

Semantically Consistent Alignment for Novel Object Discovery in Open-Vocabulary 3D Object Detection

by

Adrian Chow

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2025

© Adrian Chow 2025

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

3D object detection is a fundamental task in the autonomous driving perception pipeline, where identifying and localizing objects within the surrounding environment is critical for safe and robust decision-making. However, traditional 3D object detectors are limited by their reliance on a closed set of training categories, rendering them incapable of recognizing novel or out-of-distribution objects encountered in open-world driving scenarios. To address this limitation, the field of open-vocabulary (OV)-3D object detection has emerged, aiming to generalize beyond predefined label sets by leveraging vision-language models (VLMs) to align 3D object proposals with semantically rich 2D language-informed features.

Despite promising results, a major challenge in OV-3D object detection lies in achieving robust cross-modal alignment between 3D and 2D features, which is often compromised by noisy annotations, occlusions, and resolution inconsistencies that disrupt semantic coherence. In this thesis, we present OV-SCAN, a novel framework for Open-Vocabulary 3D object detection that enforces Semantically Consistent Alignment for Novel object discovery. OV-SCAN introduces a two-stage strategy: (1) discovering precise 3D annotations for novel objects using vision-language supervision, and (2) filtering out semantically inconsistent or low-quality 3D–2D training pairs that arise from annotation errors and sensor limitations.

We validate the effectiveness of OV-SCAN through comprehensive experiments on autonomous driving benchmarks, where our framework consistently outperforms existing methods in the OV-3D object detection task. Overall, OV-SCAN underscores the critical role of semantic consistency in cross-modal alignment and demonstrates its potential as a scalable solution for discovering and localizing novel objects in real-world autonomous driving scenarios.

Acknowledgements

I would like to thank my advisor, Prof. Krzysztof Czarnecki, for fostering an environment where curiosity and independence were not only welcomed but actively encouraged. Without his trust in allowing me to chart my own research path, I would not have developed the same depth of academic growth or confidence in myself as a researcher. I would also like to thank Dr. Michal Antkiewicz for taking a chance on me in the summer of 2021 and helping lay the foundation for my research journey. The time we spent troubleshooting the Autonomoose affirmed my passion for this field.

I would also like to thank all the members of the WISE lab for their camaraderie, insightful discussions, and unwavering support throughout my graduate studies. In particular, I'm grateful to Chengjie Huang and Yimu Wang for consistently offering their sharp technical insight and thoughtful advice. A special thank you to James and Evelien Riddell for sharing this master's journey with me—it was a privilege to go through it together.

To my family - Mom, Dad, and Cassie—thank you for always being there, offering your support and company through every step. Mom, as a first-generation immigrant and single mother, you gave us everything, and I'm forever grateful for the strength, sacrifice, and love you gave so selflessly. To my deepest friends — you know who you are —thank you for your friendship, laughter, and for always being there when I needed a break or a boost. Lastly, to my life partner Karen—thank you for your encouragement, humor, and unwavering support. Your belief gives me courage, your love inspires me, and this journey has been brighter and more meaningful because of you.

Dedication

To Karen

Table of Contents

Author’s Declaration	ii
Abstract	iii
Acknowledgements	iv
Dedication	v
List of Figures	ix
List of Tables	xi
1 Introduction	1
2 Background	5
2.1 LiDAR-based 3D Object Detection	5
2.1.1 Traditional 3D Object Objection	5
2.1.2 Point Cloud Feature Representation	6
2.1.3 Object Feature Proposal Networks	7
2.1.4 Open-Vocabulary 3D Object Objection	7
2.2 Vision-Language Models	8
2.2.1 CLIP	8
2.2.2 GLIP	9
2.2.3 Grounding DINO	9

3	Related Work	10
3.1	Related Work	10
3.1.1	Traditional 3D Object Detection	10
3.1.2	Open-Vocabulary 2D Object Detection	11
3.1.3	Open-Vocabulary 3D Object Detection	11
4	Method	12
4.1	Notation and Preliminaries	12
4.1.1	3D Object Detection	12
4.1.2	Open-Vocabulary 3D Object Detection	13
4.2	Semantically Consistent NOD (SC-NOD)	14
4.2.1	Novel Object Proposals	14
4.2.2	Cross-Modal Association	14
4.2.3	Adaptive 3D Box Search	15
4.3	Model Architecture	19
4.3.1	Hierarchical Two-Stage Alignment (H2SA) Head	19
4.3.2	Multi-Sensor Fusion	20
4.4	Training	21
4.4.1	Training Loss	21
4.4.2	Selective Alignment	22
4.5	Prompt-based Classification	22
5	Experiments	24
5.1	Datasets	24
5.2	Implementation Details	25
5.2.1	Novel Object Discovery	25
5.2.2	Adaptive 3D Box Search	26
5.2.3	Training	26

5.2.4	Prompt-based Classification	27
5.2.5	Selective Alignment	28
5.3	Evaluation Metrics	29
5.4	Results	29
5.4.1	Performance on NuScenes	29
5.4.2	Performance on KITTI	30
5.4.3	Novel Object Discovery	31
5.4.4	Selective Alignment	32
5.5	Ablation Studies	33
5.5.1	Adaptive 3D Box Search Ablation Study	33
5.5.2	Selective Alignment Ablation Study	34
5.5.3	Alignment Head Ablation Study	35
5.6	Qualitative Results	35
6	Conclusion	38
6.1	Conclusion	38
6.2	Limitations and Future Work	38
	References	40
	APPENDICES	50
A	TEST	51
A.1	Pseudocode for Cross-Modal Association	51
A.2	Pseudocode for Cross-Modal Target Preparation	51

List of Figures

1.1	3D Annotation Errors. Common 3D annotation errors during box parametrization, including but not limited to, poor L-shape fitting, misinterpreted surfaces, and misaligned surfaces. Ground truth is in green and potential mistakes are in red.	2
1.2	Sources of Semantic Discrepancies. (a) CLIP similarity scores for a truck reveal that occlusion cases result in an ambiguous 2D image feature. (b) CLIP similarity scores for a distant pedestrian demonstrate that insufficient resolution leads to degraded 2D image feature.	3
4.1	Overall Framework for OV-SCAN. During novel object discovery, SC-NOD associates novel object proposals with corresponding object clusters, creating cross-modal proposals. SC-NOD performs an adaptive search to fit 3D annotations and extracts 2D image features to prepare cross-modal targets for supervision. SC-NOD identifies which samples are fit for alignment based on cross-modal semantic consistency. During training, all 3D annotations are used, while only consistent novel objects guide alignment.	13
4.2	Illustration of the Novel Object Proposals. Given a list of novel classes, SC-NOD leverages Grounding DINO [30] to generate 2D bounding boxes and uses SAM [25] to produce corresponding instance masks.	14
4.3	Illustration of Cross-Modal Association and Adaptive 3D Box Search. Starting with a 2D novel object proposal (1), a 3D frustum is projected and points are extracted (2). Object clusters within the frustum are identified and paired with the proposal (3). Random 3D boxes are initialized (4), optimized (5), and filtered to retain the box closest to the ego vehicle (6).	18

4.4	Illustration of the Hierarchical Two-Stage Alignment (H2SA) Head. H2SA first predicts the high-level novel classes, then derives class-based text prompts. H2SA then uses text prototypes to incrementally map 3D features to their 2D counterparts.	20
5.1	Percentage of instance pixels across object classes. Vehicles, such as cars, typically occupy a greater proportion of pixels within their 2D proposals compared to objects like pedestrians and traffic cones.	28
5.2	Distribution of training labels.	32
5.3	Cross-modal Alignment Performance. The red CDF shows the distribution of the distance between the 3D embedding produced by a baseline OV-3D detector and the corresponding 2D embedding from CLIP on the nuScenes validation set. The green CDF shows this distribution using OV-SCAN instead of the baseline. The latter is shifted to the left, showing an improved alignment between the 3D and 2D embeddings due to supervision by our higher quality 3D-2D proposal pairings and our alignment head. . . .	33
5.4	Visualization results using open-vocabulary on a variety of traffic conditions. OV-SCAN performs inference on a set of urban scenes identifying a diverse set of objects. To demonstrate OV capabilities, objects are classified into more fine-grained classes as opposed to traditional closed-set classes.	36
5.5	Comparison between OV-SCAN and Find n’ Propagate [15]. OV-SCAN regresses more precise bounding boxes than Find n’ Propagate without requiring any human-annotated labels. We compare to Find n’ Propagate in “Setting 2” which uses 3 base classes (<i>car, pedestrian, and bicycle</i>) leaving the rest as novel.	37

List of Tables

5.1	Novel classes used to identify novel object proposals for each dataset. These classes are referred to as “novel” since their ground truth labels, although available in the respective datasets, are not used in training OV-SCAN.	25
5.2	PSO hyperparameters used in adaptive 3D box search.	26
5.3	Fine-grained subclasses for each high-level novel class in the nuScenes dataset.	27
5.4	Parameterization of τ_{occ} for different novel classes.	28
5.5	Results on nuScenes. We report the overall mAP, NDS, and individual class APs. All classes are novel (<i>i.e.</i> , no human-annotations are used to train except in CoDA). CoDA uses “car” as a base-class. Best in bold and “*” denotes omitted methods for fair comparison.	30
5.6	KITTI AP_{3D@50} results for OV-SCAN. All classes are treated as novel.	30
5.7	Results on KITTI. We report the overall AP _{3D@50} for each class at medium difficulty. Best in bold .	31
5.8	Ablations Study on Adaptive 3D Box Search. We evaluate performance on nuScenes using different box search methods and varying numbers of search iterations per proposal.	34
5.9	Ablation Study on 3D Box Search Cost Weights. Selected setting is marked in gray.	34
5.10	Ablation Study on Selective Alignment Filters. Selected settings are marked in gray.	35
5.11	Ablation Study on Alignment Head Design. Selected settings are marked in gray.	35

Chapter 1

Introduction

Recent advances in autonomous driving have intensified research in 3D object detection, with most methods operating in a closed-set setting, classifying objects into a small set of predefined categories (e.g. vehicle, cyclist, pedestrian). However, such high-level object identification is inadequate for navigating complex real-world environments. Autonomous systems need to recognize objects at a deeper semantic level. For instance, distinguishing between a fire truck and a school bus, as opposed to simply labeling them as trucks or buses, is critical for safe and context-aware decision-making.

To address the need for deeper semantic understanding in 3D vision systems, open-vocabulary (OV)-3D object detection has emerged. In OV-2D object detection, OV capabilities rely on large-scale datasets of image-text pairs, which are often unavailable in 3D [29, 28, 39, 11, 60]. Without similar large-scale 3D-text datasets, methods must find alternative ways to provide accurate 3D annotations for novel objects and achieve alignment between 3D and text.

OV-3D object detection faces two main challenges: (1) novel object discovery (NOD), which involves generating 3D labels for novel objects in order to train an off-the-shelf LiDAR-based detector, and (2) semantic alignment between 3D features and textual embeddings. Existing NOD approaches generally fall into two categories, online and offline methods. The former starts with a set of base objects and progressively expand the training labels to include novel objects during training [5, 6]. The latter generates pseudo-labels for novel objects ahead of time [34, 15, 66]. Online methods often struggle to balance training with discovery, leading to biases in object shape and scale that result in overfitting to base categories. Due to the aforementioned challenges with online methods, recent work has increasingly shifted toward offline approaches. In these offline pipelines, OV-2D detectors

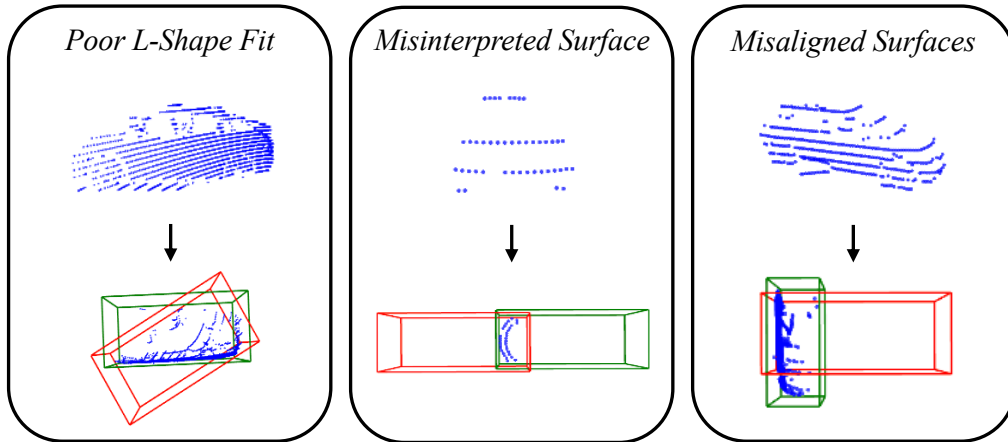


Figure 1.1: **3D Annotation Errors.** Common 3D annotation errors during box parametrization, including but not limited to, poor L-shape fitting, misinterpreted surfaces, and misaligned surfaces. Ground truth is in green and potential mistakes are in red.

such as Grounding DINO [29] and OWL-ViT [38] first generate 2D proposals from multi-view images, which are typically paired with point cloud. These 2D proposals are then projected into 3D space and reformulated as 3D pseudo-labels for training. However, because LiDAR point clouds are inherently sparse, naïve projection methods such as simple clustering struggle to accurately fit boxes to objects that are only partially observable and not visibly dense. Most recently, Find n’ Propagate [15] employs an exhaustive greedy search to optimally choose 3D bounding box parameters for each 2D proposal, yet this approach still suffers from several 3D annotation errors shown in Figure 1.1. Such noise hinders the 3D object detector’s ability to extract discriminative 3D features.

