semantic hierarchy (Fig. 3) defined by nuScenes. We derive partial credit as a function of semantic similarity using the least common ancestor (LCA) distance metric. Hierarchical metrics have been proposed for image classification [43], but have not been explored for object detection. Extending this metric for object detection is challenging because we must consider how to jointly evaluate semantic and spatial overlap. We define our hierarchical AP metric as follows, according to the semantic hierarchy defined in Figure 3. For clarity, we will describe the procedure in context of computing AP$_H$ for some arbitrary class C.

LCA=0: Consider the set predictions and ground-truth boxes for C. Label the set of predictions that overlap with ground-truth boxes for C as true positives. Other predictions are false positives. *This is identical to the standard AP metric.*

LCA=1: Consider the predictions for C, and ground-truth boxes for C and all sibling classes of C (that have LCA distance to C of 1). Label the set of predictions that overlap a ground-truth box of C as a true positive. Label the set of predictions that overlap sibling classes as `ignored` [42]. All other predictions are false positives.

LCA=2: Consider the predictions for C and ground-truth boxes for C and all classes that have LCA distance to C less than 2. For nuScenes, this includes all classes. Label the set of predictions that overlap ground-truth boxes for C as true positives. Label the set of predictions that overlap other classes as `ignored`. All other predictions are false positives.

5 Experiments

We conduct experiments to better understand the LT3D problem, and gain insights by validating our techniques described in Section 3. Specifically, we aim to answer the following questions:[1]

1. Are `rare` classes more difficult to detect than `common` classes?
2. Are objects from `rare` classes sufficiently localized but mis-classified?
3. Does training with the semantic hierarchy improve detection performance for LT3D?
4. Does multimodal fusion help detect `rare` classes?

We start this section by introducing the model architecture, implementation and dataset.

Model Architecture. For LiDAR-based 3D detectors, we use PointPillars [8], CBGS [9], and CenterPoint [13], which are widely benchmarked in the literature. For fusion-based 3D detectors, we use MVP [22] and TransFusion [17], which are recently released state-of-the-art methods. MVP uses off-the-shelf RGB segmentation models to localize objects and densify LiDAR point clouds. TransFusion proposes an end-to-end DETR-like approach [44] for multimodal 3D detection.

Implementation. We use the PyTorch toolbox [45] to train all models for 20 epochs with the Adam optimizer [46] and a one-cycle learning rate scheduler [47]. In training, we adopt data augmentation techniques introduced by [13]. When using the introduced data augmentation schedule (cf. Section 3), we train models for 15 epochs with data augmentation enabled, and 5 epochs without.

[1] Answers: yes, yes, yes, yes.

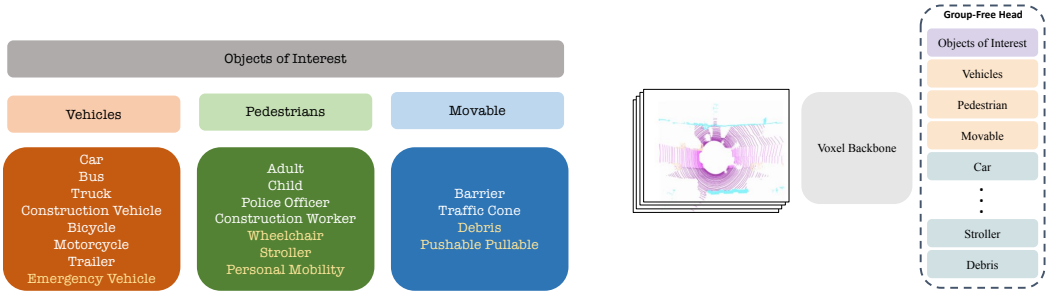

Figure 3: nuScenes defines a semantic hierarchy (on the **left**) for all annotated classes (Fig. 1). We highlight `common` classes in white and `rare` classes in gold. The standard nuScenes benchmark makes two choices for dealing with rare classes: (1) ignore them (e.g., `stroller` and `pushable-pullable`), or (2) group them into coarse-grained classes (e.g., `adult`, `child`, `construction-worker`, `police-officer` are grouped as `pedestrian`). Since the `pedestrian` class is dominated by `adult` (Fig. 1), the standard benchmarking protocol masks the challenge of detecting rare classes like `child` and `police-officer`. We leverage this hierarchy during training (on the **right**) by predicting class labels at *multiple* levels of the hierarchy. Specifically, we train detectors to predict three labels for each object: its fine-grained label (e.g., `child`, its coarse class (e.g., `pedestrian`), and the root-level class `object`. This means that the final vocabulary of classes is no longer mutually exclusive, complicating the application of multi-class softmax losses. To address this, use per-class logit loss functions that learn separate spatial heatmaps for each class.

