Long-Tailed 3D Detection via Multi-Modal Fusion

Neehar Peri

CMU-RI-TR-23-58

June 15

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Deva Ramanan, *advisor*
Katerina Fragkiadaki
Tarasha Khurana

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

To my parents.

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 only focus on 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 detection performance 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 multi-modal cues are crucial to long-tailed 3D detection.

We empirically show that (a) high-resolution RGB images help recognize rare objects, (b) LiDAR provides precise 3D localization, and (c) uni-modal detectors can be trained with more diverse examples because they do not require aligning and annotating multi-modal data. With these insights, we propose a simple late-fusion framework that combines RGB and LiDAR detections. We examine three critical components in this framework and consider whether to train 2D or 3D RGB detectors, whether to match RGB and LiDAR detections in the 2D image plane or 3D bird's-eye-view (BEV), and how to fuse matched detections. Our modifications improve accuracy by 12.2% AP on average for all classes, and dramatically improve AP for `rare` classes (e.g., `stroller` AP improves from 0.1 to 37.7)!

Funding

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

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

Introduction

3D object detection is a key component in many robotics systems such as autonomous vehicles (AVs) [4, 16]. To facilitate research in this space, the AV industry has released large-scale 3D annotated multi-modal datasets [4, 7, 56]. However, these datasets often only benchmark on a few `common` classes such as `pedestrian` and `car`. In the real open world, safe navigation [57, 67] requires AVs to also reliably detect `rare` objects such as `child` and `stroller`. This motivates our study of *Long-Tailed 3D Detection* (LT3D), a problem that requires detecting objects from both `common` and `rare` classes.

Status Quo. Among contemporary AV datasets, nuScenes [4] has exhaustively annotated objects of various classes crucial to AVs (Fig. 1.1) and organizes them with a semantic hierarchy (Fig. 3.1). 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 (LT3D) by re-purposing *all* annotated classes in nuScenes. Importantly, this challenging new problem is not simply solved by training state-of-the-art (SOTA) methods on more classes [43], e.g., TransFusion [2], a SOTA multi-modal transformer-based detector, achieves only 3.0 AP on the rare `child` class despite attaining 84.4 AP on the common `car` class.

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`).

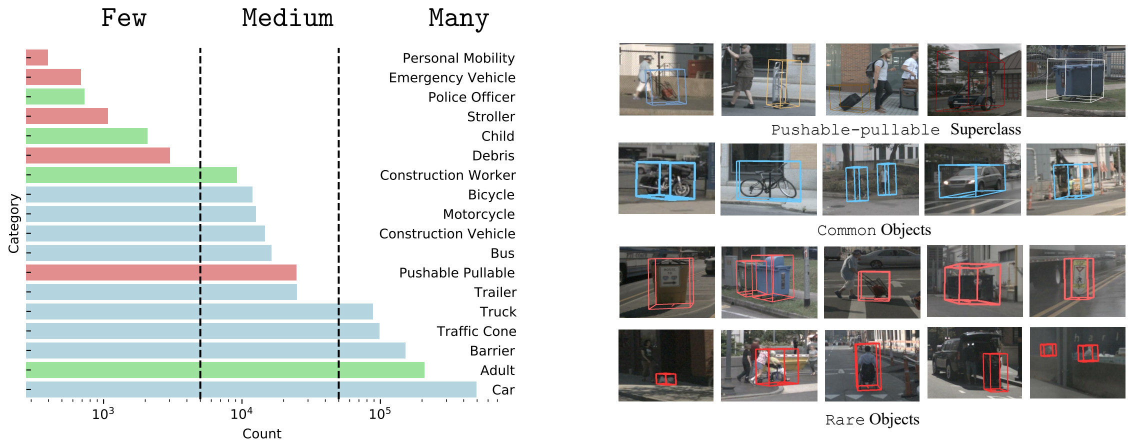

Figure 1.1: **nuScenes Dataset Statistics**. According to the histogram of per-class object counts (on the **left**), the nuScenes benchmark focuses on the common classes in cyan (e.g., `car` and `barrier`) but ignores rare ones in red (e.g., `stroller` and `debris`). In fact, the benchmark creates a 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 behavior. Following [41], we report performance for three groups of classes based on their cardinality (split by dotted lines): `Many`, `Medium`, and `Few`.

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 that exploits inter-class relationships to award partial credit (Fig. 3.1). 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 small tail objects, such as `strollers`. We propose

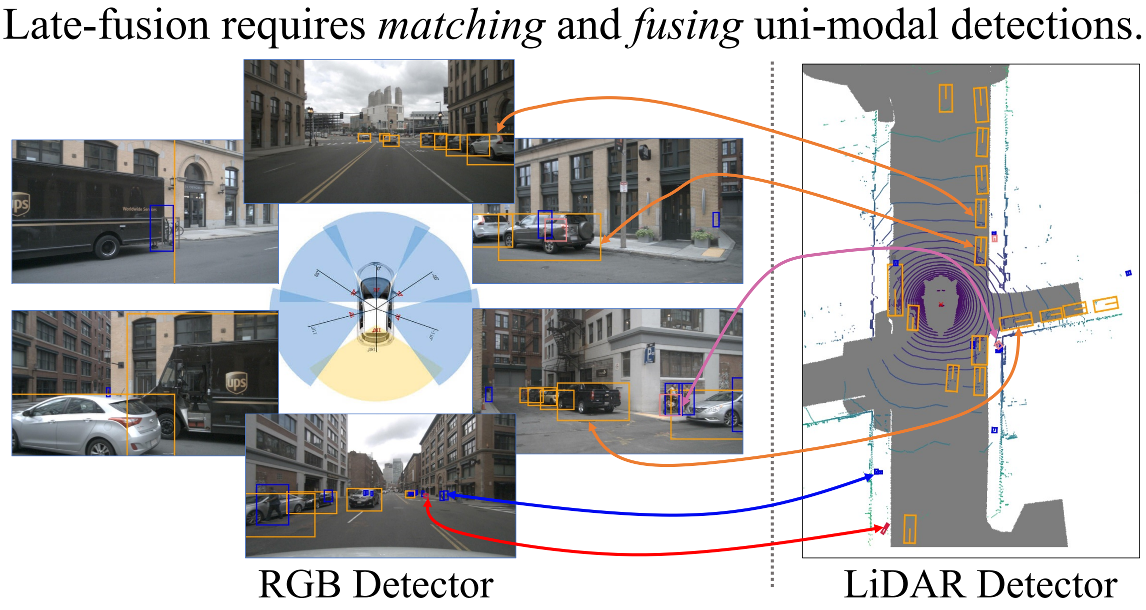
Late-fusion requires *matching* and *fusing* uni-modal detections.
RGB Detector
LiDAR Detector

Figure 1.2: **Late Fusion Overview**. We extensively explore the simple late-fusion framework for LT3D by ensembling RGB and LiDAR uni-modal detectors [43]. We rigorously examine three critical components within this framework (Fig. 1.3) and propose a simple method that fuses detections produced by a 2D RGB-detector (e.g., DINO [74]) and a 3D LiDAR-detector (e.g., CenterPoint [72]). Our method achieves 51.4 mAP on LT3D benchmarks based on the well-established nuScenes [4] dataset, significantly improving over baselines by 12.2% (Table 4.1).

a simple late-fusion framework (Fig. 1.2) and study three critical design choices (Fig. 1.3). First, we propose a simple approach that post-processes LiDAR-based 3D detections with monocular RGB-based 3D detections, filtering away detections that are inconsistent across modalities. This significantly improves 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`). Next, we evaluate the impact of using 2D RGB detectors instead of monocular 3D RGB detectors for late-fusion, and find that the former is straightforward to train, can easily leverage external data, and leads to higher AP averaged over all classes. This is practically meaningful because annotating 2D boxes on RGB images is significantly cheaper than aligning multi-modal RGB-LiDAR data and annotating them with 3D amodal cuboids. Fourth, we consider the impact of matching RGB and LiDAR detections on the 2D image plane instead of the 3D bird's-eye-view (BEV). We contrast 2D matching in the image-plane with prior work that performs 3D matching by lifting

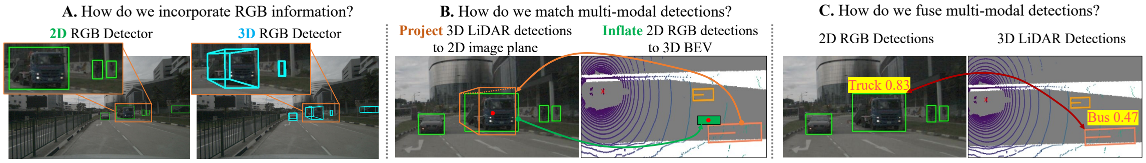
A. How do we incorporate RGB information?
2D RGB Detector
3D RGB Detector
B. How do we match multi-modal detections?
Project 3D LiDAR detections to 2D image plane
Inflate 2D RGB detections to 3D BEV
C. How do we fuse multi-modal detections?
2D RGB Detections
3D LiDAR Detections
Truck 0.83
Bus 0.47

Figure 1.3: **Examining Late Fusion Strategies**. We examine three key components in effectively fusing RGB and LiDAR detectors. We explore: **A.** whether to train 2D or 3D monocular RGB detectors for late-fusion, **B.** whether to match multi-modal detections in the 2D image plane or 3D bird's-eye-view (BEV), and **C.** how to optimally fuse matched detections. Perhaps surprisingly, our exploration reveals that using 2D RGB detectors, matching in the 2D image plane, and fusing scores probabilistically with calibration leads to better LT3D performance.

