Towards Long-Tailed 3D Detection

Neehar Peri[1], Achal Dave[1], Deva Ramanan[1,2], Shu Kong[3]
[1]Robotics Institute, Carnegie Mellon University
[2]Argo AI, [3]Dept. of Computer Science and Engineering, Texas A&M University
fnperi, adave, devag@andrew.cmu.edu shu@tamu.edu

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, these 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, in the real open world, 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 yield 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 detectors, 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, 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 0.1 to 31.6)! Our code is available on GitHub.

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

1 Introduction

3D object detection is a key component in many robotics 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 the real open world, safe navigation [5, 6] requires AVs to reliably detect rare-class objects such as child and stroller. This motivates Long-Tailed 3D Detection (LT3D), a problem requiring 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). 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.

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

6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.

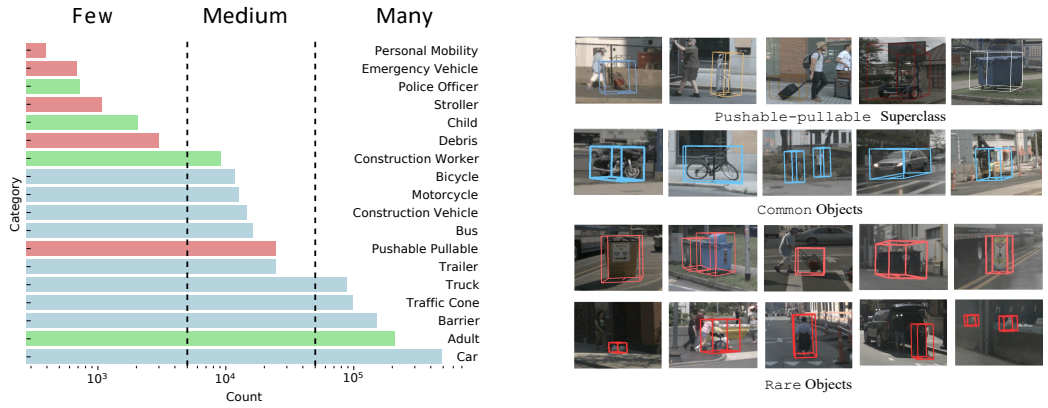

Figure 1: According to the histogram of per-class object counts (on the left), the nuScenes benchmark focuses on the common classes in cyan (e.g., car and barrier) but ignores rare ones in red (e.g., stroller and debris). In fact, the benchmark creates a superclass pedestrian by grouping multiple classes 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 for three groups of classes based on their cardinality (split by dotted lines): Many, Medium, and Few.

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 0.1 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, 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 on LT3D by 5 % AP on average, greatly boosting 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-based detections to filter out false-positive LiDAR-based detections, leading to significant improvement of rare-class detection.

2 Related Works

3D Object Detection. Contemporary benchmarks, often in the AV setting, 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. RGB-based 3D detectors underperform LiDAR-based methods because the monocular 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, 19], augment LiDAR points with either RGB features [20], augment RGB images with LiDAR points [21] or add semantic information obtained by processing RGB inputs [22, 23]. Others propose stage-wise methods that first detect boxes over images and localize in 3D with LiDAR [24] and fuse detections from single-modal detectors [25, 26]. 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 [27], 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, 28, 29]. Existing methods propose reweighting losses [30, 31, 32, 33, 34, 35], rebalancing data sampling [36, 37, 38], balancing gradients computed from imbalanced classes [39], and balancing network weights [29]. Others study LTP through the lens of 2D object detection over RGB images [40]. Compared to 2D image-based recognition, 3D long-tailed detection has unique opportunities and challenges because sensors such as LiDAR directly provide geometric and ego-motion cues that are difficult to extract from 2D images. 2D detectors must detect objects of different scales due to perspective image projection, dramatically increasing the complexity of the output space (e.g., requiring more anchor boxes). In contrast, 3D objects do not exhibit as much scale variation, but far-away objects tend to be sparse, imposing different challenges. Finally, 3D detectors often use class-aware heads (i.e. each class has its own binary classifier) while 2D long-tail recognition typically use shared softmax heads (which may make it easier to enforce hierarchies, as explained above). To the best of our knowledge, long-tailed 3D detection (LT3D) has not yet been explored. 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]nand CenterPoint [13]), an RGB-based detector (FCOS3D [41]), and multimodal detector (TransFusion [17]). We further introduce several modifications that consistently improve their LT3D performance. Please refer to the supplement for training details.