5.1 Dataset

Among many AV datasets (e.g., Argoverse [3], KITTI [1] and Waymo [4]), we use nuScenes to explore LT3D because it has the most long-tailed classes (23 in total) arranged in a semantic hierarchy (Fig. 3). The nuScenes benchmark ignores most of these annotated classes and the hierarchy, resulting in 10 classes for its final benchmark (Fig. 1). For LT3D, we evaluate on the 19 classes at the finest level (i.e., two levels below the root, cf. Fig. 3). Following prior work, we use the nuScenes official train-set to train all the models and evaluate on the nuScenes validation set.

Retraining state-of-the-art 3D detectors for LT3D. We retrain four 3D detectors, namely FCOS3D [40], PointPillars [8], CBGS [9], and CenterPoint [13]. FCOS3D operates on monocular images. The other three detectors take an aggregated stack of ten LiDAR-sweeps as input. All four models predict 3D bounding boxes for 19 classes as defined by the nuScenes LT3D protocol. Table 1 shows that mAP of `rare` classes are much lower than `common` classes, confirming that `rare` classes are more difficult to detect than `common` ones. Moreover, LiDAR-based detectors that perform well on `common` classes tend to also perform well on `rare` classes. See details in the caption of Table 1.

Training with Semantic Hierarchy. Next, we modify our LiDAR-based detectors to jointly predict class labels at different levels of the semantic hierarchy. For example, we modify the detector to additionally classify `stroller` as `pedestrian` and `object`. The semantic hierarchy naturally groups classes based on shared attributes and may have complementary features. Moreover, training with the semantic hierarchy allows `rare` classes within each group to learn better features by sharing with `common` classes. This approach is generally effective, as shown in Table 1, improving accuracy for classes with `Many` examples by 2% AP, `Medium` examples by 2% AP and `Few` examples by 1% AP. Notably, training with the semantic hierarchy improves CenterPoint by 4% AP on `Many` classes.

Data Augmentation Schedule. Prior works [24, 17] suggest disabling copy-paste augmentation for the last few epochs of training to reduce the number of false positive detections. We validate this claim for various detector architectures and find that although it seems to help `rare` classes by 3% AP, but hurts `common` classes by 4% AP (c.f. CenterPoint).

Multimodal Filtering. Detecting `rare` classes from LiDAR-only is challenging since its difficult to learn to recognize objects from sparse LiDAR points and from limited examples. As a result, LiDAR-detectors often have many high-scoring FPs (Fig. 2), resulting in low AP. Using RGB detections to filter the LiDAR detections results in significant performance improvement on `rare` classes – 4$\sim$11 AP increases on classes of `Medium` and `Few` for all models (Table 1)!

Table 1: **Benchmarking detectors for LT3D** (measured by mAP). We present several salient conclusions. First, training with the semantic hierarchy improves all methods for LT3D, e.g., improving PointPillars and CBGS by 1% AP and CenterPoint by 3% AP averaged over `All` classes. It seems to have a bigger impact on the heatmap-based detector head than the standard anchor-based detector heads. It also slightly helps multimodal detectors (within 1% AP for MVP and TransFusion). Second, multimodal filtering yields 4∼11 AP improvement on `Medium` and `Few` classes! This is surprising given that FCOS3D is a less powerful 3D detector on its own. Interestingly, it also improves multimodal detectors (3.4% and 1.6% AP improvement for MVP and TransFusion on `All` classes), demonstrating the importance of using RGB to improve LT3D with better recognition. Third, perhaps surprisingly, post-hoc multimodal filtering of LiDAR-only detector CenterPoint with RGB-only detector FCOS3D performs the best, surpassing multimodal detectors MVP and TransFusion. Lastly, data augmentation schedules do not necessarily improve LT3D performance, demonstrating the challenge of 3D detection in the long-tail.