2D detections to 3D (e.g., by relying on depth imputed from LiDAR points that project into the 2D detections [43, 65]) and find that 2D matching is more robust. Lastly, we explore score calibration prior to fusion. We find that calibrating our detection scores improves rare class detection and enables probabilistic fusion of LiDAR and RGB detections. Notably, this boosts performance compared to the standard non-maximum suppression (NMS) fusion strategy.

Contributions. We make three major contributions. First, we formulate the problem of LT3D, emphasizing detection of both `common` and `rare` classes in safety-critical applications like 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 generalizes across different RGB and LiDAR architectures. We conduct extensive experiments to ablate our design choices and demonstrate that our simple method achieves state-of-the-art results on LT3D benchmarks.

Chapter 2

Related Works

2.1 3D Object Detection for AVs

Contemporary approaches for 3D object detection can be broadly classified as LiDAR-only, RGB-only, and sensor-fusion methods. Recent work in 3D detection is heavily inspired by prior work in 2D detection [6, 38, 77]. LiDAR-based detectors like PointPillars [30], CBGS [80], and PVRCNN++ [55] adopt an SSD-like architecture [38] that regresses amodal bounding boxes from a bird's-eye-view (BEV) feature map. More recently, CenterPoint [72] adopts a center-regression loss that is inspired by CenterNet [77]. Despite significant progress, LiDAR-based detectors often produce many false positives because it is difficult to distinguish foreground objects from background given sparse LiDAR returns. Monocular RGB-based methods have gained popularity in recent years due to increased interest in camera-only perception. FCOS3D [63] extends FCOS [59] by additionally regressing the size, depth, and rotation for each object. More recently, methods such as BEVDet and BEVFormer [24, 25, 33] construct a BEV feature-map by estimating the per-pixel depth of each image feature [44].

PolarFormer [26] introduces a polar-coordinate transformation that improves near-field detection. Importantly, many of these state-of-the-art 3D RGB detectors are commonly pre-trained on large external datasets like DDAD [18]. Monocular RGB detectors accurately classify objects but struggle to estimate depth, particularly for far-field detections [21]. Despite recent advances in LiDAR and RGB 3D detectors, we find that multi-modal fusion is essential for LT3D (detailed next). Importantly, using both RGB (for better recognition) and LiDAR (for better 3D localization) helps detect rare classes. We study the late-fusion framework described in Fig 1.2 to determine how to effectively fuse RGB and LiDAR uni-modal detectors for LT3D.

2.2 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 has high-resolution (but lacks 3D information). Multimodal fusion for 3D detection is an active field of exploration. Popular approaches can be categorized as input-fusion, feature-fusion, and late-fusion. Input-fusion methods typically augment LiDAR points using image-level features. For example, PointPainting [61] projects LiDAR points onto the output mask of a semantic segmentation model and appends corresponding class scores to each point. MVP [73] densifies regions of LiDAR sweeps that correspond with objects in semantic segmentation masks. In contrast, Frustum PointNets [46] leverage 2D RGB detections to localize objects within the box frustum using PointNets [45].

Recent works show that feature-fusion can be more effective than input-fusion. PointFusion [69] fuses global image and point-cloud features prior to detection and MSMDFusion [27] fuses LiDAR and RGB features at multiple scales. TransFusion [2] and BEVFusion [39] fuse features in the BEV space using multi-headed attention. Despite the success of transformers for detecting common objects, [43] finds that TransFusion struggles to detect rare classes. We posit that the transformer architecture, as adopted in TransFusion and BEVFusion, suffers from limited training data (particularly for classes in the long tail). For transformers to work well in practice, they should be trained on diverse, large-scale datasets [12, 47]. Further, end-to-end trained multi-modal detectors require paired multi-modal data for training. Therefore, we opt to study late fusion of uni-modal detectors, which do not require aligned RGB-LiDAR paired training data.

CLOCs [42] is a late-fusion method that learns a separate network to fuse RGB and LiDAR detections, showing promising results for 3D detection. More recently, Peri et. al. [43] introduces a simple non-learned filtering algorithm that effectively removes false-positive LiDAR-detections based on proximity to a 3D RGB detection. We delve into this simple (non-learned) late-fusion framework, study three crucial design choices, and present a method that significantly outperforms the state-of-the-art for LT3D.

2.3 Long-Tailed Perception

AV datasets follow a long-tailed class distribution: a few classes like `car` and `pedestrian` are dominant, while others like `stroller` and `debris` are rarely seen. However, this problem is not unique to the AV domain. [50]. Long-Tailed Perception (LTP) is a long-standing problem in the literature [41] and has been widely studied through the lens of image classification, aiming for high accuracy averaged across imbalanced classes [1, 41, 76].

Existing methods propose reweighting losses [5, 10, 23, 28, 29, 75], rebalancing data sampling [8, 13, 22], balancing gradients computed from imbalanced classes [58], and balancing network weights [1]. Others study LTP through the lens of 2D object detection with RGB images [20]. 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. Further, 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 have sparse LiDAR returns, 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 approaches typically use shared softmax heads.

Recently, CBGS [80] explicitly addresses rare-class 3D detection by up-sampling LiDAR-sweeps with instances of rare classes, and pasting instances of rare objects copied from different scenes. Although this works well for improving detection of infrequently seen classes (e.g. classes with `medium` number of examples like `bicycle` and `construction vehicle`), it does not provide significant improvement for classes with only a `few` examples like `debris` and `stroller`. Additionally, rare classes, such as `child` and `stroller`, are typically small in size and have a limited number of

LiDAR returns. As a result, LiDAR-only detectors may struggle to accurately detect these rare classes. In LT3D, we find a unique challenge: rare classes are not only infrequent but are also difficult to distinguish using LiDAR alone. We address the problem of LT3D in this work by fusing RGB and LiDAR uni-modal detectors.

Chapter 3

Method

To address LT3D, we first retrain SOTA 3D detectors on *all* classes, including LiDAR-based detectors (PointPillars [30] and CenterPoint [72]), RGB-based detectors (FCOS3D [63], PolarFormer [26], BEVFormer [33], YOLOV7 [62], and DINO [74]), and multimodal detectors (TransFusion [2], BEVFusion [39], and DeepInteraction [71]). We further introduce several modifications that consistently improve their LT3D performance.

3.1 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, learning a class-specific head easily overfits to rare-classes. Our solution is to merge all classes into a single group with a proportionally heavier (single) 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 per-class. This significantly reduces the number of

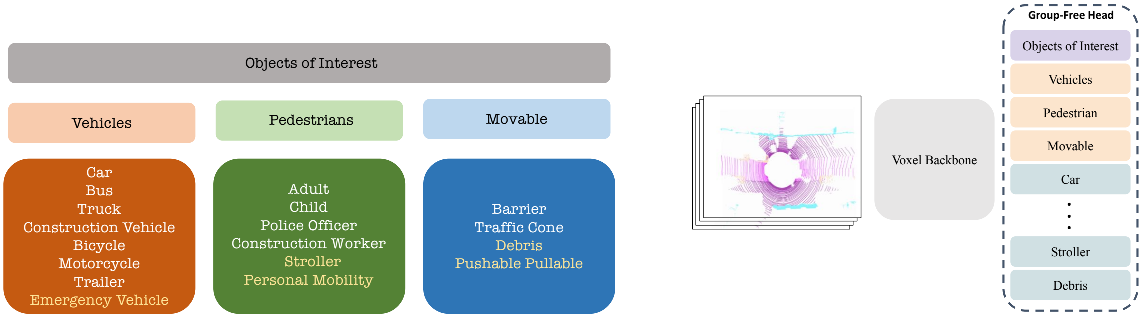

Figure 3.1: **nuScenes Semantic Hierarchy**. nuScenes defines a semantic hierarchy (on the **left**) for all annotated classes (Fig. 1.1). We highlight `common` classes in white and `rare` classes in gold. The standard nuScenes benchmark makes two choices for dealing with rare classes: (1) ignore them (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. 1.1), 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.