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 and are difficult to scale for a large number of classes. Therefore, we first consider making each class its own group to avoid hand-crafted grouping heuristics. However, the multi-head architecture has per-class detectors that consist of multiple layers with lots of parameters, hence learning them easily overfits to rare-classes. Our final solution is to merge all classes into a single group with a proportionally heavier detector head to simplify training. Our group-free (i.e. single-head) architecture has a shared backbone across all classes, and each class has only one linear layer as its detector. This significantly reduces the number of parameters and allows learning the shared feature backbone collaboratively with all classes, effectively mitigating overfitting to rare-classes. Adding a new class is as simple as adding a single linear layer to the detector head. Our grouping-free detector head achieves improved accuracy over grouping-based methods, as shown in the supplement.

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

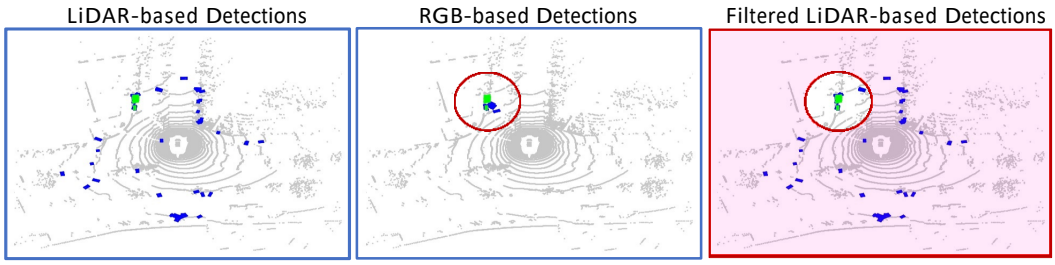

Figure 2: Multimodal filtering effectively removes high-scoring false-positive LiDAR detections. The green boxes are ground-truth strollers, while the blue boxes are stroller detections from our best performing models, liDAR-based detector CenterPoint [13] (left) and RGB-based detector FCOS3D [41] (mid). 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. We adopt a grouping-free detector head that outputs separate "multitask" heatmaps for each class, and use a per-class sigmoid focal loss rather than multi-class cross-entropy loss. It is worth noting that this simple "multitask" learning strategy does not necessarily enforce a hierarchy, hence can extend to more complex label relationships. 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. We explore alternatives in the supplement and conclude that they achieve similar LT3D performance. Perhaps surprisingly, this training method improves detection performance not only for rare classes, but also for common classes.

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 [22, 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-hoc 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 keep LiDAR-based detections within a radius of m meters and remove all the others (that are not close to any RGB-based detections). We use FCOS3D [41] 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. 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 instances per class), medium (5k50k), and few (<5k). We describe the metrics below.

Standard Detection Metrics. Mean average precision (mAP) is an established metric for object detection [42, 1, 43]. 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 but 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. 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 [44], 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. For clarity, we will describe the procedure in context of computing AP_H for some arbitrary class C.

LCA=0: Consider the 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 [43]. All other predictions for C are false positives.

LCA=2: Consider the predictions for C and ground-truth boxes for C and all sibling classes of C (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 for C 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] and CenterPoint [13], which are widely benchmarked in the literature. For fusion-based 3D detectors, we use TransFusion [17], a recently released state-of-the-art method. TransFusion proposes an end-to-end DETR-like approach [45] for multimodal 3D detection.

Implementation. We use the PyTorch toolbox [46] to train all models for 20 epochs with the Adam optimizer [47] and a one-cycle learning rate scheduler [48]. 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. We further describe our implementation in the supplement.