To achieve semantic alignment between 3D data and text, existing methods use the 2D image space as an implicit intermediary [66, 34, 5, 65]. Each 3D bounding box label is associated with a corresponding 2D image region from which a visual embedding is extracted using a VLM such as CLIP [47]. During training, a base 3D object detector is supervised using the 3D annotations for box regression, while simultaneously aligning 3D object features with target 2D image embeddings. During test time, novel objects are classified via prompt-based classification, assigning labels based on the highest similarity between 3D features and text embeddings. Achieving strong alignment requires both semantically distinct 3D and 2D features. However, existing methods often overlook common autonomous driving scenarios where objects are partially occluded (Figure 1.2a) or appear small at longer distances (Figure 1.2b). In such cases, the 2D features become ambiguous

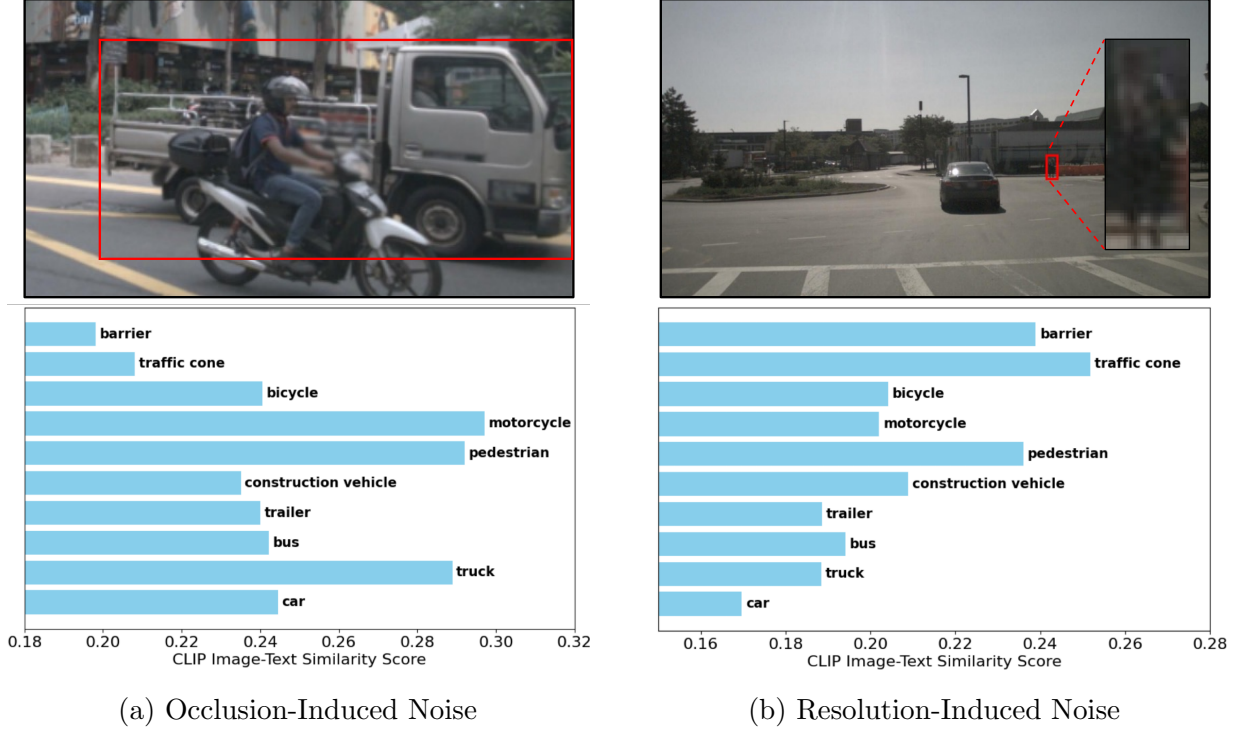


Figure 1.2: **Sources of Semantic Discrepancies.** (a) CLIP similarity scores for a truck reveal that occlusion cases result in an ambiguous 2D image feature. (b) CLIP similarity scores for a distant pedestrian demonstrate that insufficient resolution leads to degraded 2D image feature.

or lack sufficient representation, leading to confusion during cross-modal alignment.

In this work, we introduce OV-SCAN, a OV-3D object detector which targets strong OV detection performance from enhanced semantic alignment between 3D objects and 2D image features. More specifically, we introduce the Semantically-Consistent Novel-Object Discovery (SC-NOD) module to handle the inherit challenges of noisy cross-modal alignment. To address *3D annotation noise*, SC-NOD reformulates 3D box search as a non-linear optimization problem and employs an adaptive search strategy to efficiently optimize 3D bounding box parameters from 2D proposals. To mitigate *occlusion-induced noise* and *resolution-induced noise*, SC-NOD enables a selective alignment mechanism by identifying the alignment pairs that are semantically consistent. During training, all 3D annotations contribute to box-related losses in order to maintain high recall, while only semantically consistent alignment pairs guide the cross-modal alignment. We further propose the Hierarchical Two-Stage Alignment (H2SA) head to enhance cross-modal alignment. Novel

classes are organized hierarchically by first grouping them into broad, high-level categories (e.g., car) and then subdividing these categories into more detailed subclasses (e.g., SUV, sedan). This hierarchical structure allows H2SA to initially perform coarse-grained classification, before carefully aligning class-informed object features for fine-grained discrimination among closely related subclasses. We summarize our main contributions as followed:

- We present OV-SCAN, an OV-3D object detector benefiting from improved cross-modal alignment.
- We propose the SC-NOD module, extending existing 3D box search methods and introducing selective cross-modal alignment during training. In addition, the proposed H2SA head effectively aligns 3D-to-2D alignment pairs by introducing a two-stage alignment process.
- We validate OV-SCAN on the nuScenes [2] and KITTI [17] datasets, demonstrating that OV-SCAN achieves state-of-the-art performance in an outdoor OV-3D object detection setting without supervising with any human-annotated data.

Chapter 2

Background

This chapter reviews LiDAR-based 3D object detection, from traditional pipelines and point cloud representations to object proposal mechanisms. It introduced open-vocabulary 3D detection as a means to overcome closed-set limitations, leveraging vision-language models like CLIP [47], GLIP [28], and Grounding DINO [30] for text-driven semantic generalization. These foundations support the development of open-world 3D perception explored in the following chapters.

2.1 LiDAR-based 3D Object Detection

2.1.1 Traditional 3D Object Detection

The goal of LiDAR-based 3D object detection is to identify, classify, and accurately localize all relevant objects in a scene using raw point clouds captured by one or more LiDAR sensors. The output typically includes a set of oriented 3D bounding boxes—defined by each object’s position, dimensions, and heading angle—along with semantic class labels, confidence scores, and in some cases, estimated kinematic properties such as lateral and longitudinal velocity. Accurate 3D object detection underpins downstream perception tasks—tracking, motion prediction, and planning—and is therefore central to safe autonomous driving.

Given a point cloud $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N\} \subset \mathbb{R}^C$, where each point $\mathbf{p}_i \in \mathbb{R}^C$ typically includes features such as $(x, y, z, \text{intensity})$ and the points are unordered, the objective of

LiDAR-based 3D object detection is to learn a function

$$f : \mathbb{R}^{N \times C} \rightarrow \mathcal{O},$$

where $\mathcal{O} \in \mathbb{R}^{M \times D}$ is a set of M detected objects, each represented as a vector of $D = 10$ parameters:

$$o_j = [x, y, z, l, w, h, r_y, v_x, v_y, s],$$

with (x, y, z) denoting the 3D center of the bounding box, (l, w, h) its dimensions (length, width, height), r_y the yaw angle (orientation), (v_x, v_y) the object’s longitudinal and lateral velocities in the ground plane, and $s \in [0, 1]$ the confidence score associated with the detection.

A common training objective for such a function is a multi-task loss that jointly optimizes bounding box regression, velocity estimation, and classification, for example by combining a smooth L1 (Huber) loss for the continuous parameters with a cross-entropy or focal loss for the class score.

2.1.2 Point Cloud Feature Representation

Learning the detection function f , which maps a raw point cloud to a structured set of objects, begins with extracting meaningful feature representations. Since point clouds are sparse, unordered, and variable in size, effective feature encoding is crucial and non-trivial. The quality of these features directly affects downstream object detection performance.

Processing point clouds poses unique challenges not present in images. Their lack of a regular grid rules out standard convolutional architectures, prompting the development of specialized methods. PointNet [44] and PointNet++ [45] were early approaches that apply shared MLPs to individual points and aggregate global features via symmetric pooling. While efficient and permutation-invariant, they fail to capture local geometric context essential for detailed reasoning. DGCNN [56] improves on this by dynamically building a graph of local neighborhoods using k-nearest neighbors and applying edge convolutions to model point relationships. This captures fine-grained structure but introduces high computational overhead. Point Transformer [67] further enhances expressiveness through attention mechanisms that model local and global dependencies, but its self-attention scales poorly with point count, making it impractical for real-time use.

While these methods achieve high accuracy, their computational cost limits their suitability for real-time 3D detection, especially in high-resolution or large-scale environments. In contrast, voxelization-based representations offer a practical trade-off between accuracy

and efficiency. By discretizing 3D space into a voxel grid, point clouds are converted into sparse tensors suitable for convolutional processing. This enables the use of efficient 3D or sparse convolutions, as demonstrated by models like VoxelNet[69] and SECOND [58], supporting fast inference while retaining sufficient geometric detail for reliable object detection.

2.1.3 Object Feature Proposal Networks

Following the extraction of point-level features, the next stage in the 3D detection pipeline involves aggregating these features into compact object-level representations. This is typically achieved through object feature proposal networks, which process the encoded point cloud and generate candidate object regions—referred to as proposals—each associated with a learned feature vector that summarizes its geometry and semantics.

Object proposal features, denoted by $\mathbf{f}_{\text{obj}}$, are regressed from the dense or sparse feature maps derived from voxelized or point-based encoders. Proposal generation mechanisms vary by architecture but commonly involve either anchor-based schemes (e.g., in SECOND [58]) or anchor-free formulations (e.g., in CenterPoint [62]). In both cases, the network identifies candidate object centers and extracts region-specific features using methods such as region pooling, attention, or learned sampling.

Each object proposal feature encapsulates not only the predicted 3D bounding box parameters—position (x, y, z) , size (l, w, h) , and orientation (r_y) —but also a high-dimensional object feature vector. This vector encodes the local geometry and contextual information around the object and may be used for downstream tasks such as classification, velocity estimation, or multimodal fusion. In some architectures, the object feature also includes motion cues like lateral and longitudinal velocity components (v_x, v_y) , and a confidence score s indicating the likelihood that the proposal corresponds to a valid object.

2.1.4 Open-Vocabulary 3D Object Objectection

In traditional 3D object detection, classification is performed under a *closed-set* assumption, where the model is trained to recognize objects from a fixed set of predefined categories (e.g., car, pedestrian, cyclist). Each object proposal o_j , after being regressed from the point cloud features, is passed through a classification head—typically a fully connected layer followed by a softmax activation—which outputs a probability distribution over the known classes:

$$\mathbf{p}_j = \text{softmax}(W \cdot \mathbf{f}_{\text{obj}} + b),$$

where $\mathbf{f}_j$ is the object feature embedding for the proposal, and W, b are the parameters of the classification layer. The network is trained using a cross-entropy loss between the predicted distribution $\mathbf{p}_j$ and the ground-truth label. At inference time, the proposal is assigned the class with the highest probability. While effective in constrained domains, this approach fails to generalize to novel or unseen categories, as the model is limited to the label space defined during training.

To address this limitation, OV-3D object detection removes the reliance on a fixed set of categories by incorporating semantic information from pretrained language models. Rather than classifying proposals using a static classifier, the object feature $\mathbf{f}_j$ is projected into a shared embedding space, where it is compared against a set of language-derived class embeddings (e.g., CLIP [47] text embeddings). Classification becomes a similarity-based retrieval task:

$$\text{score}(\mathbf{f}_{\text{obj}}, t_k) = \frac{\phi(\mathbf{f}_{\text{obj}}) \cdot t_k}{\|\phi(\mathbf{f}_{\text{obj}})\| \|t_k\|},$$

where t_k is the text embedding for category k , and $\phi(\cdot)$ is a learned projection function. This formulation allows the detector to recognize and assign labels to objects described by arbitrary text inputs—even those never seen during training.