Method	Multimodal	`Many`	`Medium`	`Few`	`All`
FCOS3D [40]		39.0	23.3	2.9	20.9
PointPillars [8]		64.2	28.4	3.4	30.0
+ Hierarchy		**66.4**	30.4	2.9	31.2
w/ Data Aug.		54.4	24.2	1.8	25.1
w/ Filtering	✓	66.2	**41.0**	**4.4**	**35.8**
CBGS [9]		47.2	10.4	**0.1**	17.2
+ Hierarchy		**49.5**	11.1	**0.1**	18.1
w/ Data Aug.		49.9	17.1	**0.1**	20.6
w/ Filtering	✓	48.0	**20.3**	**0.1**	**21.5**
CenterPoint [13]		73.7	41.3	3.0	37.5
+ Hierarchy		**77.1**	45.1	4.3	40.4
w/ Data Aug.		73.8	44.5	7.4	40.3
w/ Filtering	✓	**77.1**	**49.0**	**9.4**	**43.6**
MVP [22]	✓	65.6	31.6	1.5	31.0
+ Hierarchy	✓	67.0	33.0	0.1	31.6
w/ Data Aug.	✓	65.9	35.8	0.1	32.5
w/ Filtering	✓	**67.1**	**39.2**	**1.6**	**34.4**
TransFusion [17]		68.5	**42.8**	8.4	38.5
+ Camera	✓	**73.9**	41.2	**9.8**	39.8
w/ Data Aug.	✓	73.4	40.9	8.2	39.0
w/ Filtering	✓	**73.9**	42.5	9.1	**40.1**

End-to-End Multimodal Methods. Since multimodal cues significantly improve LT3D, we are motivated to explore end-to-end approaches. Specifically, we retrain MVP [22] and TransFusion [17] on all 19 classes. MVP trains a 2D semantic segmentation model (c.f. CenterNet2 [48]) on nuImages. Using this trained model, MVP segments classes of interest and augments the point cloud with virtual points. Applying this method to LT3D shows 10 AP worse performance compared to LiDAR-only methods (c.f. CenterPoint), because (1) it is challenging to retrain the segmentation network CenterNet2 for long-tailed classes and it works particularly poorly on `rare` classes. We explore this further in the supplement. We retrain TransFusion, downsampling the RGB images by a factor of 2 to fit the model in GPU memory. Surprisingly, TransFusion performs worse than CenterPoint for LT3D, indicating that strong performance on `common` classes (as widely benchmarked) is not indicative of trends for all long-tailed classes. Lastly, our multimodal filtering strategy still improves these end-to-end fusion methods slightly, e.g., it increases mAP on `All` classes by 3.4% and 0.3% AP for MVP and TransFusion (cf. Table 1).

Benchmarking with Hierarchical Average Precision (mAP_H). For safety-critical AVs, mis-classification between semantically similar classes is preferable to missed detections. Our proposed mAP_H allows for more careful diagnosis of 3D detectors. We find performance increases significantly from LCA=0 to LCA=1 compared against LCA=1 to LCA=2, confirming that most mis-classification occurs amongst semantically similar classes. That said, improving `rare` object classification should greatly improve LT3D. Unsurprisingly, we see a considerable jump between LCA=0 and LCA=1 for rare classes, confirming that objects from `rare` classes are often detected but mis-classified. Somewhat surprisingly, we find that despite allowing for partial credit, `child` AP is still alarmingly

Table 2: **Diagnosis using the mAP$_H$ metric on selected classes**. We analyze the best-performing LiDAR-only model CenterPoint and multimodal model TransFusion, with / without our hierarchical loss (*hier*) and mutimodal filtering technique (*filtering*). Comparing the rows of "LCA=0" for with and without our techniques (for CenterPoint and TransFusion respectively), we see our techniques bring significantly improvements on classes with `medium` and `few` examples such as `construction-vehicle` (CV), `bicycle`, `motorcycle` (MC), `construction-worker` (CW), `stroller`, and `pushable-pullable` (PP). Moreover, performance increases significantly from LCA=0 to LCA=1 compared against LCA=1 to LCA=2, suggesting that most misclassification occurs amongst semantically similar classes (Table 2). Unsurprisingly, we see a considerable jump between LCA=0 and LCA=1 for rare classes, confirming that objects from `rare` classes are often detected but misclassified as some sibling classes.