Datasets. We use nuScenes [2] and Argoverse 2.0 (AV2) [49] to explore LT3D. Both have fine-grained classes (18 and 26 classes in nuScenes and AV2 respectively) that follow long-tailed distributions. To quantify the long-tail, we use the imbalance factor (IF) defined as the ratio between the numbers of annotations of the max-class and min-class [32]; nuScenes and AV2 have IF=1670 and 2500 respectively – significantly more imbalanced than existing long-tail image recognition benchmarks, e.g., iNaturalist (IF=500) [50] and ImageNet-LT (IF=1000) [51]. NuScenes arranges classes in a semantic hierarchy (Fig. 3); AV2 does not provide a semantic hierarchy but we construct one based on the nuScenes' hierarchy. Following prior work, we use their official train-sets for training and evaluate on the their official val-sets. We focus on nuScenes in the main paper and AV2 in the supplement. Our primary conclusions hold for both datasets.

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

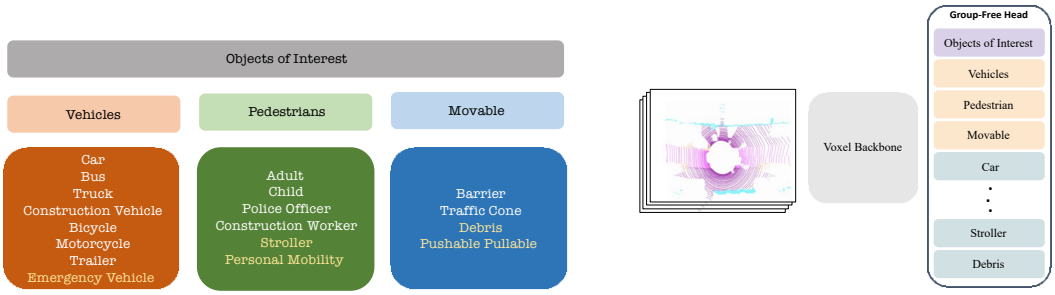

Figure 3: nuScenes defines a semantic hierarchy (on the left) for all annotated classes (Fig. 1). We highlight common classes in white and rare classes in gold. The standard nuScenes benchmark makes two choices for dealing with rare classes: (1) ignore them (e.g., stroller and pushable-pullable), or (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 a sigmoid focal loss that learns separate spatial heatmaps for each class.

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 by 1% AP averaged over All classes. Second, multimodal filtering yields between 411 AP improvement on Medium and Few classes! This is surprising given that FCOS3D is a less powerful 3D detector on its own. Importantly, we do not modify FCOS3D for LT3D. Interestingly, it also improves multimodal detectors (1.6% AP improvement for 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 the multi-modal TransFusion model. 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 (RGB-only) [41]		39.0	23.3	2.9	20.9
PointPillars (LiDAR-only) [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/ Multimodal Filtering	X	66.2	41.0	4.4	35.8
CenterPoint (LiDAR-only) [13]		76.4	43.1	3.5	39.2
+ Hierarchy		77.1	45.1	4.3	40.4
w/ Data Aug.		73.8	44.5	7.4	40.3
w/ Multimodal Filtering	X	77.1	49.0	9.4	43.6
TransFusion-L (LiDAR-only)[17]		68.5	42.8	8.4	38.5
w/ Multimodal Filtering	X	73.2	42.5	8.3	39.6
TransFusion (LiDAR + RGB)	X	73.9	41.2	9.8	39.8
w/ Data Aug.	X	73.4	40.9	8.2	39.0
w/ Multimodal Filtering	X	73.9	42.5	9.1	40.1

Retraining state-of-the-art 3D detectors for LT3D. We retrain several 3D detectors, namely FCOS3D [41], PointPillars [8] 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 18 classes as defined by the nuScenes LT3D protocol. As shown in Table 1, LiDAR-based detectors that perform well on common classes tend to also perform well on rare classes.