For OV-3D object detection to be effective, object features must be semantically expressive, capturing not only geometric properties but also contextual and appearance cues that enable alignment with diverse category descriptions. By enabling recognition beyond a fixed label set, OV-3D object detection offers a flexible and scalable alternative to traditional methods.

2.2 Vision-Language Models

2.2.1 CLIP

CLIP (Contrastive Language-Image Pretraining) [47] is a foundational vision-language model that learns to associate images and text by projecting them into a shared embedding space through contrastive learning on large-scale internet data. Given a batch of image-text pairs $\{(I_i, T_i)\}_{i=1}^N$, CLIP encodes each image I_i using a vision encoder to obtain an image embedding v_i , and each text T_i using a text encoder to obtain a text embedding t_i . These embeddings are normalized and compared using cosine similarity:

$$s_{ij} = \frac{v_i \cdot t_j}{\|v_i\| \|t_j\|}$$

CLIP is trained using a symmetric cross-entropy loss over the similarity matrix $S = [s_{ij}]$, encouraging high similarity when $i = j$ (i.e., the image and text match) and low similarity otherwise. This contrastive objective aligns paired image and text representations while separating unrelated ones, enabling CLIP to generalize to novel concepts without explicit supervision. By leveraging natural image-text pairings from the web, CLIP learns broad semantic associations and demonstrates strong zero-shot and open-vocabulary classification capabilities. These properties have made it a key component in multimodal learning, with recent applications extending into areas such as 3D object detection for autonomous driving.

2.2.2 GLIP

GLIP (Grounded Language-Image Pretraining) [28] builds on CLIP [47] by extending contrastive image-text alignment to the level of object regions and textual spans. Instead of learning global correspondences, GLIP aligns region proposals from a visual detector with natural language phrases using a region-text contrastive loss. A pretrained language encoder processes the input query, while a vision backbone extracts dense region features. Region and phrase features interact through attention, with contrastive learning driving semantic alignment, while detection losses enforce accurate localization. Trained on large-scale phrase grounding datasets, GLIP enables open-vocabulary object detection by localizing free-form language descriptions, making it a key advancement in unified vision-language modeling.

2.2.3 Grounding DINO

Grounding DINO [30] extends the capabilities of GLIP [28] by integrating open-vocabulary grounding into a high-performance transformer-based object detector. Built upon the DINO [8] framework, which leverages DETR-style [7] object queries and dense transformer decoding for end-to-end detection, Grounding DINO introduces cross-modal attention between image features and language tokens to enable text-conditioned object localization. A natural language query directly guides the object queries during decoding, allowing the model to focus on regions relevant to the textual description. This design enables accurate zero-shot detection and grounding of arbitrary concepts without requiring class-specific supervision.

Chapter 3

Related Work

3.1 Related Work

In this chapter, we survey the relevant related work spanning traditional 3D object detection, recent advances in open-vocabulary 2D object detection, and the emerging field of open-vocabulary 3D object detection.

3.1.1 Traditional 3D Object Detection

Traditional 3D object detection methods focus primarily on closed-set scenarios, training models on a limited number of common classes. LiDAR-based approaches are categorized into single-representation, hybrid-representation, and transformer-based methods. Single-representation methods such as VoxelNet [69] discretize point clouds into dense voxel tensors processed with 3D CNNs, achieving fine geometric reasoning at the cost of high memory and compute. Efficiency has since improved through sparse convolutions [58, 10], pillar [26], and BEV representations [62] for efficiency. Point-based methods like PointNet [44] and PointRCNN [50] directly process raw point clouds, achieving detailed geometry at the cost of speed. Hybrid models like PV-RCNN [48, 49] first build a sparse voxel backbone for global context, then aggregate high-resolution point features for local refinement, striking strong detection performance at the cost of efficiency. Transformer-based architectures [52, 43, 41, 33, 37] replace convolutional blocks with self-attention, capturing long-range spatial dependencies and improving detection of distant objects. LiDAR-camera fusion approaches [1, 32, 61] combine LiDAR’s spatial precision with RGB images to enhance semantic understanding.

3.1.2 Open-Vocabulary 2D Object Detection

The emergence of large-scale pretrained models with strong text-vision alignment and impressive zero-shot transfer capabilities has driven significant interest in tackling the OV problem in 2D object detection. Approaches like OVR-CNN [64], ViLD [19], OWL-ViT [38], and OV-DETR [63] address this challenge through knowledge distillation, using pretrained VLMs such as CLIP [47]. These methods integrate semantic knowledge into traditional detectors by contrastively aligning region features or object embeddings. Subsequently, GLIP [28] reformulates object detection as phrase grounding, exploiting large-scale image-text pair pretraining to unify localization and classification while enabling generalization to unseen categories through semantic understanding. Grounding DINO [29] advances this idea further by applying tightly coupled fusion strategies on top of DINO [9], ensuring effective integration of language and vision. Extending OV-2D object detection to real-time applications, YOLO-World [11] uses vision-language modelling within the efficient YOLO architecture [23].

3.1.3 Open-Vocabulary 3D Object Detection

Building on the success of OV-2D object detectors, recent efforts have aimed to extend these capabilities to 3D. A key challenge in this adaptation is the scarcity of diverse labeled data required for large-scale 3D training. OV-3Det [34] introduced the 2D-to-3D paradigm in an indoor setting, where VLMs generate annotations for a diverse set of objects, which are then lifted to 3D using simple point cloud clustering. CoDA [5] pretrains a base detector with a small set of human-annotated labels and introduces a framework to incorporate novel labels during training. FM-OV3D [65] further improves detection performance by exploiting the strengths of multiple foundational models. OpenSight [66] extends open-vocabulary detection into the outdoor domain, incorporating temporal and spatial awareness in box annotations. In addition, their method also introduces 2D-to-3D geometric priors to guide annotation under sparse point cloud conditions. Find n’ Propagate [15] addresses the issue of insufficient recall of novel objects and propagates detection capabilities to more distant areas. ImOV3D [59] demonstrates an effective method to train a OV-3D object detector without any 3D point clouds, instead using a pseudo-multimodal representation. Despite advancements, existing methods often neglect the impact of existing noise sources that weaken cross-modal alignment.

Chapter 4

Method

In this section, we detail OV-SCAN. An overview is provided in Figure 4.1. SC-NOD (Sec. 4.2) generates 3D annotations from 2D proposals to form cross-modal alignment pairs. SC-NOD then determines the subset of semantically consistent pairs to guide cross-modal alignment. OV-SCAN comprises an off-the shelf 3D object detector with a specialized H2SA alignment head (Sec. 4.3), trained using the novel objects discovered by SC-NOD (Sec. 4.4).

4.1 Notation and Preliminaries

4.1.1 3D Object Detection

In traditional LiDAR-based 3D object detection, the objective is to train a detector using input-target pairs $\mathbf{D} = \{\mathcal{P}, \Omega\}$. Given a point cloud $\mathcal{P} = \{p_i\}_{i=1}^M$ with M points, where each point $p_i = (x_i, y_i, z_i)$ represents spatial coordinates, the detector predicts targets $\Omega = \{(B_i, c_i)\}_{i=1}^N$ for N objects. Each target consists of a 3D bounding box $B_i = (x_i, y_i, z_i, l_i, w_i, h_i, (r_y)_i)$ with center (x_i, y_i, z_i) , dimensions (l_i, w_i, h_i) , orientation $(r_y)_i$, and a class label c_i . We assume that each LiDAR frame is accompanied by a set of K images from different perspectives (multi-view), represented as $\mathcal{I} = \{I_j\}_{j=1}^K$. LiDAR points can be mapped to image space by using the extrinsic and intrinsic calibrations T_{ext} and T_{int} .

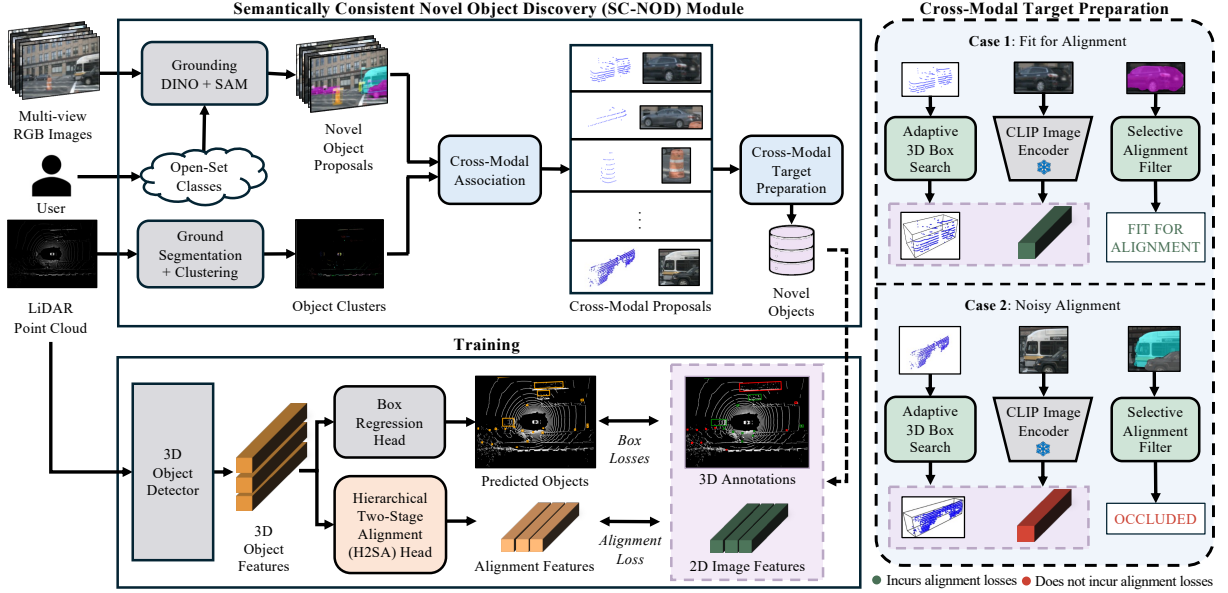


Figure 4.1: **Overall Framework for OV-SCAN.** During novel object discovery, SC-NOD associates novel object proposals with corresponding object clusters, creating cross-modal proposals. SC-NOD performs an adaptive search to fit 3D annotations and extracts 2D image features to prepare cross-modal targets for supervision. SC-NOD identifies which samples are fit for alignment based on cross-modal semantic consistency. During training, all 3D annotations are used, while only consistent novel objects guide alignment.

4.1.2 Open-Vocabulary 3D Object Detection

In an OV setting, 3D object detectors do not have access any target labels Ω and instead rely on VLMs to generate annotations for novel objects, creating a *novel object bank*. Our method extends the traditional target pair of 3D bounding box and class label, into a triplet target denoted by $\Omega' = \{(B_i, c_i, \mathcal{A}_{2D,i})\}_{i=1}^N$. $\mathcal{A}_{2D,i} \in \mathbb{R}^D$ represents a 2D alignment embedding that captures the semantic correspondence between the 3D object and its 2D image projection. Traditional LiDAR-based 3D object detection methods are designed to regress 3D object features $\mathcal{O}_{3D} \in \mathbb{R}^H$ given the input point-cloud $\mathcal{P}$. Our method aims to learn a function f that maps a set of 3D object features $\mathcal{O}_{3D}$ to both the set of predicted 3D bounding boxes B and predicted alignment vectors $\mathcal{A}_{2D}$, thereby bridging the 3D and 2D domains in OV-3D object detection. These alignment features are then used for prompt-based classification by comparing them with text embeddings generated from class prompts, enabling fine-grained recognition of novel objects.

4.2 Semantically Consistent NOD (SC-NOD)

4.2.1 Novel Object Proposals

A key component of open-vocabulary detection is identifying novel objects from an arbitrary set of classes. Following prior approaches [65, 66, 15], we employ Grounding DINO [29] to detect proposals from a given list of novel classes. For each LiDAR frame, novel object proposals $\mathbf{P}_{2D}$ are generated on K multi-view images, capturing objects from multiple perspectives. These proposals, defined as 2D bounding boxes, are represented for each image I_k as $\mathbf{P}_{2D,k} = \{b_{k,j}\}_{j=1}^{N_k}$, where $b_{k,j}$ specifies position and size. The complete set for a given frame is $\mathbf{P}_{2D} = \bigcup_{k=1}^K \mathbf{P}_{2D,k}$. SAM [25] then pairs instance masks $\mathbf{M} = \{m_{k,j}\}_{k=1,j=1}^{K,N_k}$ with each proposal, where $m_{k,j} \in \{0, 1\}^{H \times W}$ denotes a binary mask for each $b_{k,j}$, with $m_{k,j}(x, y) = 1$ indicating instance pixels.