Method	mAP_H	Car	Adult	Truck	CV	Bicycle	MC	Child	CW	Stroller	PP
CenterPoint	LCA=0	86.5	84.0	53.9	23.5	47.2	60.2	0.1	20.2	3.6	32.2
	LCA=1	87.3	84.7	59.5	25.2	48.8	61.7	0.1	26.4	3.8	32.4
	LCA=2	87.3	85.0	59.6	25.3	49.5	62.1	0.1	27.2	4.0	32.9
CenterPoint *w/ Hier.. & Filter*	LCA=0	88.5	86.6	63.4	29.0	58.5	68.2	5.3	35.8	31.6	39.3
	LCA=1	89.4	87.4	72.4	31.3	61.2	69.7	15.2	52.0	37.7	39.4
	LCA=2	89.5	87.7	72.5	31.5	62.3	69.9	16.9	56.3	38.8	39.8
TransFusion	LCA=0	84.4	84.5	58.5	15.1	44.9	57.2	1.0	15.1	3.2	19.6
	LCA=1	85.5	85.7	67.4	21.8	46.7	59.1	1.6	21.8	3.7	19.8
	LCA=2	85.5	86.1	67.5	22.6	47.7	59.9	1.7	22.6	4.2	20.4
TransFusion *w/ Hier. & Filtering*	LCA=0	84.4	84.2	58.4	25.3	52.3	62.8	4.0	27.5	14.7	27.3
	LCA=1	86.0	85.4	67.3	26.6	55.7	64.0	25.1	46.7	24.3	27.4
	LCA=2	86.0	85.9	67.4	27.0	56.9	64.3	25.8	48.6	28.3	27.9

low compared to `adult`. Models cannot confidently detect `child`, even accounting for misclassification.

6 Limitations

Our work explores long-tailed 3D detection (LT3D) in the context of autonomous vehicles (AVs). LT3D emphasizes object detection for `rare` classes which can be safety-critical for downstream AV tasks such as motion planning and collision avoidance. However, our work does not study how solving LT3D directly affects these tasks. Future work should address this limitation.

Another limitation, shared by contemporary benchmarks, is that our setup does not consider the correlation between individual classes. For example, the rare-class `stroller` is often pushed by an `adult`. One may argue that detecting `adult` is sufficient for safe navigation. However, edge cases can occur in the real world where a `stroller` can be unattended. Therefore, we expect future work to diagnose class correlation and such edge cases.

7 Conclusion

We explore the problem of long-tailed 3D detection (LT3D), detecting objects not only from `common` classes but also from many `rare` classes. This problem is motivated by the operational safety of autonomous vehicles (AVs), which must reliably detect `rare` classes for appropriate motion planning and collision avoidance. To study LT3D, we establish rigorous evaluation protocols that allow for partial credit to better diagnose 3D detectors. Our study shows that state-of-the-art LiDAR-based 3D detectors achieve poor performance on `rare` classes because they often misclassify `rare`-class objects. We propose several algorithmic innovations to improve LT3D, including a group-free detector head, hierarchical losses that promote feature sharing across long-tailed classes, and a simple multimodal fusion method that uses RGB-based detections to filter LiDAR-based detections, achieving significant improvement for LT3D. Importantly, our study shows that multimodal cues are crucial to LT3D, suggesting that multimodal 3D detectors can shine in the long-tail.

References

[1] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[2] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[3] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[4] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[5] A. Taeihagh and H. S. M. Lim. Governing autonomous vehicles: emerging responses for safety, liability, privacy, cybersecurity, and industry risks. *Transport Reviews*, 39(1):103–128, 2019.

[6] K. Wong, S. Wang, M. Ren, M. Liang, and R. Urtasun. Identifying unknown instances for autonomous driving. In *CoRL*, 2020.

[7] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu. Large-scale long-tailed recognition in an open world. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[8] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[9] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint arXiv:1908.09492*, 2019.