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

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, (Table 2), 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 (OTS)	LCA=0	82.4	81.2	49.4	19.7	33.6	48.9	0.1	14.2	0.1	21.7
	LCA=1	83.9	82.0	58.7	20.5	35.2	50.5	0.1	18.3	0.1	22.0
	LCA=2	84.0	82.4	58.8	20.7	36.4	51.0	0.1	19.5	0.1	22.6
CenterPoint (Group-Free)	LCA=0	88.1	86.3	62.7	24.5	48.5	62.8	0.1	22.2	4.3	32.7
	LCA=1	89.0	87.1	71.6	26.7	50.2	64.7	0.1	29.4	4.5	32.9
	LCA=2	89.1	87.5	71.7	26.8	51.1	65.2	0.1	30.5	4.8	33.4
CenterPoint (Group-Free) w/ Hierarchy	LCA=0	88.6	86.9	63.4	25.7	50.2	63.2	0.1	25.3	8.7	36.8
	LCA=1	89.5	87.6	72.4	27.5	52.2	65.2	0.1	32.4	9.4	37.0
	LCA=2	89.6	88.0	72.5	27.7	53.2	65.7	0.1	34.0	9.8	37.6
CenterPoint (Group-Free) w/ Hier. & MM. Filtering	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-L (LiDAR-only)	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 (LiDAR + RGB)	LCA=0	84.4	84.2	58.4	24.5	46.7	60.8	3.1	21.6	13.3	25.3
	LCA=1	86.0	85.4	67.3	26.3	50.1	63.5	14.4	34.7	20.6	25.6
	LCA=2	86.0	85.9	67.4	26.8	52.2	65.1	15.2	36.1	22.8	26.4
TransFusion w/ Multimodal 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

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%, Medium examples by 2% and Few examples by 1% AP.

Data Augmentation Schedule. Prior works [25, 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 Fusion via Filtering. Detecting rare classes from LiDAR-only is challenging since its difficult to recognize objects from sparse LiDAR points alone. 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, improving by 411 AP on classes with Medium and Few examples for all models (Table 1)!

End-to-End Multimodal Methods. Since multimodal cues significantly improve LT3D, we are motivated to explore end-to-end approaches. Specifically, we retrain TransFusion [17] on all 18 classes. We retrain TransFusion, downsampling the RGB images by a factor of 2 to fit the model in GPU memory. Surprisingly, lidar-only and multi-modal variants of TransFusion perform worse than CenterPoint for LT3D. Further hyperparameter tuning and full-resolution training may help. Lastly, our multimodal filtering strategy still improves this end-to-end fusion methods slightly, e.g., it increases mAP on All classes by 0.3% AP for TransFusion (cf. Table 1). We also find that applying multimodal filtering on the LiDAR-only branch of TransFusion yields similar performance to the multimodal model, suggested that TransFusion is simply learning a multimodal filtering function.

Analysis of Misclassifications. For 3D detection, localization and classification are two important measures of 3D detection performance. In practice, we cannot achieve perfect performance for either. In safety-critical applications, detecting but misclassifying objects (as a semantically related category)