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. In addition, we show that our grouping-free detector head achieves improved accuracy over grouping-based methods.

3.2 Training with Semantic Hierarchies

nuScenes defines a semantic hierarchy (Fig. 3.1) 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 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, and can be 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 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.

3.3 Augmentation Schedule

Class-balanced resampling is a common technique in learning with long-tailed distributions. 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 [2, 61] 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. This further suggests that strategies that work for common classes may not work in the long-tail, further emphasizing the need to study LT3D.

3.4 Late-Fusion of RGB and LiDAR for LT3D

As depicted in Fig. 1.2, our simple late-fusion framework ensembles uni-modal RGB and LiDAR detectors respectively. Within this framework, we investigate three crucial design choices previewed in Fig. 1.3. We first describe the benefits and drawbacks of using 2D and monocular 3D RGB detectors in Sec 3.4.1, present simple algorithms for matching RGB and LiDAR-detections in Sec. 3.4.2, and finally describe score calibration and fusion in Sec. 3.4.3.

3.4.1 How Do We Incorporate RGB Information?

Although LiDAR offers accurate localization, contemporary LiDAR detectors predict numerous false positives due to the challenging task of distinguishing foreground objects from the background using sparse LiDAR points alone. RGB images provide complementary information that is essential for identifying objects and disambiguating semantically similar classes. Therefore, we focus on identifying which RGB detectors can be best fused with 3D LiDAR detectors. We compare the impact of using monocular 3D RGB detectors and 2D RGB detectors below.

2D RGB Detectors. 2D object detection is a fundamental problem in computer vision [15, 35, 51] that has matured in recent years. Large-scale 2D detection datasets are widely available, and model trade-offs are well understood [37, 38, 49, 51]. As 2D detectors do not predict 3D attributes like depth and rotation, understanding how to best leverage 2D detectors in the context of long-tailed 3D detection is a key challenge. In this work, we consider two state-of-the-art 2D RGB detectors, YOLOV7 [62] and DINO [74]. YOLOV7 is a real-time detector that identifies a number of training tricks that nearly doubles the inference efficiency over prior work without sacrificing performance. Similarly, DINO is a recent transformer-based detector that improves upon DETR [6] using denoising anchor boxes.

3D RGB Detectors. RGB-based 3D object detection is more complex than conventional 2D detection, as it requires additional predictions such as depth and orientation [3, 63]. Importantly, 3D RGB detection is an ill-posed problem due to the inconsistency between the 2D input data and the 3D output predictions. To address this problem, FCOS3D transforms the commonly defined 7-DoF 3D targets to the image domain and decouples them as 2D and 3D attributes [63]. Moreover, 3D RGB detection is challenging because it relies on accurate sensor extrinsics to transform 3D detections between the global and image coordinate frame. Since annotating 3D amodal cuboids is both expensive and non-trivial (compared to bounding-box annotations for 2D detection), datasets for monocular 3D RGB detection are considerably smaller and less diverse than their 2D detection counterparts. For example, nuScenes (published in 2020) annotates 144K RGB images of 23 classes [4] while COCO (an early 2D detection dataset published in 2014) annotates 330K images [35] of 80 classes.

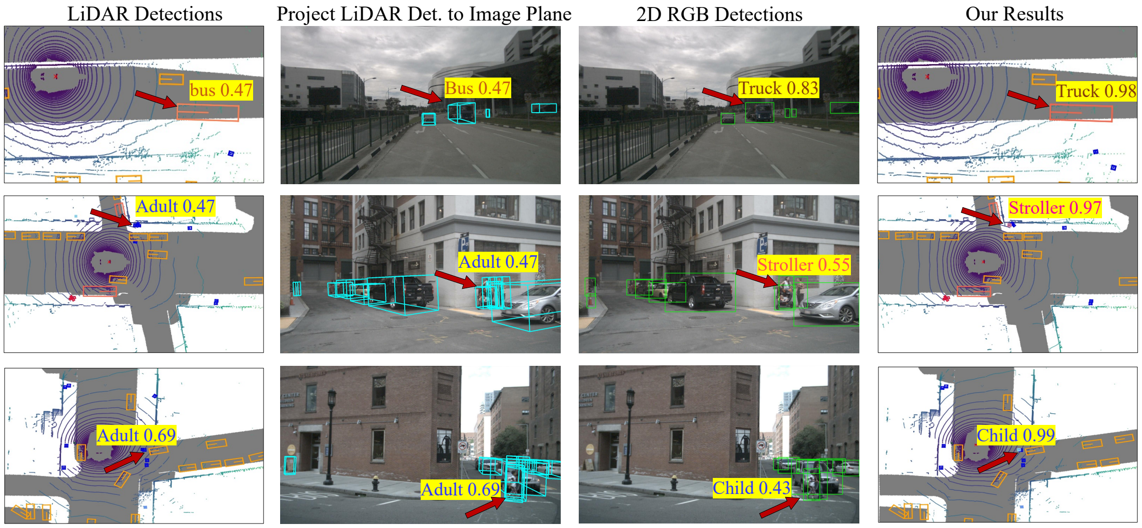

Figure 3.2: **Qualitative Improvement From Multi-Modal Late Fusion**. Late-fusion of 2D RGB and 3D LiDAR detections improves LT3D performance. Projecting 3D LiDAR detections onto the image-plane makes matching RGB and LiDAR detections more robust. In contrast, matching inflated 2D RGB detections in the 3D BEV is more challenging due to noisy depth estimates. We find that late-fusion is able to boost the confidence score of LiDAR-based detections when both LiDAR and RGB detections agree, and correct labels when they don't agree. For example, our late fusion algorithm correctly relabels predictions with semantically similar (according to the nuScenes labeling hierarchy [43]), but visually distinct classes like `adult` and `stroller`, or `adult` and `child`.

Although adapting these 2D detectors for multi-modal filtering of 3D LiDAR-based detections is challenging, training 2D RGB detectors only requires *2D bounding box* annotations, which is significantly cheaper to collect than 3D cuboids used for 3D RGB-detector training [63]. In addition, 2D RGB detectors can leverage diverse, publicly available 2D detection datasets to train better 2D detectors [31, 48, 64, 70, 79]. We further demonstrate that using "freely available" 2D detection datasets helps train stronger 2D detectors that further improve LT3D performance. Particularly, when scaling up the 2D training data, late-fusion boosts average performance by 2.6 mAP, and improves rare class mAP by 3.5%.

3.4.2 How Do We Match Multi-Modal Detections?

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 that a simple strategy of post-hoc fusion works remarkably well. Finding correspondence between two sets of uni-modal detections is an essential step prior to late-fusion (Fig. 1.3**A**). However, this matching process is non-trivial when considering RGB and LiDAR-based detections. We present two approaches for matching between modalities, and empirically evaluate the effectiveness of each method.

Option 1: Spatial Matching in the 2D Image Plane. We explore two potential implementations of this below. Using the provided sensor extrinstics, we can project 3D LiDAR detections onto the 2D image plane [42]. Next, we use the IoU metric to determine overlap between (projected) LiDAR and 2D RGB detections. We determine that a 2D RGB detection and (projected) 3D LiDAR detection match if the IoU is greater than a fixed threshold. Although conceptually simple, we find that it works well.

In principle, we can project 3D RGB detections onto the 2D image plane, but we find that using 2D RGB detections works better in practice.

Option 2: Spatial Matching in 3D BEV. We explore two potential implementations of this below. First, we can use a 3D RGB 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. 3.3 demonstrates this matching 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).

Similarly, matching 2D RGB detections in the 3D BEV is an ill-posed problem because it is impossible to precisely estimate depth using a single monocular RGB image. Instead, we inflate the 2D RGB detection using the mediod of the LiDAR points within the frustum of the 2D predicted boxes [46]. We find that it is crucial to filter out LiDAR returns from (far-away) background points (c.f. Fig. 4.1). Specifically, since LiDAR points on the edge of an object produce a depth discontinuity (adding noise to the depth prediction), we opt to estimate depth using points in a small region around the center of the bounding box.

We empirically evaluate both options and find that spatially matching 2D RGB detections and 3D LiDAR detections in the 2D image plane works best in practice.