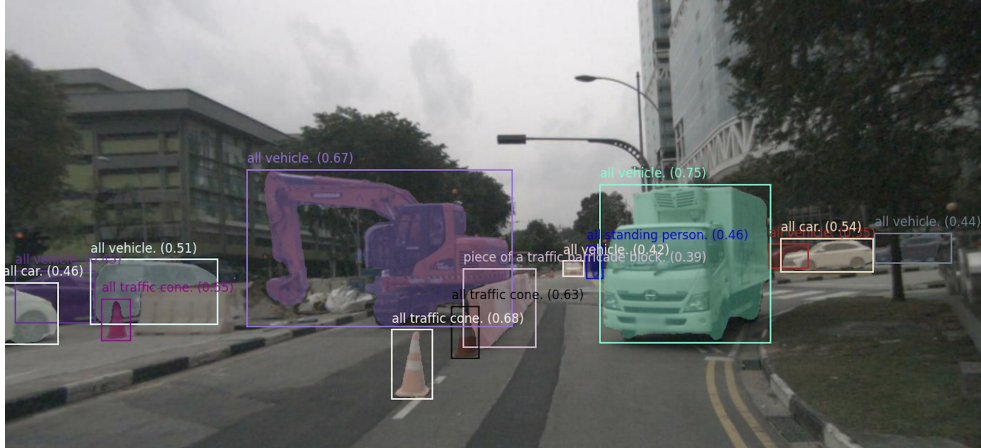


Figure 4.2: **Illustration of the Novel Object Proposals.** Given a list of novel classes, SC-NOD leverages Grounding DINO [30] to generate 2D bounding boxes and uses SAM [25] to produce corresponding instance masks.

4.2.2 Cross-Modal Association

During cross-modal association, novel object proposals are paired with object clusters to form cross-modal proposals. To generate object clusters, we first deploy Patchwork++ [27] to segment the scene and remove ground points, isolating potential object regions. The remaining points are then clustered using HDBScan [4], which groups spatially coherent

regions into individual object clusters. Each 2D proposal is back-projected into a 3D frustum defined by the camera parameters and a depth range ($d_{\min}, d_{\max}$). Within this frustum, candidate object clusters are identified by evaluating their spatial proximity to the center frustum ray. Following a *one-to-many* matching strategy, each 2D proposal forms pairs with any cluster whose minimum point-to-ray distance falls below a matching threshold τ_{match} , resulting in an initial set of cross-modal proposals.

In the *one-to-many* strategy, we allow a single 2D proposal to match with multiple object clusters. This is necessary to handle ambiguities introduced by partial occlusion, point cloud sparsity, or clustering errors—conditions that often lead to fragmented or incomplete object clusters. While multiple object clusters may represent parts of the same ground-truth object, it is often too ambiguous to consolidate clusters and determine which cluster can produce the best labels downstream. Instead, we treat each cross-modal proposal independently and potentially label the same object multiple times. These conflicts can simply be solved downstream using non-maximal suppression (NMS).

Additionally, we address *many-to-one* scenarios where large objects—such as transit buses or trucks—may span multiple camera views, resulting in 2D proposals from the ground truth object being associated with the same 3D cluster. Similarly, this case can be solved with the same approach to the *one-to-many* case using NMS.

4.2.3 Adaptive 3D Box Search

For each cross-modal proposal, an amodal 3D bounding box is generated to accurately capture the object’s position, dimensions, and orientation in 3D. To mitigate the impact of 3D annotation noise, it is crucial to explore the parameter space effectively and identify the optimal configuration for novel objects. Find n’ Propagate [15] employs an exhaustive greedy search over a frustum-defined area to estimate box parameters for novel object proposals. However, this approach inefficiently distributes a large number of search candidates to suboptimal configurations due to its expansive search space and limited precision. Our method addresses this inefficiency by reformulating the task as a continuous nonlinear optimization problem within a localized search area. Additionally, while their cost function considers only point density and multi-view alignment, our method extends it to incorporate surface alignment, further mitigating annotation errors.

Given a cross-modal proposal, comprising a 2D bounding box b_{img} and a set of object points $\mathcal{P}_{\text{obj}}$, the objective is to determine the 3D bounding box parameters $\theta = (x, y, z, l, w, h, r_y)$ that best represent the object. Following similar works [15, 66, 21], the search is guided by constraining the bounding box dimensions to a predefined set of

box anchors $\mathbf{A} = \{(\mathbf{A}_{\min}^i, \mathbf{A}_{\max}^i)\}_{i=1}^C$, corresponding to each novel class. Our method relies on CLIP [47] to classify the object into its corresponding novel class c . To reduce ambiguity in orientation, the yaw angle r_y is constrained to the range $[0, \pi]$. The continuous nonlinear optimization problem is then formulated in standard form:

$$\begin{aligned} \min_{\theta} \mathcal{J}(\theta, \mathcal{P}_{\text{obj}}, \mathbf{e}, b_{\text{img}}) &= \mathcal{J}_{3\text{D}}(\theta, \mathcal{P}_{\text{obj}}, \mathbf{e}) + \mathcal{J}_{2\text{D}}(\theta, b_{\text{img}}) \\ \text{subject to } \mathbf{A}_{\min} &\leq (l, w, h) \leq \mathbf{A}_{\max} \\ 0 &\leq r_y \leq \pi, \end{aligned} \quad (4.1)$$

where $\mathcal{J}_{3\text{D}}$ and $\mathcal{J}_{2\text{D}}$ are the cost functions detailed below, b_{img} is the 2D bounding box coordinates, $\mathcal{P}_{\text{obj}}$ represents object points in 3D space, and $\mathbf{e}$ is the ego’s position in 3D space.

To evaluate the fit of the box with respect to the object points, $\mathcal{J}_{3\text{D}}$ is defined as:

$$\begin{aligned} \mathcal{J}_{3\text{D}}(\theta, \mathcal{P}_{\text{obj}}, \mathbf{e}) &= \lambda_1 \mathcal{J}_{\text{density}}(\theta, \mathcal{P}_{\text{obj}}) \\ &\quad + \lambda_2 \mathcal{J}_{\ell\text{-shape}}(\theta, \mathcal{P}_{\text{obj}}) \\ &\quad + \lambda_3 \mathcal{J}_{\text{surface}}(\theta, \mathbf{e}). \end{aligned} \quad (4.2)$$

The point density term $\mathcal{J}_{\text{density}}$ quantifies the proportion of the object’s points that are enclosed within the 3D box. This encourages the parameters to adjust so that all points are captured by the box. $\mathcal{J}_{\text{density}}$ is defined as:

$$\mathcal{J}_{\text{density}}(\theta, \mathcal{P}_{\text{obj}}) = -\frac{\mathcal{G}_{\text{inside}}(\theta, \mathcal{P}_{\text{obj}})}{|\mathcal{P}_{\text{obj}}|}, \quad (4.3)$$

where

$$\mathcal{G}_{\text{inside}}(\theta, \mathcal{P}_{\text{obj}}) = \sum_{p_i \in \mathcal{P}_{\text{obj}}} \mathbf{1}(p_i \in B_{3\text{D}}(\theta)). \quad (4.4)$$

The ℓ -shape fitting term $\mathcal{J}_{\ell\text{-shape}}$ measures how well points within the 3D bounding box align with two anchoring 2D edges, E_1 and E_2 . Of the eight corners defining the box, the top four corners form edges, and the two closest to the ego reference point are selected. Minimizing the distance of object points p_i to the closest edge $E \in \{E_1, E_2\}$ ensures the bounding box aligns with the most relevant structural edges. In this case, $d(p_i, E)$ is the euclidean distance from p_i to E . $\mathcal{J}_{\ell\text{-shape}}$ is defined as:

$$\mathcal{J}_{\ell\text{-shape}}(\theta, \mathcal{P}_{\text{obj}}) = \sum_{p_i \in \mathcal{P}_{\text{obj}}} \frac{\min_{E \in \{E_1, E_2\}} d(p_i, E) \cdot \mathbf{1}(p_i \in B_{3\text{D}}(\theta))}{\mathcal{G}_{\text{inside}}(\theta, \mathcal{P}_{\text{obj}})}. \quad (4.5)$$

The surface fitting term $\mathcal{J}_{\text{surface}}$ assigns object points to the box faces nearest to the ego, aligning surfaces closer to the ego reference point $\mathbf{e}$. This is achieved by introducing a bias that pushes the bounding box away from the ego, clipped by a constant C_{surface} in the cost function. $\mathcal{J}_{\text{surface}}$ is defined as:

$$\mathcal{J}_{\text{surface}}(\theta, \mathbf{e}) = -\min(\|(\theta_x, \theta_y) - (\mathbf{e}_x, \mathbf{e}_y)\|_2, C_{\text{surface}}). \quad (4.6)$$

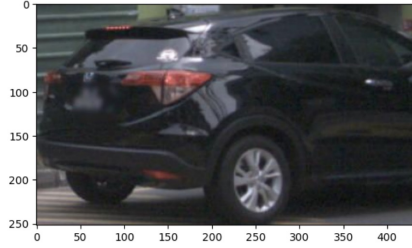
Finally, consistency within the image space is measured by applying calibration transformations T_{ext} and T_{int} . These transformations project the 3D bounding box θ into its 2D counterpart in the image space using $P_{3\text{D} \rightarrow 2\text{D}}$. The resulting projection is evaluated against the original 2D bounding box using the IoU metric:

$$\mathcal{J}_{2\text{D}}(\theta, b_{\text{img}}) = \gamma \cdot \text{IoU}_{2\text{D}}(P_{3\text{D} \rightarrow 2\text{D}}(\theta), b_{\text{img}}). \quad (4.7)$$

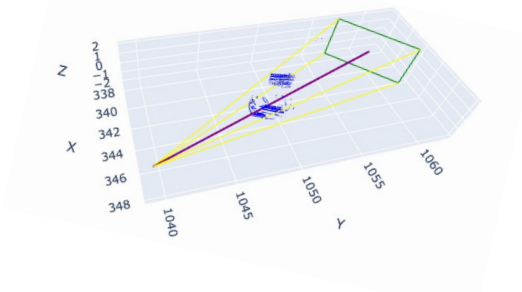
The optimization is governed by a cost function that balances multiple objectives. The objectives are weighted by control constants $(\lambda_1, \lambda_2, \lambda_3, \gamma)$.

Due to the non-differentiability and non-convexity of our cost function, evolutionary algorithms that are not gradient-dependant are employed to directly search the parameter space for optimal bounding box predictions. Specifically, our method employs the Particle Swarm Optimization (PSO) [24] to iteratively refine the parameter space with increasing granularity. Box candidates are initialized within localized regions around object clusters to enhance exploration efficiency. 3D bounding boxes with insufficient $\text{IoU}_{2\text{D}}$ relative to their corresponding 2D proposals are discarded. Each scene frame is annotated, and results are compiled into a novel object bank. Duplicate 3D bounding boxes, arising from multi-view overlaps or multiple clusters for the same object, are resolved using NMS. Additionally, ImmortalTracker [54] performs multi-object tracking to filter noisy annotations and estimate velocities for each novel object.

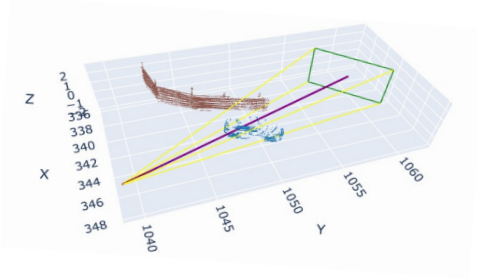
1. For each 2D novel object proposal



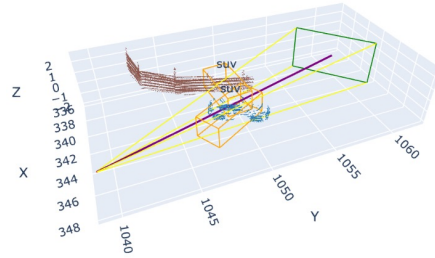
2. Project a 3D frustum and extract points



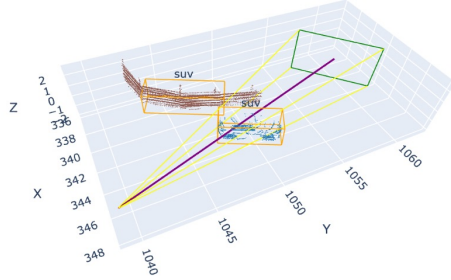
3. Identify clusters within 3D frustum and form cross-modal proposals



4. Initialize random 3D bounding boxes for each proposal



5. Optimize 3D bounding box parameters for each proposal



6. Filter out invalid 3D bounding boxes and select box that is closest to ego

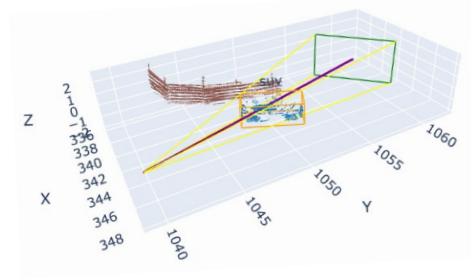


Figure 4.3: **Illustration of Cross-Modal Association and Adaptive 3D Box Search.** Starting with a 2D novel object proposal (1), a 3D frustum is projected and points are extracted (2). Object clusters within the frustum are identified and paired with the proposal (3). Random 3D boxes are initialized (4), optimized (5), and filtered to retain the box closest to the ego vehicle (6).