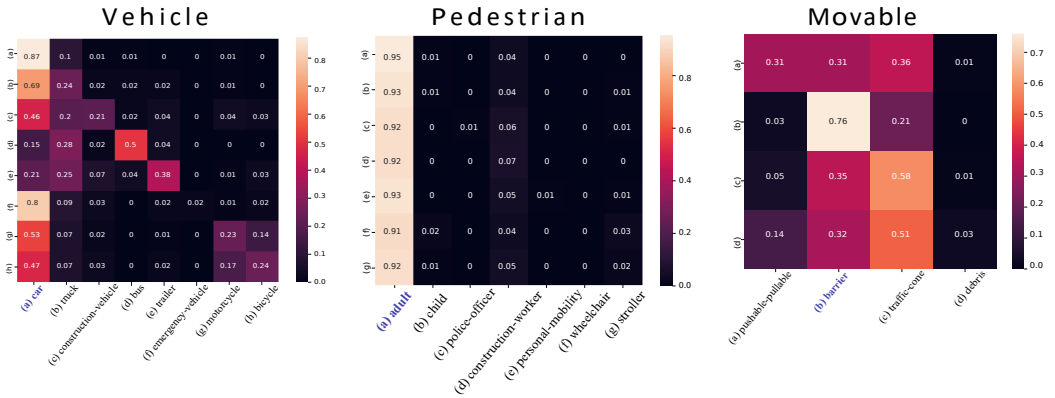

Figure 4: Breakdown analysis of misclassifications within superclasses. We analyze our best-performing model (CenterPoint w/ hierarchical training and multimodal filtering). Fine-grained classes are most often confused by the dominant class (in blue) in each superclass: (left) Vehicle is dominated by car, (mid) Pedestrian is dominated by adult, and (right) Movable is dominated by barrier. We find that class confusions are reasonable. Car is often mistaken for truck. Similarly, truck, construction-vehicle and emergency-vehicle are most often mistaken for car. Bicycle and motorcycle are sometimes misclassified as car, presumably because they are sometimes spatially close (within the 2m match threshold) to cars. Adults have similar appearance to police-officer and construction-worker, and they are often co-localized with child and stroller; all of these might cause significant class confusion.

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. Applying this hierarchical AP reveals that classes are most often misclassified as their LCA=1 siblings within coarse-grained superclasses. We use confusion matrices to further analyze the misclassifications within superclasses, as shown in Fig. 4. Below, we explain how to compute a confusion matrix for the detection task.

For each superclass, we make a confusion matrix, in which the entry (i; j) indicates the misclassification rate of class-i objects as class-j. Specifically, given a fine-grained class i, we find its predictions that match ground-truth boxes within 2m center-distance of class-i and all its sibling classes (LCA=1, within the corresponding superclass); we ignore all unmatched detections. This allows us to count the mis-classifications of class-i objects into class-j, with which a simple normalization produces misclassification rates.

Limitations. 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.

6 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), but has broad robotic applications, e.g., elder-assistive robots [52] that fetch diverse items [53] should address LT3D. To study LT3D, we establish rigorous evaluation protocols that allow for partial credit to better diagnose 3D detectors. 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.

Acknowledgement

This work was supported by the CMU Argo AI Center for Autonomous Vehicle Research.

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. Krahenbuhl. 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, D. Ramanan, and S. Kong. Multimodal object detection via probabilistic ensembling. In European Conference on Computer Vision (ECCV), 2022.

[19] S. Gupta, J. Kanjani, M. Li, F. Ferroni, J. Hays, D. Ramanan, and S. Kong. Far3det: Towards far-field 3d detection. In NeurIPS, 2022.

[20] 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.

[21] 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.

[22] 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.

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

[24] 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.

[25] 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.

[26] 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.

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

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

[29] 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.

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

[31] 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.

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

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

[34] 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.

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

[36] 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.

[37] 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.

[38] 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.

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

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

[41] 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.

[42] 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.

[43] 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.

[44] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3):211–252, 2015.

[45] 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.

[46] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, 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.

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

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

[49] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, D. Ramanan, P. Carr, and J. Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In Neural Information Processing Systems Datasets and Benchmarks Track, 2021.

[50] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie. The inaturalist species classification and detection dataset. In CVPR, 2018.

[51] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu. Large-scale long-tailed recognition in an open world. In CVPR, pages 2537–2546, 2019.

[52] N. Savage. Robots rise to meet the challenge of caring for old people. Nature, 2022.

[53] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, M. Martin, T. Nagarajan, I. Radosavovic, S. K. Ramakrishnan, F. Ryan, J. Sharma, M. Wray, M. Xu, E. Z. Xu, C. Zhao, S. Bansal, D. Batra, V. Cartillier, S. Crane, T. Do, M. Doulaty, A. Erapalli, C. Feichtenhofer, A. Fragomeni, Q. Fu, C. Fuegen, A. Gebreselasie, C. Gonzalez, J. Hillis, X. Huang, Y. Huang, W. Jia, W. Khoo, J. Kolar, S. Kottur, A. Kumar, F. Landini, C. Li, Y. Li, Z. Li, K. Mangalam, R. Modhugu, J. Munro, T. Murrell, T. Nishiyasu, W. Price, P. R. Puentes, M. Ramazanova, L. Sari, K. Somasundaram, A. Southerland, Y. Sugano, R. Tao, M. Vo, Y. Wang, X. Wu, T. Yagi, Y. Zhu, P. Arbelaez, D. Crandall, D. Damen, G. M. Farinella, B. Ghanem, V. K. Ithapu, C. V. Jawahar, H. Joo, K. Kitani, H. Li, R. Newcombe, A. Oliva, H. S. Park, J. M. Rehg, Y. Sato, J. Shi, M. Z. Shou, A. Torralba, L. Torresani, M. Yan, and J. Malik. Ego4d: Around the world in 3, 000 hours of egocentric video. Computer Vision and Pattern Recognition 2022.