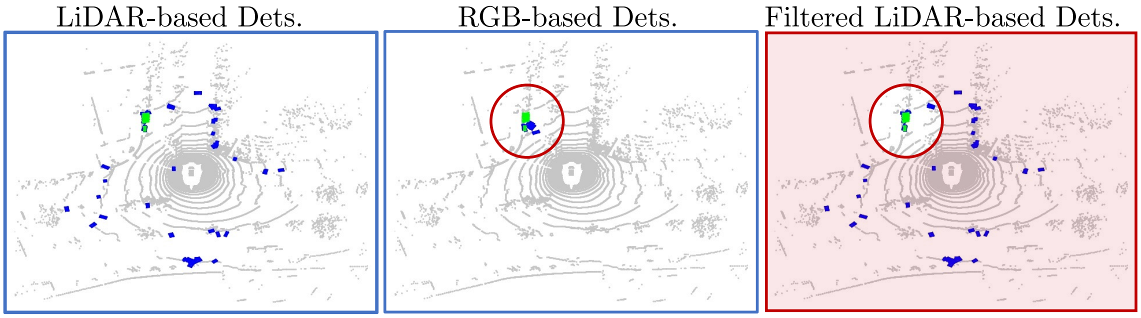

Figure 3.3: **Multi-Modal Filtering**. Spatial matching in 3D BEV effectively removes high-scoring false-positive LiDAR detections. The green boxes are ground-truth `strollers`, while the blue boxes are `stroller` detections from our best performing models, LiDAR-based detector CenterPoint [72] (**left**) and RGB-based detector FCOS3D [63] (**mid**). The final filtered result removes LiDAR detections not within m meters of any RGB detection (**right**).

Handling Unmatched Detections. After spatially matching RGB and LiDAR detections, we often have three categories of detections to consider: matched detection, unmatched RGB detections, and unmatched LiDAR detections. For 2D RGB-detections that do not match with any LiDAR-detections, we simply remove these predictions. Since LiDAR detectors achieve high recall, any RGB detections that are unmatched are likely to be false positives On the other hand, for 3D LiDAR detections that do not match with any RGB-detections, we down-weight their confidence scores by multiplying by w which is tuned via validation ($w = 0.4$).

Semantic Matching. As illustrated by Fig. 1.3**C**, detections may match spatially, but not semantically. To address this, we propose a semantic matching heuristic to better fuse LiDAR and RGB detections. Given a pair of spatially matched RGB and LiDAR detections, we consider two cases. If both modalities predict the same semantic class, we perform score-fusion (which we describe next). Otherwise, if both modalities predict different semantic classes, we use the confidence score *and* label of the RGB prediction. Intuitively, we expect that RGB detectors can more reliably predict semantics from high resolution images. This simple method helps correct misclassifications produced by the 3D LiDAR-detector, as shown in Fig. 3.2.

3.4.3 How Do We Fuse Multi-Modal Detections?

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 address these FPs by either removing them via multi-modal filtering [43], or down-weighting their confidence via score calibration and fusion. We empirically evaluate the effectiveness of each method and find that score calibration and fusion works the best.

Score Calibration and Fusion. Score calibration of matched detections produced by different uni-modal detectors is required to accurately compare detections w.r.t their confidence scores for late-fusion. We explore score calibration of matched RGB detections x_{RGB} and LiDAR detections x_{LiDAR} in the context of late-fusion (cf. Fig. 1.3**C**) below. Following [9], we assume independent class prior $p(c)$, and conditional independence given the class label, i.e., $p(x_{\text{RGB}}, x_{\text{LiDAR}}|c) = p(x_{\text{RGB}}|c)p(x_{\text{LiDAR}}|c)$. We denote the posteriors for class-c as $p(c|x_{\text{RGB}})$ and $p(c|x_{\text{LiDAR}})$, and the fused score / posterior $p(c|x_{\text{LiDAR}}, x_{\text{LiDAR}})$. We have

$$
\begin{aligned}
& p(c|x_{\text{LiDAR}}, x_{\text{LiDAR}}) && && (3.1)\\
= \; & \frac{p(x_{\text{LiDAR}}, x_{\text{LiDAR}}|c)p(c)}{p(x_{\text{LiDAR}}, x_{\text{LiDAR}})} && \text{Bayes Rule} && (3.2)\\
\propto \; & p(x_{\text{LiDAR}}, x_{\text{LiDAR}}|c)p(c) && && (3.3)\\
\propto \; & \frac{p(c|x_{\text{RGB}})p(c|x_{\text{LiDAR}})}{p(c)} && \text{Conditional Independence} && (3.4)\\
\propto \; & \frac{p(c|x_{\text{RGB}})p(c|x_{\text{LiDAR}})}{p(c)} && && (3.5)
\end{aligned}
$$

This suggests that optimal calibration requires tuning class prior $p(c)$. However, tuning class priors $p(c)$ is exponentially expensive w.r.t an mAP measure. Therefore, we tune them greedily, one by one ordered by class cardinality. Further, we also tune a temperature on the logits [9, 19]. The overall score calibration improves LT3D performance from 44.6 to 45.0 in mAP. After calibrating all classes, we fuse scores for matched uni-modal detections. Inspired by [9], we explore probabilistic fusion and non-maximal suppression (NMS). Intuitively, fusing scores with NMS is equivalent to performing a max-pooling operation on matched detections. In contrast, if two detections fire on the same object, the fused score should be larger than the individual scores because there is more evidence. We find that probabilistic fusion results in an additional 0.5 AP improvement averaged over all classes compared to NMS.

Chapter 4

Experiments

We conduct extensive experiments to better understand the LT3D problem, and gain insights by validating our techniques described in Chapter 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?

4.1 Implementation Details.

We follow the training procedure of the respective detectors which have open-source code. We describe important implementation details below.

- *Input.* We adopt 10-frame aggregation for LiDAR densification when training LiDAR-based detectors on nuScenes and a 5-frame aggregation on Argoverse 2. We assume that we are provided with ego-vehicle poses for prior frames to align all LiDAR sweeps to the current ego-vehicle pose. Since LiDAR returns are sparse, this densification step is essential for accurate 3D detection. By default, we train the 2D RGB detectors on the 2D bounding boxes derived by projecting

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

3D annotations to the 2D image plane and additionally train with 2D bounding boxes from nuImages where denoted. Our 2D RGB detectors YOLOV7 and DINO are pre-trained on the ImageNet [11] and COCO [34] datasets.

- *Model Architecture.* We adopt the architecture in [80] 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) [37] 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: $L = L_{HM} + \lambda L_{REG}$, where $L_{HM} = \sum_{i=0}^{C} SigmoidFocalLoss(X_i, Y_i)$ and $L_{REG} = |X_{BOX} - Y_{BOX}|$, 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). λ is set to 0.25. Modifications for other detectors similarly follow.
- *Optimization.* We train all LiDAR-only detectors for 20 epoch using an AdamW optimizer and a cyclic learning rate. We adopt a basic set of data augmentations, including global 3D tranformations, flip in BEV, and point shuffling during training. We train our model with 8 RTX 3090 GPUs and a batch size of 1 per GPU. The training noise (from random seed and system scheduling) is $<$ 1% of the accuracy (standard deviation normalized by the mean).
- *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).

Datasets. We use nuScenes [4] and Argoverse 2 (AV2) [66] 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 calculate the imbalance factor (IF), defined as the ratio between the numbers of annotations of the max-class and min-class [5]; nuScenes and AV2 have IF=1670 and 2500 respectively – significantly more imbalanced than existing long-tail image recognition benchmarks, e.g., iNaturalist (IF=500) [60] and ImageNet-LT (IF=1000) [40]. NuScenes arranges classes in a semantic hierarchy (Fig. 3.1); AV2 does not provide a semantic hierarchy but we construct one based on the nuScenes' hierarchy. Following prior work, we use official train-sets for training and evaluate on the official val-sets.

4.2 Evaluation Metrics

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. 1.1-left): *many* (>50k instances 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 [14, 16, 36]. 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 [4]. 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., detecting but misclassifying a `child` as an `adult` is better than 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. 3.1) 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 [53], but have not been extensively 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` [36]. 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.

4.3 nuScenes Results

We first start by evaluating existing uni-modal and multi-modal models on the popular nuScenes dataset. Benchmarking SOTA models yields poor performance for `rare` classes on the nuScenes-LT3D benchmark, highlighting the importance of addressing 3D detection in the long tail setting rather than only focusing on common categories.

Retraining State-Of-The-Art 3D Detectors for LT3D. We retrain several 3D detectors, namely FCOS3D [63], PolarFormer [26], BEVFormer [33], PointPillars [30], CenterPoint [72], TransFusion [2], and DeepInteraction [71]. FCOS3D, PolarFormer and BEVFormer operate on monocular images. PointPillars and CenterPoint take an aggregated stack of LiDAR-sweeps as input. TransFusion and DeepInteraction take both RGB frames and LiDAR sweeps as input. All models predict 3D bounding

Table 4.1: **Comparison with the nuScenes State-of-the-Art**. We find that our late-fusion approach of fusing 3D LiDAR and 2D RGB detections in the 2D image plane using score calibration and probabilistic ensembling performs the best on *all* categories, notably improving performance for classes with `medium` and `few` examples.