4.3 Model Architecture

The architecture of OV-SCAN comprises the off-the-shelf TransFusion-L [1] 3D detector and our Hierarchical Two-Stage Alignment (H2SA) head. TransFusion-L processes a LiDAR point cloud to generate 3D object embeddings for novel objects and decodes their bounding boxes via its regression head. Meanwhile, H2SA transforms the 3D object embeddings into predicted alignment features for prompt-based classification.

4.3.1 Hierarchical Two-Stage Alignment (H2SA) Head

To enable strong semantic alignment between 3D objects and 2D image embeddings, H2SA introduces a hierarchical two-stage alignment process that builds on TransFusion-L [1]. In Stage 1, H2SA employs high-level classification as an auxiliary task to generate text prompts that guide downstream alignment. Specifically, TransFusion-L predicts a high-level novel class c , from which a class-based text prompt embedding $\hat{\mathcal{A}}_{\text{text}} \in \mathbb{R}^D$ is derived. This embedding serves as a semantic anchor for aligning 3D object features $\mathcal{O}_{3D} \in \mathbb{R}^H$ with 2D target embeddings $\mathcal{A}_{2D} \in \mathbb{R}^D$, extracted using the CLIP ViT-H/14 model trained on DFN-5B [16]. Notably, CLIP’s image and text encoders output high-dimensional representations where $D = 8H$, necessitating the upsampling of 3D features to facilitate alignment.

To bridge this dimensional gap, H2SA generates multi-scale text prototypes using three embedding scales $s \in \{H, 2H, 4H\}$, as illustrated in Figure 4.4. The class-based prompt $\hat{\mathcal{A}}_{\text{text}}$ is passed through a series of linear layers to yield initial prototypes $\{W_H, W_{2H}, W_{4H}\}$. These are then refined through a Cross-Modal Distillation Block (CMDB), which integrates 3D object features and progressively enhances alignment fidelity across scales. The first CMDB step applies multi-head attention to the base prototype and 3D features:

$$W'_H = \text{LN}(\text{MHA}(W_H, \mathcal{O}_{3D}, \mathcal{O}_{3D}) + W_H), \quad (4.8)$$

where LN denotes layer normalization and MHA denotes multi-head attention. The refined prototype W'_H is then fused with the original 3D embedding via channel-wise concatenation, followed by a feed-forward network to produce a unified representation at the next scale:

$$U_{2H} = \text{LN}(\text{FFN}(\text{concat}(W'_H, \mathcal{O}_{3D}))). \quad (4.9)$$

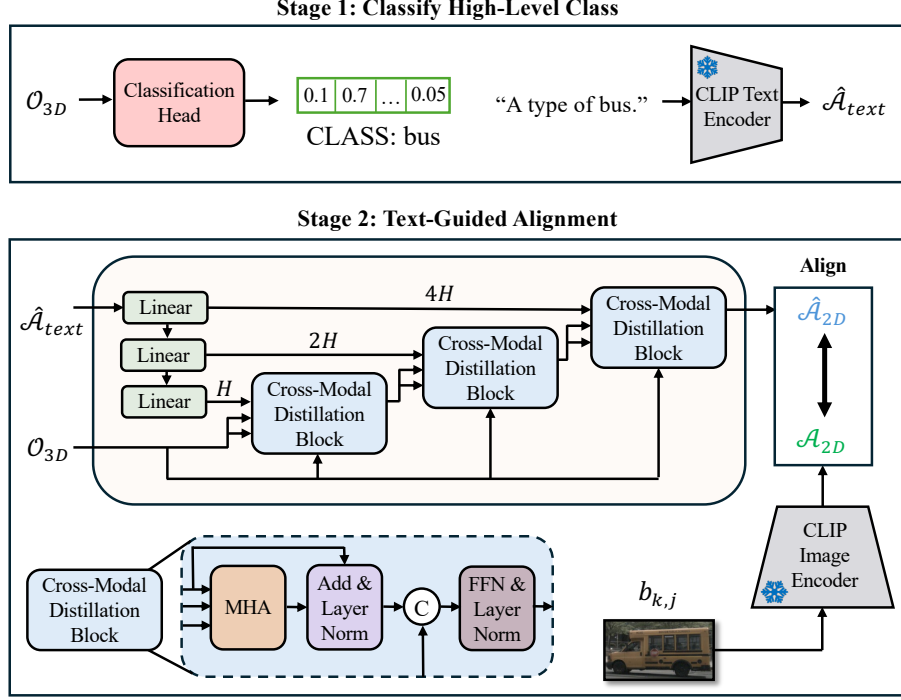


Figure 4.4: **Illustration of the Hierarchical Two-Stage Alignment (H2SA) Head.** H2SA first predicts the high-level novel classes, then derives class-based text prompts. H2SA then uses text prototypes to incrementally map 3D features to their 2D counterparts.

This iterative refinement continues across scales, enabling the model to integrate both coarse and fine semantic cues. By aligning to CLIP-derived targets with progressively enriched text prototypes, H2SA facilitates robust cross-modal alignment. Importantly, the entire alignment head is lightweight and seamlessly integrates into TransFusion-L, preserving its efficiency and real-time inference capability.

4.3.2 Multi-Sensor Fusion

To show that OV-SCAN is also effective in the multi-sensor setting, we introduce OV-SCAN-Fusion, which integrates an image backbone following BEVFusion [32]. We adopt pre-trained InternImage-T [55] to prevent distribution bias, since BEVFusion’s SWIN-T [31] backbone is pre-trained on NuImages [2].

4.4 Training

4.4.1 Training Loss

To train OV-SCAN, our method employs the TransFusion-L loss [1] with an addition alignment loss term. We define the loss as:

$$\ell_{\text{overall}} = w_{\text{heatmap}} \cdot \ell_{\text{heatmap}} + w_{\text{box}} \cdot \ell_{\text{box}} + w_{\text{class}} \cdot \ell_{\text{class}} + w_{\text{align}} \cdot \ell_{\text{align}}, \quad (4.10)$$

where the TransFusion loss terms are defined as:

- ℓ_{heatmap} : a *Gaussian focal loss* that penalizes discrepancies between the predicted objectness heatmap and Gaussian kernels centered on ground-truth object locations.
- ℓ_{box} : a L_1 *regression loss* applied to the 3D bounding box parameters (center, size, orientation, and velocity).
- ℓ_{class} : a *sigmoid focal classification loss* that down-weights easy negatives while focusing learning on hard positives. Guides high-level classification for class-based heatmaps.

Each term is weighted by coefficients w_{heatmap} , w_{box} , and w_{class} defined in the training configuration. To enable cross-modal embedding alignment, we incorporate an additional cross-modal alignment loss term into the overall loss function weighted by w_{align} :

$$\ell_{\text{align}} = \frac{\sum_{i=1}^{N_{\text{pos}}} \left(1 - S_C \left(\text{H2SA}(\mathcal{O}_{3\text{D}}^{(i)}, \hat{\mathcal{A}}_{\text{text}}^{(i)}, \mathcal{A}_{2\text{D}}^{(i)}) \right) \right) \cdot \mathbb{1}_{\text{align},i}}{\sum_{i=1}^{N_{\text{pos}}} \mathbb{1}_{\text{align},i}}, \quad (4.11)$$

where S_c is cosine similarity.

All N_{pos} positively matched prediction-target pairs contribute to each loss term, except for the alignment loss. Only pairs with targets selected by the selective alignment filter, $\mathbb{1}_{\text{align},i}$, contribute to the alignment loss. We adopt the same class-based data augmentation strategies as prior LiDAR-only works [70] by using novel class c . Specifically, we balance each training sample by keeping a database of object point clouds and ensuring class distribution balance through the targeted injection of objects during training.

4.4.2 Selective Alignment

Our method motivates the idea of having only high-quality alignment pairs contribute to the alignment loss during training, while all 3D annotations remain included in the box-related losses. Prior works [15, 5] restrict alignment filtering to multi-view alignment methods, as formulated in Eq. (4.7). However, as illustrated in Figure 1.2a, such filtering is insufficient in cases of partial occlusion or low resolution, where multi-view alignment appears nearly perfect. To address this, SC-NOD deploys additional filters to detect and exclude unreliable alignments, ensuring they do not contribute to the alignment loss during training.

To mitigate *occlusion-induced corruption*, the instance mask m of each novel object proposal is used to filter out pairs with inadequate object representation. Specifically, the proportion of pixels occupied by the object within the 2D crop is thresholded by τ_{occ} :

$$\frac{\sum_{x=1}^{W_{\text{crop}}} \sum_{y=1}^{H_{\text{crop}}} m(x, y)}{H_{\text{crop}} W_{\text{crop}}} \leq \tau_{\text{occ}}. \quad (4.12)$$

This process identifies and excludes occluded objects from cross-modal alignment. Different values of τ_{occ} are selected depending on the novel class, outlined in Tab. 5.4.

Next, SC-NOD filters out samples associated with image crops having an insufficient number of pixels, as these are prone to *resolution-induced* noise. Samples with a resolution below the threshold τ_{res} are marked as having insufficient semantic resolution for discriminative 2D features:

$$H_{\text{crop}} W_{\text{crop}} \leq \tau_{\text{res}}. \quad (4.13)$$

4.5 Prompt-based Classification

To classify objects, OV-SCAN employs a prompt-based classification strategy that leverages the alignment embeddings produced by H2SA. Objects are first classified into fine-grained subclasses and then mapped to their associated high-level novel classes for evaluation. To enable this, the frozen CLIP text encoder is used to generate subclass text embeddings from the template prompt "a type of {SUBCLASS}.". For each detected object, the fine-grained subclass is selected by identifying the subclass whose text embedding

has the highest similarity to the predicted alignment embedding. Formally, the predicted subclass label $\hat{c}_{\text{fg}}$ is computed as:

$$\hat{c}_{\text{fg}} = \arg \max_{c_i \in C_{\text{fg}}} S_C(\text{H2SA}(\mathcal{O}_{3\text{D}}, \hat{\mathcal{A}}_{\text{text}}), t_{c_i}), \quad (4.14)$$

where C_{fg} denotes the set of fine-grained subclasses, t_{c_i} is the text embedding for subclass c_i , and $S_C(\cdot)$ denotes the similarity function. This approach allows OV-SCAN to infer high-level novel classes by mapping each fine-grained subclass to its corresponding broader category (e.g., mapping “minivan” to “car”).

Chapter 5

Experiments

In this chapter, we present the NuScenes and KITTI datasets used to evaluate the performance of OV-SCAN in open-vocabulary 3D object detection. We conduct a series of evaluations to quantify OV detection accuracy, including ablation studies on the SC-NOD and H2SA components to assess their individual contributions. Additionally, we provide both quantitative and qualitative comparisons with existing methods to benchmark performance.

5.1 Datasets

Our OV-3D object detection experiments are conducted on the nuScenes [2] and KITTI [17] datasets, two widely-used benchmarks in the field of autonomous driving and 3D perception. To simulate an open-vocabulary setting, we treat all object classes as novel and rely exclusively on human-annotated labels for evaluation purposes, without using them during training. This setup allows us to rigorously assess the generalization capability of OV-SCAN to previously unseen object categories.

The nuScenes dataset provides a large-scale, diverse urban driving environment collected in Boston and Singapore. It includes 1,000 driving scenes, each lasting 20 seconds, with full sensor coverage including 6 cameras, 5 radars, and a LiDAR sensor. The dataset contains a larger range of object categories annotated in 3D and offers dense annotations across time, enabling evaluation under varied traffic scenarios and conditions. We use nuScenes to test OV-SCAN’s performance in complex, dynamic settings with a wide range of object types and environmental variability.

The KITTI dataset, in contrast, offers a more constrained but well-established benchmark for 3D object detection. It includes data from a front-facing stereo camera and LiDAR sensor, captured in semi-urban and highway settings in Karlsruhe, Germany. KITTI contains annotations for 3 classes: cars, pedestrians, and cyclists, but we reinterpret the task by treating these as novel classes in our open-vocabulary setup. Despite its limited class diversity, we leverage KITTI to demonstrate the generalization ability of OV-SCAN to autonomous driving datasets outside of NuScenes.