[54] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In ICCV, 2017.

[55] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, 2015.

[56] C. J. Wu, M. Tygert, and Y. LeCun. A hierarchical loss and its problems when classifying non-hierarchically. PLoS ONE, 14, 2019.

[57] Y. Li, T. Wang, B. Kang, S. Tang, C. Wang, J. Li, and J. Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[58] A. E. Welchman, V. L. Tuck, and J. M. Harris. Human observers are biased in judging the angular approach of a projectile. Vision research, 44(17):2027–2042, 2004.

[59] 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 Workshops, 2021.

Appendix

A Implementation Details

We follow the training procedure of the respective detectors [8, 13, 17] which have open-source code. We describe important implementation details below.

- Model Architecture. We adopt the architecture in [9] but make an important modification. The original architecture (for the standard nuScenes benchmark) has six heads designed for ten classes; each head has 64 filters. We first adapted this architecture for LT3D using seven heads designed for 18 classes. We then replace these seven heads with a single head consisting of 512 filters shared by all classes.
- Training Losses. We use the sigmoid focal loss (for recognition) [54] and L1 regression loss (for localization) below. Existing works also use the same losses but only with fine labels; we apply the loss to both coarse and fine labels. Concretely, our loss function for CenterPoint is as follows: $L = L_{HM} + L_{REG}$, where $L_{HM} = \sum_{i=0}^{C} SigmoidFocalLoss(X_i; Y_i)$ and $L_{REG} = |X_{BOX} \quad Y_{BOX}|$, where X_i and Y_i are the i^{th} class' predicted and ground-truth heat maps, while X_{BOX} and Y_{BOX} are the predicted and ground-truth box attributes. Without our hierarchical loss, C=18. With our hierarchical loss, C=22 (18 fine grained + 3 coarse + 1 object class). is set to 0.25. Modifications for other detectors similarly follow.
- Post-processing. We use non-maximum suppression (NMS) on detections within each class to suppress lower-scoring detections. In contrast, existing works apply NMS on all detections across classes, i.e., suppressing detections overlapping other classes' detections (e.g., a pedestrian detection can suppress other pedestrian and traffic cone detections).

B Analysis on Multi-Head Architectures and Class Grouping

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. We study the impact of grouping for both the standard and LT3D problem setups. We define the groups used for this study below. Each group is enclosed in curly braces. Our group-free head includes all classes into a single group.

- Original: {Carg, Truck, Construction Vehicleg, Bus, Trailerg, Barrierg, Motorcycle, Bicycleg, Pedestrian, Traffic Coneg}
- LT3D: {Carg, Truck, Construction Vehicleg, Bus, Trailerg, Barrierg, Motorcycle, Bicycleg, Adult, Child, Construction Worker, Police Officer, Traffic Coneg, Pushable Pullable, Debris, Stroller, Personal Mobility, Emergency Vehicleg}

We use the class groups proposed by prior works [9, 13] for the standard benchmark and adapt this grouping for LT3D. Our proposed group-free detector head architecture consistently outperforms grouping-based approaches on both the standard and LT3D benchmarks. We note that sub-optimal grouping strategies (such as those adopted for LT3D) may yield significantly diminished performance, whereas optimized grouping strategies (such as those adopted for the standard setup) have comparable performance to the group-free approach. The group-free approach simplifies architecture design, while also providing competitive performance.