Method	MM	`Many`	`Medium`	`Few`	`All`
FCOS3D (RGB-only) [43]		39.0	23.3	2.9	20.9
BEVFormer (RGB-only) [33]		52.3	31.6	1.4	27.3
PolarFormer (RGB-only) [26]		54.0	31.6	2.2	28.0
PointPillars (LiDAR-only) [30]		64.2	28.4	3.4	30.0
CenterPoint (LiDAR-only) [43]		76.4	43.1	3.5	39.2
TransFusion (LiDAR + RGB) [2]	✓	73.9	41.2	9.8	39.8
DeepInteraction (LiDAR + RGB) [71]	✓	76.2	51.1	7.9	43.7
Multi-Modal Late-Fusion (Ours)	✓	**77.9**	**59.4**	**20.0**	**51.4**

boxes for 18 classes as defined by the nuScenes LT3D protocol. As shown in Table 4.1, LiDAR-based detectors perform well on `common` classes, but struggle on classes with `few` examples. This is unsurprising as it is difficult to identify rare objects from sparse LiDAR points alone. However, we find that multi-modal models achieve strong performance across all cateogires.

End-to-End Multi-Modal Methods. We find that TransFusion [2] and DeepInteraction [71], which fuses both RGB and LiDAR, are able to perform better than the LiDAR-only models, suggesting that multi-modal input can improve object detection by removing false positives. TransFusion marginally improves over CenterPoint overall, but provides considerable performance gains for `rare` classes, improving by 6.4%. DeepInteraction provides modest improvements over TransFusion, notably improving on classes with `many` examples by 2.3% and `medium` examples by 9.9%. Although DeepInteraction beats CenterPoint by 3.5% overall, it requires complex multi-stage training pipelines and paired multi-modal data. We aim to simplify multi-modal training via late fusion, which we describe below.

Multi-Modal Late-Fusion. Our late-fusion approach combines 3D LiDAR detections with 2D RGB detections in the 2D image plane using score calibration and probabilistic ensembling. We compare our approach against other methods in Table 4.1 and provide qualitative results in Fig. 3.2. By carefully considering design choices outlined in Fig. 1.3, we are able to improve over the prior state-of-the-art by

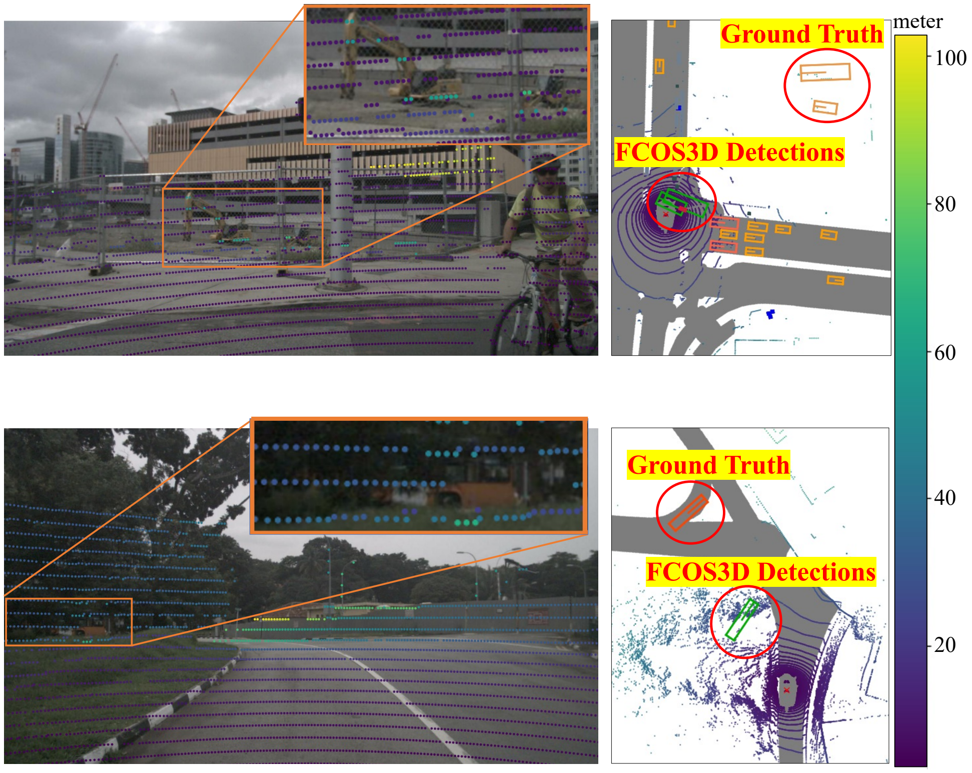

Figure 4.1: **Failure Mode of Inflating 2D Detections via LiDAR Points**. We find that 3D BEV filtering using inflated 2D detections does not work well due to noisy depth predictions. For example, background points on fences and on trees leads to imperfect depth prediction from inflation. Attempting to filter 3D LiDAR detections using these noisy inflated 2D RGB detections in the BEV introduces many missed-detections and false positives.

12.2%. As shown in Fig. 3.2, we find that RGB-based depth predictions are often incorrect, leading to suboptimal matching and filtering. Naively using LIDAR depth to inflate 2D detections into the 3D BEV for matching and filtering yields poor results due to noisy LiDAR returns (c.f. Fig. 4.1). As a result, we simply opt to project all detections onto the 2D image plane to factor out the impact of depth estimation on matching.

Table 4.2: **nuScenes Per-Class Breakdown.** Multi-modal models like DeepInteraction and our late-fusion approach achieve the highest per-class AP on 8 out of 10 classes shown below. Out late-fusion approach significantly improves over DeepInteraction, improving bicycle accuracy by 5.8%, construction worker by 15.2 %, stroller by 6.8 %, and pushable-pullable by 17.3 %. Note, CV is construction vehicle, MC is motorcycle, PP is pushable-pullable, CW is construction-worker, and Stro. is stroller. We highlight classes with Medium and Few examples per class in blue.

Method	Car	Adult	Truck	CV	Bicy	MC	Child	CW	Stro.	PP
FCOS3D [63]	52.1	46.5	28.7	10.0	31.4	37.2	2.1	20.2	4.4	26.6
CenterPoint[72]	**87.7**	**86.7**	61.6	28.4	49.6	65.9	1.1	28.5	5.1	34.9
TransFusion [2]	84.4	84.2	58.4	24.5	46.7	60.8	3.1	21.6	13.3	25.3
DeepInteraction [71]	84.9	85.9	**63.2**	35.3	64.3	**76.2**	6.0	30.7	30.9	30.8
Ours	86.3	86.2	60.6	**35.3**	**70.1**	75.9	**8.8**	**55.9**	**37.7**	**58.1**

Table 4.3: **Comparison with the Argoverse 2 State-of-the-Art**. We present results AV2 evaluated at 50 meters. FCOS3D achieves poor performance, likely due to inaccurate depth estimates. In contrast, CenterPoint achieves strong performance on all classes. Our multi-modal fusion approach significantly improves over CenterPoint, achiving 8.3% improvement averaged over all classes. These results on AV2 are consistent with those on nuScenes (cf. Table 4.1), demonstrating the general applicability of our approach.