5.2 Implementation Details

5.2.1 Novel Object Discovery

OV-SCAN relies on Grounding DINO [29] and SAM [25] to detect 2D proposals from a given list of (user-defined) novel classes. Tab. 5.1 outlines the novel classes we used for our experiments for both the nuScenes [2] and KITTI [17] datasets. Notably, we adhere to each dataset’s original taxonomy, with the exception of splitting “cyclist” into “bicycle” and “motorcycle” for KITTI.

Table 5.1: **Novel classes used to identify novel object proposals for each dataset.** These classes are referred to as “novel” since their ground truth labels, although available in the respective datasets, are not used in training OV-SCAN.

Dataset	Novel Classes
nuScenes	car, truck, pedestrian, bicycle, motorcycle, bus, traffic cone, barrier block, construction vehicle
KITTI	car, van, truck, tram, bicycle, motorcycle, pedestrian, person sitting

In post-processing, our method adopts a similar approach to UP-VL [42] for removing false positives. We treat each novel class as a positive class while including a set of background classes. To refine the results, we cross-reference each image crop from Grounding DINO with CLIP [47], filtering out background classes such as “vegetation”, “fence”, “gate”, “curb”, “sidewalk”, “wall”, “building”, “railing”, and “rail guard.”

5.2.2 Adaptive 3D Box Search

To localize and regress 3D bounding box parameters for novel object proposals, OV-SCAN employs particle swarm optimization (PSO) [24] to solve the continuous nonlinear optimization problem defined in Sec. 4.2.3. Each object proposal corresponds to a unique optimization instance for estimating the 3D bounding box parameterized as $\theta = (x, y, z, l, w, h, r_y)$. We perform 150,000 total iterations per cross-modal proposal, distributed across a swarm of 50 particles with 3000 iterations each. The PSO hyperparameters used are summarized in Table 5.2, with inertia weight w annealed from 10.0 to 0.1 using a cosine schedule to balance exploration and exploitation, and both cognitive (c_1) and social (c_2) coefficients set to 1.0. Candidate positions (x, y, z) are initialized in regions with high likelihood of corresponding to true object centers—half at the closest point to the center frustum ray, and half at the mean of the object point cluster $\mathcal{P}_{\text{obj}}$. To accommodate varying object sizes, initialization noise is sampled from $\mathcal{N}(0, C_{\text{noise}} \cdot \frac{\mathbf{A}_{\text{max}} + \mathbf{A}_{\text{min}}}{2})$ with $C_{\text{noise}} = 0.1$, and added to the positions. The remaining parameters (l, w, h, r_y) are initialized uniformly. The objective function is balanced using weighting coefficients $(\lambda_1, \lambda_2, \lambda_3, \gamma) = (5.0, 1.0, 1.0, 3.0)$, and the anchor priors follow Find n’ Propagate [15].

Table 5.2: PSO hyperparameters used in adaptive 3D box search.

Parameter	Description	Value
N_{swarm}	Swarm size	50
N_{iter}	Iterations per particle	3000
w_{init}	Initial inertia weight	10.0
w_{end}	End inertia weight	0.1
c_1	Cognitive coefficient	1.0
c_2	Social coefficient	1.0
C_{noise}	Initialization noise	0.1

5.2.3 Training

OV-SCAN is trained on 8 NVIDIA V100 GPUs with a batch size of 4 for 20 epochs. OV-SCAN adopts the AdamW optimizer with a learning rate 0.001 and a weight decay of 0.01. The CLIP image and text encoders are kept frozen. For OV-SCAN-Fusion, we add an image backbone on top of pre-trained OV-SCAN and fine-tune. OV-SCAN-Fusion is trained for 5 additional epochs using a cosine annealing schedule initialized at a learning rate of 0.0001. We define the weights on the loss function, mentioned in Eq. (4.10), as

$(w_{\text{heatmap}}, w_{\text{box}}, w_{\text{class}}, w_{\text{align}}) = (1, 0.25, 0.25, 1)$. This work is built on OpenPCDet [53], an open-source 3D object detection toolbox implemented using PyTorch.

5.2.4 Prompt-based Classification

As mentioned in Sec. 4.5, OV-SCAN employs a specific-to-broad strategy, relying exclusively on H2SA for classification. First objects are classified into fine-grained subclasses before being mapped to their respective novel classes for evaluation. For the nuScenes dataset, the fine-grained subclasses used are outlined in Tab. 5.3. OV-SCAN follows the same procedure for the KITTI dataset.

Table 5.3: **Fine-grained subclasses for each high-level novel class in the nuScenes dataset.**

Novel Class	Fine-grained Subclasses
car	sedan, van, minivan, hatchback, suv, coupe, police car, sprinter van, taxi
truck	pickup truck, tow truck, semi-truck, gasoline truck, delivery truck, garbage truck, fire truck, flatbed truck, ambulance, cement truck, dump truck
bus	school bus, coach bus, double-decker bus, transit bus, shuttle bus, minibus
trailer	portable message board trailer, flatbed trailer, freight trailer, cargo trailer
construction vehicle	excavator, bulldozer, forklift, construction loader, construction lift
pedestrian	adult, construction worker, police officer, child
motorcycle	cruiser motorcycle, sport motorcycle, touring motorcycle, moped
bicycle	bicycle
traffic cone	traffic cone, traffic drum, traffic delineator post
barrier	plastic jersey barrier, concrete jersey barrier

5.2.5 Selective Alignment

During selective alignment, OV-SCAN uses class-specific thresholds τ_{occ} for the occlusion filter since objects naturally occupy varying amounts of space within a 2D proposal. As illustrated in Figure 5.1, a car generally occupies a larger area in its instance mask compared to a pedestrian, resulting in a significantly higher proportion of pixels classified as instance pixels. Consequently, the pixel distribution within instance masks varies considerably across classes, making a uniform threshold for determining occlusion level inadequate.

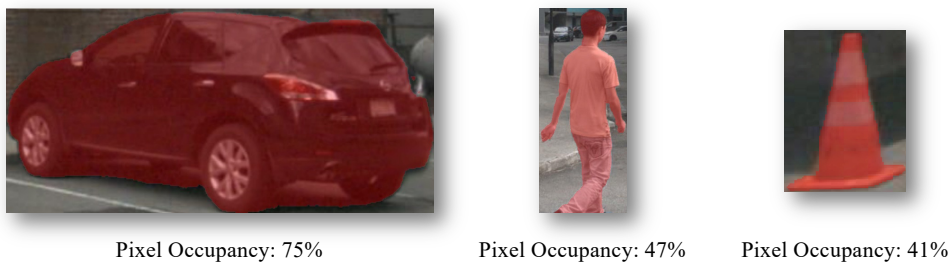


Figure 5.1: **Percentage of instance pixels across object classes.** Vehicles, such as cars, typically occupy a greater proportion of pixels within their 2D proposals compared to objects like pedestrians and traffic cones.

As shown in Tab. 5.4, reasonable values for τ_{occ} are manually determined for each class to address variability. The resolution threshold $\tau_{res} = 4000$ provided the best balance between removing overly small crops and retaining a sufficient number of training samples.

Table 5.4: Parameterization of τ_{occ} for different novel classes.

Novel Class	τ_{occ}
car	0.5
truck	0.5
pedestrian	0.25
bicycle	0.4
motorcycle	0.4
bus	0.5
traffic cone	0.25
barrier block	0.35
construction vehicle	0.5

5.3 Evaluation Metrics

To evaluate the OV-3DOD performance on the nuScenes benchmark refer to traditional 3D object detection metrics. Mean Average Precision (mAP) is the area under the precision-recall curve obtained by matching predicted 3D boxes to ground-truth boxes whose centres fall within radial thresholds of 0.5 m, 1 m, 2 m, and 4 m. The final mAP is the arithmetic mean over all ten object classes and all four thresholds, capturing the overall precision-recall trade-off. nuScenes Detection Score (NDS) augments mAP by combining it with five true-positive quality terms—translation, scale, orientation, velocity, and attribute—thereby rewarding detectors that deliver geometrically, kinematically, and semantically faithful boxes. Per-class AP values for ten categories are also reported to highlight class-specific performance.

For the KITTI benchmark, AP is evaluated on the *car*, *pedestrian*, and *cyclist* classes at a stricter 3D IoU threshold of 0.5, enforcing tighter spatial alignment than nuScenes. To stress-test algorithms further, KITTI groups every ground-truth object into one of three difficulty levels—Easy, Moderate, and Hard—based on bounding-box height, object truncation, and occlusion. Reporting AP separately for these three settings reveals how well a detector copes as visual conditions become harsher—smaller, partially hidden, and truncated objects are markedly tougher to localise and classify.

Following OpenSight [66], our ablation studies further break down performance by sensor range, reporting $\text{mAP}_{n/m/f}$ over the near (0–18 m), mid (0–34 m), and far (0–54 m) bands to evaluate how detection accuracy changes with distance.

5.4 Results

5.4.1 Performance on NuScenes

The OV-3D object detection results for top-performing methods on the nuScenes dataset can be seen in Tab. 5.5. For a fair comparison to the other methods, which are all LiDAR-based, we only consider OV-SCAN and not OV-SCAN-Fusion. Our LiDAR-based method surpasses previous benchmarks by achieving the best mAP and NDS, thus surpassing OpenSight by 8.2 mAP and 9.5 NDS. Without being given 3D human-annotations, OV-SCAN achieves an AP score above 60 for both car and pedestrian categories. We also note that our method exceeds previous benchmarks by more than 5 AP for 6 out of the 10 classes. Furthermore, we show that simply adding camera as an additional input modality

Table 5.5: **Results on nuScenes.** We report the overall mAP, NDS, and individual class APs. All classes are novel (*i.e.*, no human-annotations are used to train except in CoDA). CoDA uses “car” as a base-class. Best in **bold** and “*” denotes omitted methods for fair comparison.

Method	mAP	NDS	Car	Ped.	Truck	Motorcyc.	Bicyc.	T. Cone	Bus	Barrier	Con.V.	Trailer
OV-3DET [34]	5.7	12.0	16.1	21.1	2.9	2.0	1.8	11.0	1.3	0.3	0.3	0.0
CoDA [5]	10.3	16.3	85.5	3.5	1.9	2.0	5.9	3.9	0.0	0.0	0.0	0.0
UP-VL [42]	11.3	17.2	19.4	30.8	7.9	16.8	14.1	15.6	3.1	0.5	4.3	0.4
Find n’ Prop. [15]	16.7	22.4	24.3	22.8	8.6	35.8	34.7	21.3	11.1	4.4	4.1	0.1
OpenSight [66]	22.9	23.3	25.3	52.5	11.6	30.4	26.1	25.6	5.1	42.2	8.7	0.8
OV-SCAN	31.1	32.8	61.6	60.1	30.3	39.8	31.0	39.6	22.0	18.8	6.8	0.6
OV-SCAN-Fusion*	33.8	34.4	62.0	57.6	34.2	44.7	40.2	44.6	24.4	18.9	10.8	0.8
Supervised w/ Boxes*	49.8	60.8	81.4	75.4	34.7	55.1	39.4	62.5	52.7	70.8	13.6	12.4

to OV-SCAN and then fine-tuning can improve the overall performance. Finally, as an upper bound, we report the performance of our method using ground truth boxes instead of those generated by SC-NOD. CLIP is still used to derive high-level class labels and align.

5.4.2 Performance on KITTI

In Tab. 5.6, we present our results on KITTI, where all classes—*car*, *pedestrian*, and *cyclist*—are treated as novel, meaning no class-specific supervision is used during training. We report $AP_{3D@50}$ across the standard difficulty levels, following the official KITTI evaluation protocol. These levels are defined based on object characteristics such as bounding box height, truncation percentage, and occlusion level. Specifically, Easy samples correspond to fully visible objects with minimal truncation, while Moderate and Hard categories allow for increasing degrees of occlusion and partial visibility. Evaluation is performed using a 3D IoU threshold of 0.5 for all classes.

Table 5.6: **KITTI $AP_{3D@50}$ results for OV-SCAN.** All classes are treated as novel.

Class	Easy	Moderate	Hard
Car	55.02	45.10	38.40
Pedestrian	33.01	27.61	24.03
Cyclist	44.81	29.81	28.53

In Tab. 5.7, we compare our results to other state-of-the-art OV-3DOD methods. OV-SCAN outperforms OV-3DET [34] and ImOV3D [59] in the overall metric, achieving com-

parable results to ImOV3D [59] in the car category while surpassing both in the other two classes.

Table 5.7: **Results on KITTI.** We report the overall $AP_{3D@50}$ for each class at medium difficulty. Best in **bold**.