Two insights allow us to train the group-free architecture. First, we make the group-free head proportionally larger to train more classes. The standard grouping setup contains 6 heads, each with 64 convolutional filters. Scaling up to the nearest power of two, our group-free head has 512 convolutional filters. Second, we do not perform between-class NMS. The standard setup performs NMS between classes in each group (e.g., since pedestrians and traffic cones are tall and skinny, the model should only predict that an object is either a traffic cone or a pedestrian). However, performing NMS between classes requires that confidence scores are calibrated, which is not the case. Moreover, for LT3D, score calibration becomes more important for `rare` classes as these classes have lower confidence scores than common classes on average, meaning that common objects will

Table 3: Our proposed group-free detector head architecture consistently outperforms grouping-based approaches on both the standard and LT3D benchmarks. We note that sub-optimal grouping strategies (such as those adopted for LT3D) may yield significantly diminished performance, whereas optimized grouping strategies (such as those adopted for the standard setup) have comparable performance to the group-free approach. Note, TC is traffic-cone, CV is construction vehicle, MC is motorcycle, PP is pushable-pullable, CW is construction-worker, and PO is police-officer. We highlight classes with Medium and Few examples per class in blue.

CenterPoint	Multi-Head	Car	Ped.	Barrier	TC	Truck	Bus	Trailer	CV	MC	Bicycle
Original	X	87.7	87.7	70.7	74.0	63.6	72.7	45.1	26.3	64.7	47.9
		89.1	88.4	70.8	74.3	64.8	72.9	42.0	25.7	65.9	53.6
for LT3D	X	82.4	—	62.0	60.1	49.4	55.7	28.9	19.7	48.9	33.6
		88.1	—	72.4	72.7	62.7	70.8	40.2	24.5	62.8	48.5

		Adult	PP	CW	Debris	Child	Stroller	PO	EV	PM	All
Original	X	—	—	—	—	—	—	—	—	—	64.0
		—	—	—	—	—	—	—	—	—	64.8
for LT3D	X	81.2	21.7	14.2	1.1	0.1	0.1	1.3	0.1	0.1	31.2
		86.3	32.7	22.2	4.3	0.1	4.3	1.8	10.3	0.1	39.2

likely suppress rare objects within the same group. Our solution is to only perform within-class NMS, which is standard for 2D detectors [55].

C Ablation on Hierarchical Training and Inference

Classic methods train a hierarchical softmax (in contrast to our simple approach of sigmoid focal loss with both fine and coarse classes), where one multiplies the class probabilities of the hierarchical predictions during training and inference [56]. We implemented such an approach, but found the training did not converge. Interestingly, [56] shows such a hierarchical softmax loss has little impact on long-tailed object detection (in 2D images), which is one reason they have not been historically adopted. Instead, we found better results using the method from [57] (a winning 2D object detection system on the LVIS [40] benchmark) which multiples class probabilities of predictions (e.g. $P_{CAR} = P_{OBJ} \; P_{CAR}$) at test-time, even when such predictions are not trained with a hierarchical softmax. We tested three variants and compared it to our approach (which recall, uses only fine-grained class probabilities at inference). Table 4 compares their performance for LT3D.

Table 4: Different variants achieve similar performance. We note that other methods do improve accuracy in the tail by sacrificing performance in the head, suggesting that hybrid approaches that apply different techniques for head-vs-tail classes may further improve accuracy. Unlike [57, 56] which requires a strict label hierarchy, our approach is not limited to a hierarchy.

Method	Hierarchy	Many	Medium	Few	All
CenterPoint (w/o Hierarchy) [13]	n/a	76.4	43.1	3.5	39.2
CenterPoint w/ Hierarchy	(a)	77.1	45.1	4.3	40.4
	(b)	76.4	45.0	5.3	40.5
	(c)	76.5	45.2	5.2	40.6
	(d)	74.5	43.5	5.6	39.5

(a) Ours (e.g., Finegrain score only)

(b) Object score * Finegrain score ([57], e.g. $P_{CAR} = P_{OBJ} \; P_{CAR}$)

(c) Coarse score * Finegrain score (Variant-1 of [57], e.g. $P_{CAR} = P_{VEHICLE} \; P_{CAR}$)

(d) Object score * Coarse score * Finegrain score (Variant-2 of [57], e.g. $P_{CAR} = P_{OBJ} \; v \; P_{EHICLE} \; P_{CAR}$)

Unlike [57, 56] which require a strict label hierarchy, our approach is not limited to a hierarchy. We find that other hierarchical methods improve accuracy in the tail by sacrificing performance in the head, suggesting that hybrid approaches that apply different techniques for head-vs-tail classes may further improve accuracy.