Method	Multimodal	Many	Medium	Few	All
FCOS3D [63] (RGB-only)		27.4	17.0	7.8	14.6
CenterPoint [72] (LiDAR-only)		77.4	46.9	30.2	44.0
Multi-Modal Late Fusion (Ours)	✓	**89.4**	**54.2**	**38.7**	**52.3**

4.4 Argoverse 2 Results

We present results on the large-scale Argoverse 2 (AV2) dataset, another long-tailed dataset developed for autonomous vehicle research (Fig. 4.2 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. 4.2 on the **right**) by adapting the nuSccenes hierarchy. As show in Table 4.3, our main conclusions from nuScenes still hold for AV2. FCOS3D yields poor performance on all classes, likely due to inaccurate depth estimates. CenterPoint performs considerably better, achieving high accuracy on classes with many examples. Notably, CenterPoint performs better on AV2's rare classes (30.2 AP) compared to nuScenes's rare classes (3.5 AP), likely

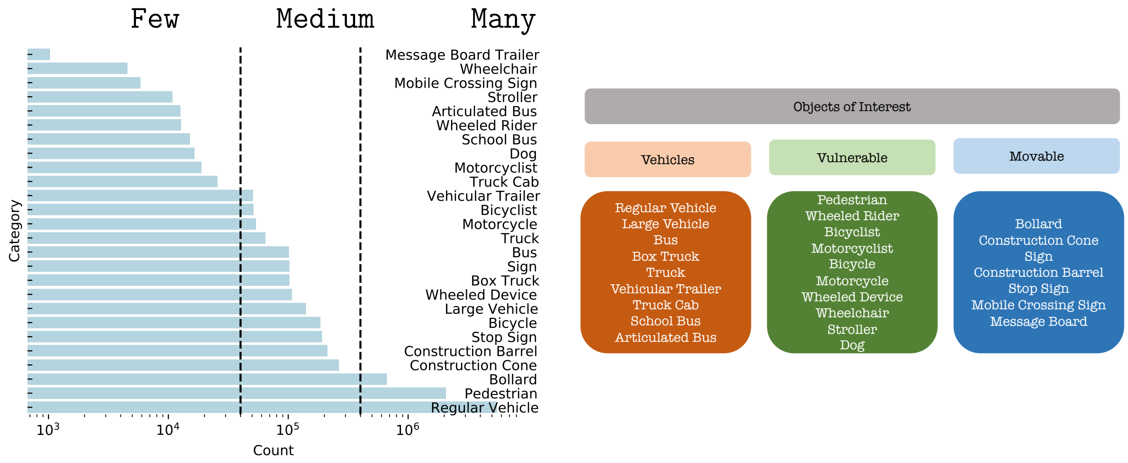

Figure 4.2: **Argoverse 2 Dataset Statistics**. According to the histogram of per-class object counts (on the **left**), classes in Argoverse 2.0 (AV2) follow a long tailed distribution. Following [41] and nuScenes (Fig. 1.1), we report performance for three groups of classes based on their cardinality (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. 4.2 on the **right).**

because AV2 has more examples per-class in-the-tail. Lastly, our proposed late-fusion approach yields an 8.3% improvement overall, improving performance for classes of all cardinalities. These new results on AV2 are consistent with those on nuScenes, demonstrating the general applicability of our approach.

4.5 Ablation Studies

We design a set of experiments to understand the impact of hierarchies, network architecture, and ablate different strategies for late fusion. We perform all ablation experiments on the nuScenes dataset.

4.5.1 Analysis on Hierarchies

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 develop

Table 4.4: **Impact of Semantic Hierarchies and Data Aug.** (measured by mAP). Training with the semantic hierarchy improves all methods for LT3D, e.g., improving by 1% AP averaged over `All` classes. 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`
PointPillars (LiDAR-only) [30]		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
CenterPoint (LiDAR-only) [72]		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

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).

Training with Semantic Hierarchy. 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 4.4, improving accuracy for classes with `Many` examples by 2%, `Medium` examples by 2% and `Few` examples by 1%.

Data Augmentation Schedule. Prior works [2, 69] 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).

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) 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

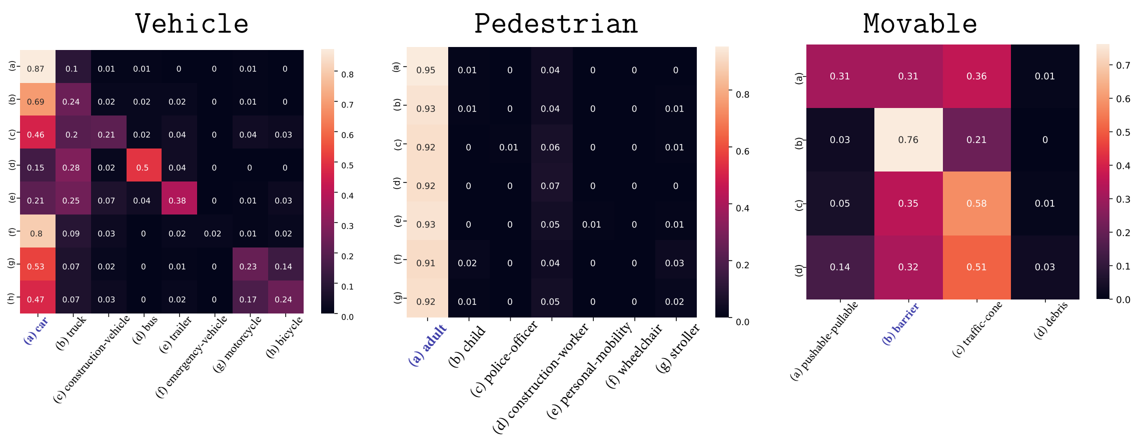

Figure 4.3: **Breakdown Analysis of Misclassifications within Superclasses.** Fine-grained classes are most often confused by the dominant class (in blue) in each superclass: (**left**) `Vehicle` is dominated by `car`, (**mid**) `Pedestrian` is dominated by `adult`, and (**right**) `Movable` is dominated 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.

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. 4.3. 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 misclassifications of class-i objects into class-j.

Impact of 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 [68]. We implemented such an

Table 4.5: **Diagnosis using the mAP$_H$ metric on selected classes**. We analyze the best-performing LiDAR-only model CenterPoint with and without our hierarchical loss. Comparing the rows of LCA=0, 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, 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 *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

approach, but found the training did not converge. Interestingly, [68] 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 [32] (a winning 2D object detection system on the LVIS [20] 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 additional variants and compared it to our approach (which recall, uses only fine-grained class probabilities at inference). Table 4.6 compares their performance for LT3D.

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

(b) Object score * Finegrain score ([32], e.g. $P_{CAR} = P_{OBJ} * P_{CAR}$)

(c) Coarse score * Finegrain score (Variant-1 of [32], e.g. $P_{CAR} = P_{VEHICLE} * P_{CAR}$)

(d) Object score * Coarse score * Finegrain score (Variant-2 of [32], e.g. $P_{CAR} = P_{OBJ} * P_{VEHICLE} * P_{CAR}$)

Unlike [32, 68] which require a strict label hierarchy, our approach is not limited

Table 4.6: **Impact of Hierarchical Softmax**. 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 [32, 68] which requires a strict label hierarchy, our approach is not limited to a hierarchy.

Method	Hierarchy	`Many`	`Medium`	`Few`	`All`
CenterPoint (w/o Hierarchy) [72]	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

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.

4.5.2 Analysis on Architecture

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: {`Car`}, {`Truck`, `Construction Vehicle`}, {`Bus`, `Trailer`}, {`Barrier`}, {`Motorcycle`, `Bicycle`}, {`Pedestrian`, `Traffic Cone`}
- LT3D: {`Car`}, {`Truck`, `Construction Vehicle`}, {`Bus`, `Trailer`}, {`Barrier`}, {`Motorcycle`, `Bicycle`}, {`Adult`, `Child`, `Construction Worker`, `Police Officer`, `Traffic Cone`}, {`Pushable Pullable`, `Debris`, `Stroller`, `Personal Mobility`, `Emergency Vehicle`}

We use the class groups proposed by prior works [72, 80] 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

Table 4.7: **Group-Free vs. Group-Based Architecture**. 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	MH	Car	Ped.	Barrier	TC	Truck	Bus	Trailer	CV	MC	Bicycle
Original	✓	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	✓	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	✓	—	—	—	—	—	—	—	—	—	64.0
		—	—	—	—	—	—	—	—	—	**64.8**
for LT3D	✓	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**

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

Table 4.8: **Ablation on Multi-Modal Fusion**. Our analysis confirms that 2D RGB are better suited for late-fusion (c-e vs. f), matching projected 3D LiDAR detections in the 2D image-plane outperforms matching 2D RGB detections inflated to the 3D BEV, and score calibration prior to probabilistic fusion improves performance.

	Method	`Many`	`Medium`	`Few`	`All`
(A)	CenterPoint [72]	76.4	43.1	3.5	39.2
(B)	+ Hierarchy	77.1	45.1	4.3	40.4
(C)	w/ 3D BEV Filtering w/ FCOS3D	76.6	48.7	8.1	42.9
(D)	w/ 3D BEV Filtering w/ BEVFormer	76.9	50.8	6.3	43.2
(E)	w/ 3D BEV Filtering w/ PolarFormer	76.8	50.0	6.1	42.8
(F)	w/ 3D BEV Filtering w/ YOLOV7	76.3	44.7	5.8	40.5
(G)	w/ 2D Img. Filtering using YOLOV7	77.0	51.3	9.8	44.6
(H)	+ Score Calibration	77.0	51.2	11.1	45.0
(I)	+ External Data	77.4	54.6	14.6	47.6
(J)	w/ 2D Img. Filtering using DINO	77.8	58.2	18.7	50.5
(K)	+ Prob. En.	77.9	59.4	20.0	51.4

average, meaning that `common` objects will likely suppress `rare` objects within the same group. Our solution is to only perform within-class NMS, which is standard for 2D detectors [52].