Method	Car	Ped.	Cyclist	$mAP_{3D@50}$
OV-3DET [34]	42.65	15.71	18.20	25.52
ImOV3D [59]	45.51	19.53	28.00	31.01
OV-SCAN	45.10	27.61	29.81	34.17

5.4.3 Novel Object Discovery

By treating all objects as novel, OV-SCAN relies exclusively on SC-NOD to supplement the high-quality annotations previously provided by human labelers. SC-NOD produces a total of 319,028 annotations for training purposes. This represents a substantial subset, yet still only a portion, of the full set of 797,179 ground-truth 3D annotations that are available in the nuScenes dataset. In Figure 5.2, we examine the high-level class distribution of training labels, where SC-NOD-generated annotations are categorized using CLIP. As shown in Figure 5.2a, the absolute number of annotations in SC-NOD is consistently lower than that of the human-labeled nuScenes dataset. Common object categories such as *car*, *pedestrian*, and *barrier* dominate in both sources, while infrequent classes like *bicycle*, *trailer*, and *bus* are similarly underrepresented. This similarity becomes more apparent in the normalized comparison shown in Figure 5.2b, where annotation counts are expressed as class-wise proportions. Despite the reduced volume, SC-NOD preserves the relative distribution of categories, ensuring that training exposure across object types mirrors the original nuScenes dataset.

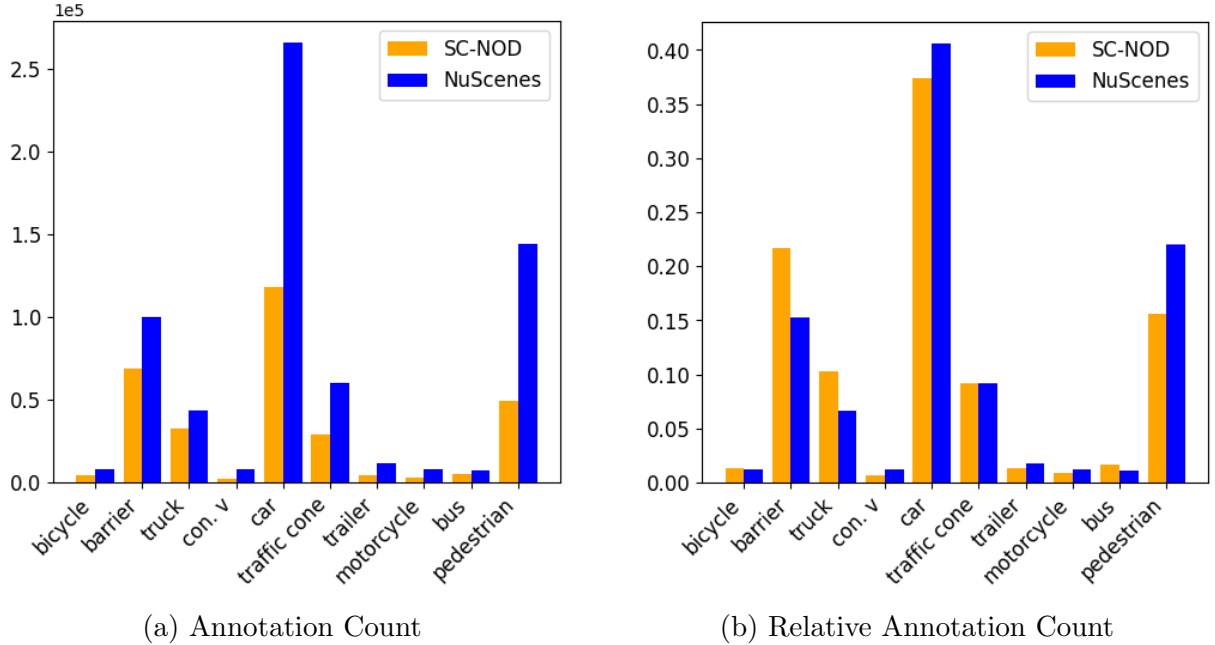


Figure 5.2: **Distribution of training labels.**

5.4.4 Selective Alignment

While all annotations contribute to the box loss, only 171,532 (54%) of those generated are utilized for cross-modal alignment. The remainder of generated annotations are excluded as a result of filtering due to significant occlusion (39%) or insufficient resolution (7%).

Traditional metrics often fall short in capturing the full capabilities of OV models. To better quantify performance, we evaluate the semantic alignment between predictions and ground truth using cosine distance on matched prediction-target pairs from the nuScenes validation set. Following the approach used during alignment training, we obtain target embeddings by projecting 3D ground truth boxes into the 2D image space and extracting corresponding image features using CLIP. To ensure a fair comparison, we exclude targets affected by semantic ambiguity using our selective alignment filter. During evaluation, we form prediction-target pairs and compute the cosine distance between their respective embeddings as a measure of semantic similarity. In Figure 5.3, we compare OV-SCAN against a baseline similar to OpenSight [66], plotting the cumulative distribution of cosine distances for the subset of targets that are successfully recovered by both detectors. The cumulative distribution function (cdf) shows that OV-SCAN achieves better 3D-2D em-

bedding alignment than the baseline, as evidenced by a leftward shift in cosine distance distribution on the nuScenes validation set.

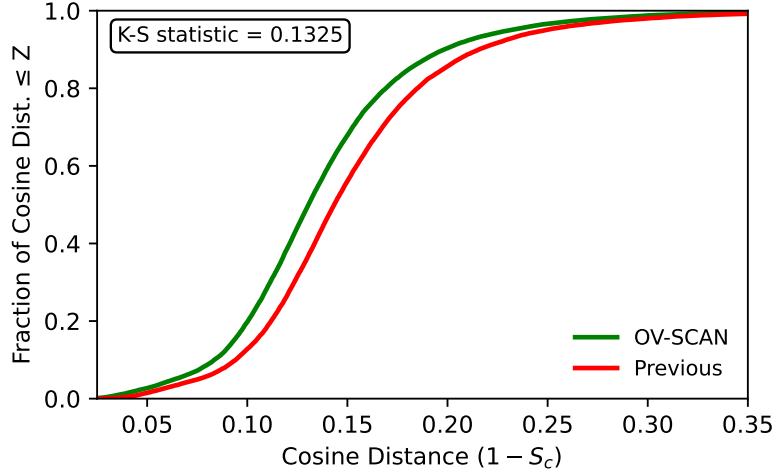


Figure 5.3: **Cross-modal Alignment Performance.** The red CDF shows the distribution of the distance between the 3D embedding produced by a baseline OV-3D detector and the corresponding 2D embedding from CLIP on the nuScenes validation set. The green CDF shows this distribution using OV-SCAN instead of the baseline. The latter is shifted to the left, showing an improved alignment between the 3D and 2D embeddings due to supervision by our higher quality 3D-2D proposal pairings and our alignment head.

5.5 Ablation Studies

5.5.1 Adaptive 3D Box Search Ablation Study

To assess the effectiveness of our adaptive 3D box search in SC-NOD, we evaluate its performance on the nuScenes dataset, comparing it to the greedy search approach used in Find N’ Propagate [15]. In their work, the authors report using 150,000 candidate boxes per proposal, a setting we adopt in our implementation. As shown in Tab. 5.8, our method consistently outperforms the greedy search baseline, even when operating with fewer iterations per novel object. To further understand the relationship between search budget and performance, we systematically reduce the number of iterations per proposal. This analysis highlights a practical trade-off between offline pseudo-labeling cost and overall detection accuracy, offering users flexibility in balancing efficiency and performance.

Table 5.8: **Ablations Study on Adaptive 3D Box Search.** We evaluate performance on nuScenes using different box search methods and varying numbers of search iterations per proposal.

Method	Iter.	mAP	mAP _n	mAP _m	mAP _f
Greedy	150,000	25.4	38.3	29.2	22.4
Adaptive	37,500	28.9	47.8	33.8	25.4
Adaptive	75,000	29.2	47.5	34.3	25.8
Adaptive	150,000	31.1	48.1	35.6	27.7

We further ablate various parameterizations of the cost function during the adaptive 3D box search, as seen in Tab. 5.9, highlighting the importance of balancing the heuristic cost terms for effective 3D box search. The ablation reveals that the density term provides a strong baseline, but significant gains arise from incorporating both ℓ -shape and surface alignment terms at a low-cost. The best performance (31.1 mAP) is achieved when all three terms are combined, confirming that geometric structure and egocentric cues are crucial for high-quality box fitting.

Table 5.9: **Ablation Study on 3D Box Search Cost Weights.** Selected setting is marked in gray .

$(\lambda_1, \lambda_2, \lambda_3, \gamma)$	mAP
(5.0, 0.0, 0.0, 3.0)	26.1
(1.0, 1.0, 1.0, 1.0)	27.4
(5.0, 2.5, 2.5, 3.0)	27.8
(5.0, 1.0, 1.0, 3.0)	31.1

5.5.2 Selective Alignment Ablation Study

Together, Tab. 5.10a and Tab. 5.10b demonstrate the effectiveness of filtering techniques used in selective cross-modal alignment. A simple occlusion filter with a fixed threshold τ_{occ} yields a notable performance gain, while class-based thresholds achieve the highest improvement (+1.7 mAP). For the resolution filter, optimizing the threshold to balance resolution quality and the number of filtered samples proves most effective (+0.7 mAP).

Table 5.10: **Ablation Study on Selective Alignment Filters.** Selected settings are marked in gray .

(a) Resolution Filter		(b) Occlusion Filter	
Filter	mAP	Filter	mAP
w/o	30.4	w/o	29.4
$\tau_{\text{res}} = 1000$	30.8	fixed ($\tau_{\text{occ}} = 0.35$)	30.7
$\tau_{\text{res}} = 4000$	31.1	class-based τ_{occ}	31.1
$\tau_{\text{res}} = 10000$	30.6		

5.5.3 Alignment Head Ablation Study

To evaluate the effectiveness of the proposed H2SA head, we introduce a simplified one-stage baseline for comparison, as shown in Tab. 5.11. In this variant, the classification loss is removed, the class-specific heatmaps from TransFusion-L are merged into a single class-agnostic heatmap, and the text-guided alignment module is replaced with a simple feed-forward network (−5.9 mAP). We also examine a two-stage version that retains the decoding structure but does not incorporate semantic alignment (+4.2 mAP). Finally, by integrating class-based text prompts into the alignment process, our full H2SA head achieves the best performance (+1.7 mAP).

Table 5.11: **Ablation Study on Alignment Head Design.** Selected settings are marked in gray .

Alignment Head	mAP
One-Stage + FFN	25.2
Two-Stage + FFN	29.4
H2SA	31.1

5.6 Qualitative Results

Figure 5.4 presents qualitative results of OV-SCAN, showing 3D point clouds with predicted bounding boxes and fine-grained open-set labels. Object points highlighted in red are paired with images for visual comparison.

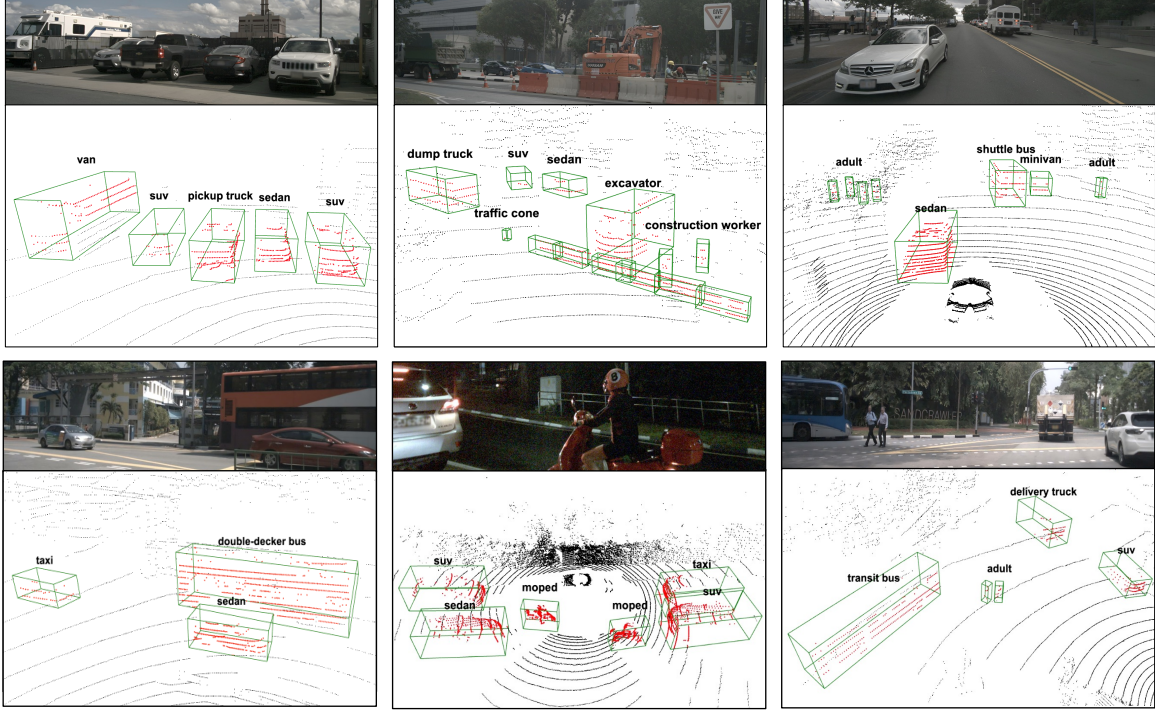


Figure 5.4: **Visualization results using open-vocabulary on a variety of traffic conditions.** OV-SCAN performs inference on a set of urban scenes identifying a diverse set of objects. To demonstrate OV capabilities, objects are classified into more fine-grained classes as opposed to traditional closed-set classes.