[10] P. Hu, J. Ziglar, D. Held, and D. Ramanan. What you see is what you get: Exploiting visibility for 3d object detection. *CoRR*, abs/1912.04986, 2019.

[11] Y. Yan, Y. Mao, and B. Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18 (10):3337, 2018.

[12] G. Wang, Y. Wang, H. Zhang, R. Gu, and J.-N. Hwang. Exploit the connectivity: Multi-object tracking with trackletnet. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 482–490, 2019.

[13] T. Yin, X. Zhou, and P. Krähenbühl. Center-based 3d object detection and tracking. *arXiv preprint arXiv:2006.11275*, 2020.

[14] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017.

[15] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. Waslander. Joint 3d proposal generation and object detection from view aggregation. *IROS*, 2018.

[16] J. H. Yoo, Y. Kim, J. S. Kim, and J. W. Choi. 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. In *European Conference on Computer Vision (ECCV)*, 2020.

[17] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C. Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. *CoRR*, abs/2203.11496, 2022.

[18] Y.-T. Chen, J. Shi, Z. Ye, C. Mertz, S. Kong, and D. Ramanan. Multimodal object detection via probabilistic ensembling. *arXiv preprint arXiv:2104.02904*, 2021.

[19] V. A. Sindagi, Y. Zhou, and O. Tuzel. Mvx-net: Multimodal voxelnet for 3d object detection. In *International Conference on Robotics and Automation (ICRA)*, 2019.

[20] Y. You, Y. Wang, W.-L. Chao, D. Garg, G. Pleiss, B. Hariharan, M. Campbell, and K. Q. Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. *arXiv arXiv:1906.06310*, 2020.

[21] S. Vora, A. H. Lang, B. Helou, and O. Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[22] T. Yin, X. Zhou, and P. Krähenbühl. Multimodal virtual point 3d detection. *NeurIPS*, 2021.

[23] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *IEEE conference on computer vision and pattern recognition*, 2018.

[24] D. Xu, D. Anguelov, and A. Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *IEEE conference on computer vision and pattern recognition*, 2018.

[25] S. Pang, D. Morris, and H. Radha. Clocs: Camera-lidar object candidates fusion for 3d object detection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[26] W. J. Reed. The pareto, zipf and other power laws. *Economics letters*, 74(1):15–19, 2001.

[27] Y. Zhang, B. Kang, B. Hooi, S. Yan, and J. Feng. Deep long-tailed learning: A survey. *arXiv:2110.04596*, 2021.

[28] S. Alshammari, Y.-X. Wang, D. Ramanan, and S. Kong. Long-tailed recognition via weight balancing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[29] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, 2019.

[30] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems*, 29(8):3573–3587, 2017.

[31] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, 2019.

[32] S. Khan, M. Hayat, S. W. Zamir, J. Shen, and L. Shao. Striking the right balance with uncertainty. In *CVPR*, 2019.

[33] C. Huang, Y. Li, C. C. Loy, and X. Tang. Deep imbalanced learning for face recognition and attribute prediction. *PAMI*, 42(11):2781–2794, 2019.

[34] S. Zhang, Z. Li, S. Yan, X. He, and J. Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *CVPR*, 2021.

[35] C. Drummond, R. C. Holte, et al. C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on learning from imbalanced datasets II*, 2003.

[36] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[37] H. Han, W.-Y. Wang, and B.-H. Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International Conference on Intelligent Computing*, pages 878–887. Springer, 2005.

[38] K. Tang, J. Huang, and H. Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *NeurIPS*, 2020.

[39] A. Gupta, P. Dollar, and R. Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019.

[40] T. Wang, X. Zhu, J. Pang, and D. Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *IEEE/CVF International Conference on Computer Vision*, pages 913–922, 2021.

[41] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision (IJCV)*, 2015.

[42] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014.

[43] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[44] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, 2020.

[45] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019.

[46] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[47] L. N. Smith. Cyclical learning rates for training neural networks. In *IEEE Winter Conference on Applications of Computer Vision, WACV*, 2017.

[48] X. Zhou, V. Koltun, and P. Krähenbühl. Probabilistic two-stage detection. In *arXiv preprint arXiv:2103.07461*, 2021.