4.5.3 Analysis on Multi-Modal Fusion

We design a set of experiments to study the trade off between using 2D and monocular 3D RGB detectors, and matching in the 2D image and 3D BEV plane. Further, we examine the impact of using additional data and study different fusion strategies. Our analysis confirms that 2D RGB are better suited for late-fusion, matching projected 3D LiDAR detections in the 2D image-plane outperforms matching 2D RGB detections inflated to the 3D BEV, and score calibration prior to probabilistic fusion improves performance.

How Do We Incorporate RGB Information? 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

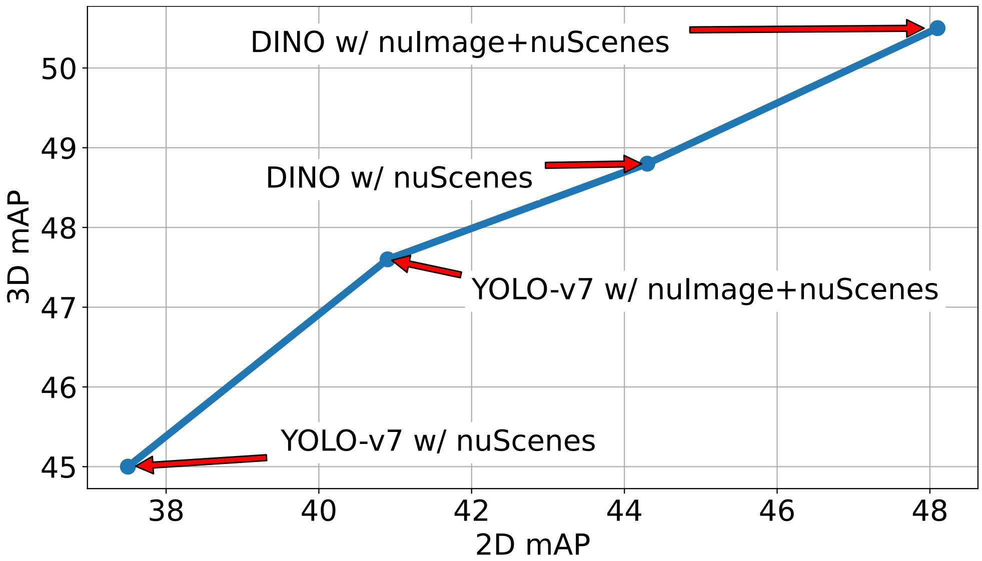

Figure 4.4: **Correlation Between 2D AP and 3D AP**. Although nuScenes is a 3D detection benchmark, we can generate 2D annotations using the provided sensor extrinsics by projecting the 3D annotations to the 2D image plane. We find that evaluating 2D detectors using these 2D nuScenes annotations is a good proxy task (x-axis) that is positively correlated with the downstream performance of the full late-fusion pipeline (y-axis). Concretely training 2D detectors with more data (e.g. training with nuScenes and nuImages), and using stronger 2D detectors (e.g. DINO) improves performance on the proxy task as well as the downstream late-fusion algorithm.

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 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). We first attempt to filter out 3D LiDAR detections in the BEV using monocular 3D detections (c.f. Table 4.8**C-E**). 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). Although this provides a 3% performance improvement over the LiDAR-only baseline Table 4.8**B**), we explore using 2D detectors as an alternative.

How Do We Match Multi-Modal Detections? We consider matching 3D LiDAR and 2D RGB detections in both the 3D BEV and 2D image plane. Naively lifting 2D RGB detections into 3D leads to imprecise depth estimates that leads to

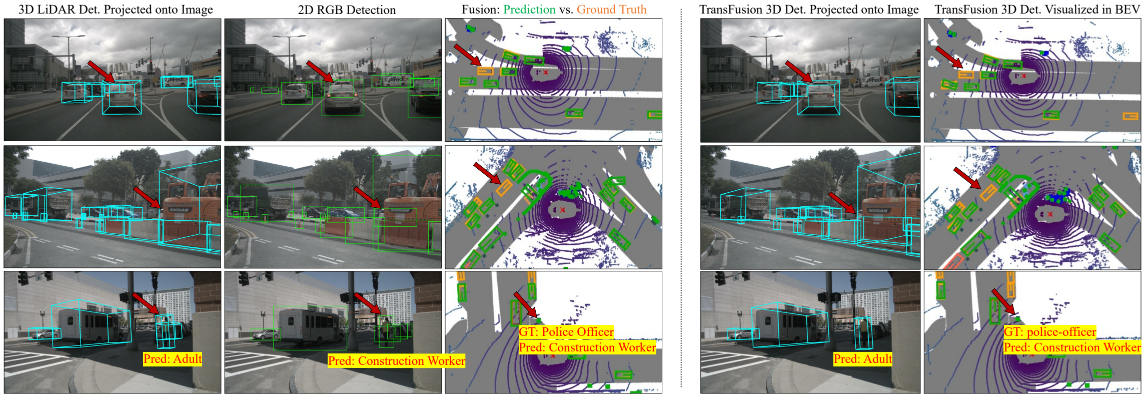
3D LiDAR Det. Projected onto Image
2D RGB Detection
Fusion: Prediction vs. Ground Truth
TransFusion 3D Det. Projected onto Image
TransFusion 3D Det. Visualized in BEV
Pred: Adult
Pred: Construction Worker
GT: Police Officer
Pred: Construction Worker
Pred: Adult
GT: police-officer
Pred: Construction Worker

Figure 4.5: **Failure Cases**. Both our method (columns 1-3) and TransFusion [2] (columns 4 -5) have the same failure cases. In the first and second row, the 2D RGB-detector DINO detects the heavily occluded cars but 3D LiDAR-detector fails to detect them. As a result, the late-fusion predictions miss these cars because our method throws away unmatched RGB-detections for which we do not have accurate 3D information. In the third row, we see that although both the LiDAR and RGB detectors fire on the object (whose ground-truth label is `police-officer`), LiDAR-detector predicts it as `adult` and RGB-detector predicts it as `construction-worker`. As a result, the final detection is incorrect w.r.t the predicted categorical label. TransFusion also misclassifies this object and predicts it as an `adult`.

missed-detections and false positives (Table 4.8**F**). As shown in Fig. 4.1, matching 3D LiDAR and 2D RGB detections in the 3D BEV does not work well in practice. Instead, we project the 3D LiDAR detections to the 2D image plane and filter using IoU (c.f. Table 4.8**G**). Two detections are considered matched if their spatial overlap exceeds a fixed threshold. We find that this approach performs considerably better than either of the approaches that perform fusion in the 3D BEV. Notably, we find that 2D image filtering with YOLOV7 improves over the 3D BEV filtering with YOLOV7 by 7.1% mAP and improves over the 3D BEV filtering with FCOS3D by 5.6% AP. Using better 2D detectors (e.g. DINO) and training with external data (c.f. Figure 4.4) yields better performance in both 3D BEV and 2D image based filtering (Table 4.8**I-J**).

How Do We Fuse Multi-Modal Detections? We evaluate multi-modal fusion using non-maximal suppression (NMS) and probabilistic ensembling (Prob. En.) (Table 4.8**J** vs. Table 4.8**K**). Prior to fusion, we first project all detections to the 2D image plane and calibrate the scores of LiDAR and RGB detections to ensure that they are comparable. Next, we pool both RGB and LiDAR detections together and match them based on their 2D IoU. If using NMS, only the highest confidence detections are kept and all lower confidence overlapping detections are removed. If using Prob. En. we use bayesian fusion to reasoning about the final score of ovelapping detections. Concretely, if two matched detections fire in the same place, the fused score should be higher than the individual scores because there is twice the evidence of an object at that particular spatial location. After score calibration, we find that Prob. En. achieves 0.5 mAP higher than NMS averaged over all classes. Notably, Prob. En. provides a considerable 1.3% AP improvement for rare classes.

Failure Cases and Visualizations We visualize common failure cases of our late-fusion approach and compare it with the failure cases of TransFusion, an end-to-end trained multi-modal detector. We find that our method fails in cases of occlusions (where there is no 3D information) and in cases where the 2D RGB-detector misclassifies the object. See Figure 4.5 for detailed analysis.