D Evaluating on Argoverse 2.0

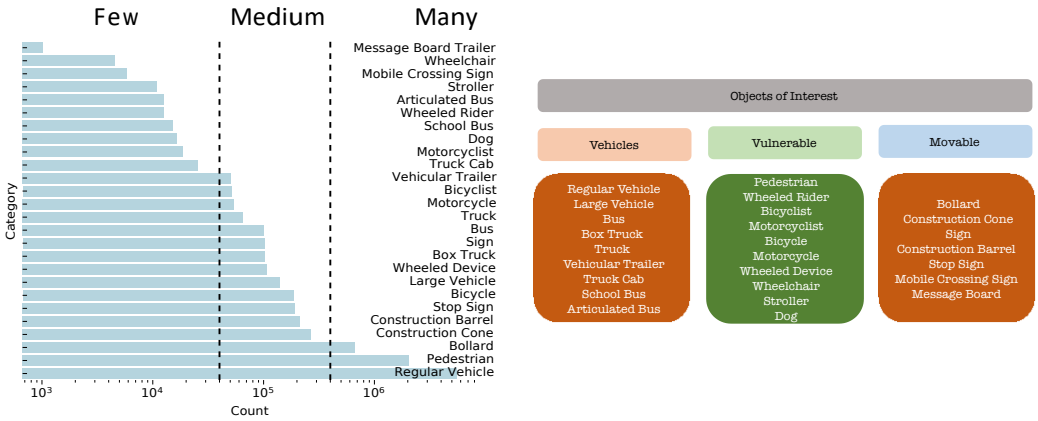

Figure 5: Left: According to the histogram of per-class object counts (on the left), classes in Argoverse 2.0 (AV2) follow a long tailed distribution. Following [7] and nuScenes (Fig. 1), we report performance for three groups of classes based on their cardinality (split by dotted lines): Many, Medium, and Few. As AV2 does not provide a class hierarchy, we construct one by referring to the nuScenes hierarchy (cf. Fig. 5 on the right).

We present results on the large-scale Argoverse 2.0 (AV2) dataset, another long-tailed dataset developed for autonomous vehicle research (Fig. 5 on the left). AV2 evaluates on 26 classes, which follow the long-tailed distribution. As AV2 does not provide a semantic hierarchy, we construct one (cf. Fig. 5 on the right) by adapting the nuSccenes hierarchy. As show in Table 5, our main conclusions still hold on AV2. Compared to the CenterPoint reported [49], our implementation improves mAP by 14.4 by carefully adopting LiDAR sweep aggregation and class-balanced sampling. Based on our implementation, exploiting hierarchical semantics improves mAP from 44.0 to 47.0. This improves performance for both common and rare classes alike. Interestingly, FCOS3D performs significantly worse than the LiDAR based detectors, yet multimodal filtering improves mAP to 48.4 mAP. These new results on AV2 are consistent with those on nuScenes, demonstrating the general applicability of our approach.

Table 5: Results (mAP in %) on Argoverse 2.0 (AV2) evaluated at 50 meters. Compared to the CenterPoint reported in [58], our implementation improves mAP by 14.4 by carefully adopting LiDAR sweep aggregation and class-balanced sampling. Based on our implementation, exploiting hierarchical semantics improves mAP from 44.0 to 47.0. Note that AV2 does not provide a semantic hierarchy; we construct one based on the nuScenes hierarchy (Fig. 3). Interestingly, FCOS3D performs significantly worse than the LiDAR based detectors, yet multimodal filtering improves mAP to 48.4 mAP. These new results on AV2 are consistent with those on nuScenes (cf. Table 1), demonstrating the general applicability of our approach.

Method	Multimodal	Many	Medium	Few	All
FCOS3D [59] (RGB-only)		27.4	17.0	7.8	14.6
CenterPoint (LiDAR-only) [13] reported by [49]		66.7	32.9	14.1	29.6
CenterPoint (LiDAR-only) [Our Implementation]		77.4	46.9	30.2	44.0
+ Hierarchy		79.0	50.3	33.6	47.0
+ Multimodal Filtering	X	79.0	51.4	35.3	48.4