Chapter 5

Conclusion

In this work, 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 applications, (e.g., elder-assistive robots [54] that fetch diverse items [17] 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 effectively combines 2D RGB-based and 3D LiDAR-based detections, achieving significant improvements for LT3D. We find that 2D RGB detectors are better suited for late-fusion than monocular 3D RGB detectors, matching projected 3D LiDAR detections in the 2D image-plane outperforms matching 2D RGB detections inflated to the 3D BEV, and score claibration prior to probabilistic fusion yields better results. Our simple late-fusion approach achieves state-of-the-art performance, improving over prior work by 12.2% mAP.

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. 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.

Future Work. LT3D remains a challenging problem that requires further study by the community. Building end-to-end multi-modal models that leverages the design principles outlined in this paper may achieve better results. Further, leveraging temporal information to interpolate missed-detections and remove false positives in each modality can help improve late-fusion. Recent work in large-scale vision language models [31, 78] show promising zero-shot results in detecting rare classes. Identifying ways of incorporating foundation models into our late-fusion framework can greatly improve LT3D. Lastly, future should consider how LT3D impacts downstream forecasting and motion planning tasks.

Bibliography

[1] Shaden Alshammari, Yu-Xiong Wang, Deva Ramanan, and Shu Kong. Long-tailed recognition via weight balancing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2.3

[2] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *CVPR*, 2022. (document), 1, 2.2, 3, 3.3, 4.3, 4.1, 4.2, 4.5.1, 4.5

[3] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9287–9296, 2019. 3.4.1

[4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. (document), 1, 1.2, 3.4.1, 4.1, 4.2

[5] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, 2019. 2.3, 4.1

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, 2020. 2.1, 3.4.1

[7] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 1

[8] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. 2.3

[9] Yi-Ting Chen, Jinghao Shi, Zelin Ye, Christoph Mertz, Deva Ramanan, and Shu

Kong. Multimodal object detection via probabilistic ensembling. In *European Conference on Computer Vision (ECCV)*, 2022. 3.4.3, 3.4.3

[10] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, 2019. 2.3

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 4.1

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2.2

[13] Chris Drummond, Robert 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. 2.3

[14] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision (IJCV)*, 2015. 4.2

[15] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009. 3.4.1

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 1, 4.2

[17] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris

Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3, 000 hours of egocentric video. *Computer Vision and Pattern Recognition 2022*. 5

[18] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2485–2494, 2020. 2.1

[19] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017. 3.4.3

[20] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 2.3, 4.5.1

[21] Shubham Gupta, Jeet Kanjani, Mengtian Li, Francesco Ferroni, James Hays, Deva Ramanan, and Shu Kong. Far3det: Towards far-field 3d detection. In *NeurIPS*, 2022. 2.1

[22] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International Conference on Intelligent Computing*, pages 878–887. Springer, 2005. 2.3

[23] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Deep imbalanced learning for face recognition and attribute prediction. *PAMI*, 42(11):2781–2794, 2019. 2.3

[24] Junjie Huang and Guan Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*, 2022. 2.1

[25] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 2.1

[26] Yanqin Jiang, Li Zhang, Zhenwei Miao, Xiatian Zhu, Jin Gao, Weiming Hu, and Yu-Gang Jiang. Polarformer: Multi-camera 3d object detection with polar transformers. *arXiv preprint arXiv:2206.15398*, 2022. 2.1, 3, 4.3, 4.1

[27] Yang Jiao, Zequn Jie, Shaoxiang Chen, Jingjing Chen, Xiaolin Wei, Lin Ma, and Yu-Gang Jiang. Msmdfusion: Fusing lidar and camera at multiple scales with multi-depth seeds for 3d object detection. *arXiv preprint arXiv:2209.03102*, 2022. 2.2

[28] Salman Khan, Munawar Hayat, Syed Waqas Zamir, Jianbing Shen, and Ling Shao. Striking the right balance with uncertainty. In *CVPR*, 2019. 2.3

[29] Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A Sohel, and Roberto Togneri. Cost-sensitive learning of deep feature representations from

imbalanced data. *IEEE transactions on neural networks and learning systems*, 29(8):3573–3587, 2017. 2.3

[30] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2.1, 3, 4.3, 4.1, 4.4

[31] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *CVPR*, 2022. 3.4.1, 5

[32] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi 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. (document), 4.5.1, (b), (c), (d), 4.5.1, 4.6

[33] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*, pages 1–18. Springer, 2022. 2.1, 3, 4.3, 4.1

[34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014. 4.1

[35] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014. 3.4.1

[36] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014. 4.2

[37] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 3.4.1, 4.1

[38] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 2.1, 3.4.1

[39] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. *arXiv preprint arXiv:2205.13542*, 2022. 2.2, 3

[40] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, pages 2537–2546, 2019. 4.1

[41] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. (document), 1.1, 2.3, 4.2

[42] Su Pang, Daniel Morris, and Hayder Radha. Clocs: Camera-lidar object candidates fusion for 3d object detection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020. 2.2, 3.4.2

[43] Neehar Peri, Achal Dave, Deva Ramanan, and Shu Kong. Towards long-tailed 3d detection. In *Conference on Robot Learning (CoRL)*, 2022. (document), 1, 1.2, 1, 2.2, 3.2, 3.4.3, 4.1

[44] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, 2020. 2.1

[45] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2.2

[46] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *IEEE conference on computer vision and pattern recognition*, 2018. 2.2, 3.4.2

[47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2.2

[48] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 3.4.1

[49] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 3.4.1

[50] William J Reed. The pareto, zipf and other power laws. *Economics letters*, 74 (1):15–19, 2001. 2.3

[51] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural*

Information Processing Systems, 2015. 3.4.1

[52] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 2015. 4.5.2

[53] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 4.2

[54] Neil Savage. Robots rise to meet the challenge of caring for old people. *Nature*, 2022. 5

[55] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. *International Journal of Computer Vision*, pages 1–21, 2022. 2.1

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

[57] Araz Taeihagh and Hazel Si Min Lim. Governing autonomous vehicles: emerging responses for safety, liability, privacy, cybersecurity, and industry risks. *Transport Reviews*, 39(1):103–128, 2019. 1

[58] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *NeurIPS*, 2020. 2.3

[59] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019. 2.1

[60] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *CVPR*, 2018. 4.1

[61] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *IEEE/CVF conference on computer vision and pattern recognition*, 2020. 2.2, 3.3

[62] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7:

Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2022. 3, 3.4.1

[63] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. FCOS3D: fully convolutional one-stage monocular 3d object detection. In *ICCV*, 2021. (document), 2.1, 3, 3.4.1, 3.4.1, 3.3, 4.3, 4.2, 4.3

[64] Xudong Wang, Zhaowei Cai, Dashan Gao, and Nuno Vasconcelos. Towards universal object detection by domain attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7289–7298, 2019. 3.4.1

[65] Benjamin Wilson, Zsolt Kira, and James Hays. 3d for free: Crossmodal transfer learning using hd maps. *arXiv preprint arXiv:2008.10592*, 2020. 1

[66] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. 4.1

[67] Kelvin Wong, Shenlong Wang, Mengye Ren, Ming Liang, and Raquel Urtasun. Identifying unknown instances for autonomous driving. In *CoRL*, 2020. 1

[68] Cinna Julie Wu, Mark Tygert, and Yann LeCun. A hierarchical loss and its problems when classifying non-hierarchically. *PLoS ONE*, 14, 2019. (document), 4.5.1, 4.5.1, 4.6

[69] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *IEEE conference on computer vision and pattern recognition*, 2018. 2.2, 4.5.1

[70] Hang Xu, Linpu Fang, Xiaodan Liang, Wenxiong Kang, and Zhenguo Li. Universal-rcnn: Universal object detector via transferable graph r-cnn. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12492–12499, 2020. 3.4.1

[71] Zeyu Yang, Jiaqi Chen, Zhenwei Miao, Wei Li, Xiatian Zhu, and Li Zhang. Deepinteraction: 3d object detection via modality interaction. In *NeurIPS*, 2022. 3, 4.3, 4.1, 4.2

[72] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *CVPR*, 2021. (document), 1.2, 2.1, 3, 3.3, 4.3, 4.2, 4.3, 4.4, 4.6, 4.5.2, 4.8

[73] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Multimodal virtual point 3d detection. *NeurIPS*, 2021. 2.2

[74] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and

Harry Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In *International Conference on Learning Representations*, 2022. (document), 1.2, 3, 3.4.1

[75] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *CVPR*, 2021. 2.3

[76] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *arXiv:2110.04596*, 2021. 2.3

[77] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. *CoRR*, abs/2004.01177, 2020. 2.1

[78] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*, pages 350–368. Springer, 2022. 5

[79] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Simple multi-dataset detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7571–7580, 2022. 3.4.1

[80] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint arXiv:1908.09492*, 2019. 2.1, 2.3, 4.1, 4.5.2