In Figure 5.5, we perform a side-by-side comparison of the box regressed when fueled by SC-NOD vs. Greedy Box Seeker from Find n’ Propagate [15]. Both methods use Transfusion-L [1] off-the-shelf, while displaying detections with a confidence over 0.05. However, during training, Find n’ Propagate natively follows its predefined “Setting 2”, treating three classes (*car*, *pedestrian*, and *bicycle*) as base and leaving the remaining classes as novel. In contrast, OV-SCAN treats every class as novel. OV-SCAN regresses notably more precise bounding boxes for novel classes. Additionally, Find n’ Propagate tends to produce more false positives due to their increased recall strategy, whereas OV-SCAN maintains better precision and localization accuracy.

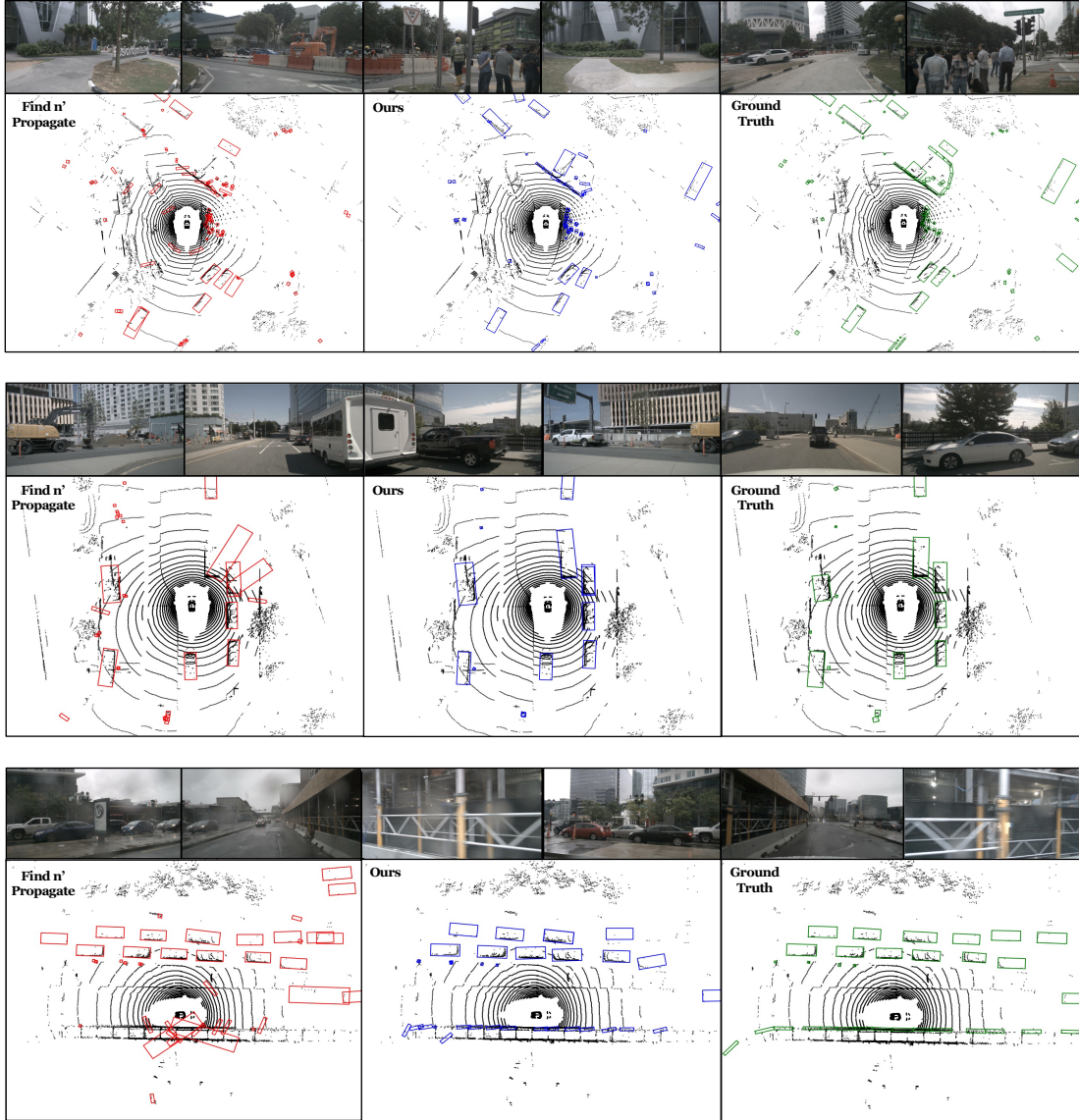


Figure 5.5: **Comparison between OV-SCAN and Find n' Propagate [15].** OV-SCAN regresses more precise bounding boxes than Find n' Propagate without requiring any human-annotated labels. We compare to Find n' Propagate in “Setting 2” which uses 3 base classes (*car*, *pedestrian*, and *bicycle*) leaving the rest as novel.

Chapter 6

Conclusion

6.1 Conclusion

We introduce OV-SCAN, an OV-3D object detector that achieves strong OV detection performed through enhanced cross-modal alignment. Without human-provided 3D annotations, SC-NOD accurately generates 3D boxes through a proposed non-linear continuous optimization. To address noise sources like annotation errors, occlusions, and resolution limits, we propose noise mitigation techniques enforcing semantic consistency across 3D-to-2D pairs. By adapting H2SA, we further strengthen alignment and enable robust open-set classification. Experiments on nuScenes and KITTI demonstrate state-of-the-art OV-3D detection.

6.2 Limitations and Future Work

A primary limitation of SC-NOD lies in its restricted ability to recover annotations, as illustrated in Figure 5.2. This limitation stems from its dependence on 2D proposals. Objects clearly visible in the point cloud may remain unlabeled if they are not captured in multi-view images. Consequently, such objects are omitted from the training set. Additionally, detecting and labeling sparse or distant objects remains a persistent challenge, leading to a marked drop in performance at greater ranges, as shown in Tab. 5.8. To enhance robustness, future work could explore integrating multiple sources of object proposals beyond just those derived from 2D-OV object detectors.

Another significant limitation is the offline annotation cost associated with labeling a large-scale autonomous driving datasets. Each object proposal requires a rigorous optimization process to accurately regress 3D bounding box parameters, which is computationally intensive. While lowering the search complexity for each proposal may reduce the annotation time, it often comes at the expense of detection performance. In addition, the search cost scales with the additional novel classes which could further expand the cost. Our method addresses this by employing a multi-threaded implementation to improve throughput; however, a GPU-based optimization pipeline could further reduce time costs and improve scalability.

Lastly, in the 3D box search process, SC-NOD utilizes class-based anchors inspired by prior works. While this approach performs well for novel classes with relatively low intra-class variance, it becomes less effective for objects with high shape variability, such as trailers, buses, and construction equipment. These limitations suggest an opportunity for future research into anchor-free box parameterization strategies that can better accommodate complex object geometries.

References

- [1] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. TransFusion: Robust LiDAR-Camera Fusion for 3D Object Detection with Transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 1080–1089. IEEE, 2022.
- [2] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [3] 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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 11618–11628. Computer Vision Foundation / IEEE, 2020.
- [4] Ricardo J. G. B. Campello, Davoud Moulavi, and Jörg Sander. Density-Based Clustering Based on Hierarchical Density Estimates. In Jian Pei, Vincent S. Tseng, Longbing Cao, Hiroshi Motoda, and Guandong Xu, editors, *Advances in Knowledge Discovery and Data Mining, 17th Pacific-Asia Conference, PAKDD 2013, Gold Coast, Australia, April 14-17, 2013, Proceedings, Part II*, volume 7819 of *Lecture Notes in Computer Science*, pages 160–172. Springer, 2013.
- [5] Yang Cao, Yihan Zeng, Hang Xu, and Dan Xu. CoDA: Collaborative Novel Box Discovery and Cross-modal Alignment for Open-vocabulary 3D Object Detection. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Con-*

ference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023, 2023.

- [6] Yang Cao, Yihan Zeng, Hang Xu, and Dan Xu. Collaborative novel object discovery and box-guided cross-modal alignment for open-vocabulary 3d object detection, 2024.
- [7] 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*, pages 213–229. Springer, 2020.
- [8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.
- [9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9630–9640. IEEE, 2021.
- [10] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. VoxelNeXt: Fully Sparse VoxelNet for 3D Object Detection and Tracking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 21674–21683. IEEE, 2023.
- [11] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. YOLO-World: Real-Time Open-Vocabulary Object Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 16901–16911. IEEE, 2024.
- [12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society, 2009.
- [13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020.

- [14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.
- [15] Djamahl Etchegaray, Zi Huang, Tatsuya Harada, and Yadan Luo. Find n’ Propagate: Open-Vocabulary 3D Object Detection in Urban Environments. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, volume 15098, pages 133–151, Cham, 2025. Springer Nature Switzerland. Series Title: Lecture Notes in Computer Science.
- [16] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander T. Toshev, and Vaishaal Shankar. Data Filtering Networks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.
- [17] A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.
- [18] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *Int. J. Robotics Res.*, 32(11):1231–1237, 2013.
- [19] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary Object Detection via Vision and Language Knowledge Distillation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [21] Rui Huang, Henry Zheng, Yan Wang, Zhuofan Xia, Marco Pavone, and Gao Huang. Training an open-vocabulary monocular 3d detection model without 3d data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [22] Tzu-Heng Huang, Changho Shin, Sui Jiet Tay, Dyah Adila, and Frederic Sala. Multimodal Data Curation via Object Detection and Filter Ensembles. *CoRR*, abs/2401.12225, 2024. arXiv: 2401.12225.
- [23] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8. <https://github.com/ultralytics/ultralytics>, 2023.

- [24] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4, 1995.
- [25] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloé Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment Anything. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 3992–4003. IEEE, 2023.
- [26] 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, Long Beach, CA, USA, June 16-20, 2019*, pages 12697–12705. Computer Vision Foundation / IEEE, 2019.
- [27] Seungjae Lee, Hyungtae Lim, and Hyun Myung. Patchwork++: Fast and Robust Ground Segmentation Solving Partial Under-Segmentation Using 3D Point Cloud. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2022, Kyoto, Japan, October 23-27, 2022*, pages 13276–13283. IEEE, 2022.
- [28] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded Language-Image Pre-training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10955–10965. IEEE, 2022.
- [29] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. arXiv: 2303.05499, 2024.
- [30] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. *CoRR*, abs/2303.05499, 2023. arXiv: 2303.05499.
- [31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. 2021.

- [32] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L. Rus, and Song Han. BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird’s-Eye View Representation. In *IEEE International Conference on Robotics and Automation, ICRA 2023, London, UK, May 29 - June 2, 2023*, pages 2774–2781. IEEE, 2023.
- [33] Zhijian Liu, Xinyu Yang, Haotian Tang, Shang Yang, and Song Han. FlatFormer: Flattened Window Attention for Efficient Point Cloud Transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 1200–1211. IEEE, 2023.
- [34] Yuheng Lu, Chenfeng Xu, Xiaobao Wei, Xiaodong Xie, Masayoshi Tomizuka, Kurt Keutzer, and Shanghang Zhang. Open-Vocabulary Point-Cloud Object Detection without 3D Annotation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 1190–1199. IEEE, 2023.
- [35] Anas Mahmoud, Mostafa Elhoushi, Amro Abbas, Yu Yang, Newsha Ardalani, Hugh Leather, and Ari S. Morcos. Sieve: Multimodal Dataset Pruning Using Image Captioning Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 22423–22432. IEEE, 2024.
- [36] Pratyush Maini, Sachin Goyal, Zachary Chase Lipton, J. Zico Kolter, and Aditi Raghunathan. T-MARS: Improving Visual Representations by Circumventing Text Feature Learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.
- [37] Jiageng Mao, Yujing Xue, Minzhe Niu, Haoyue Bai, Jiashi Feng, Xiaodan Liang, Hang Xu, and Chunjing Xu. Voxel Transformer for 3D Object Detection. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 3144–3153. IEEE, 2021.
- [38] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. Simple open-vocabulary object detection. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *European Conference on Computer Vision (ECCV)*, pages 728–755, Cham, 2022. Springer Nature Switzerland.

- [39] Matthias Minderer, Alexey A. Gritsenko, and Neil Houlsby. Scaling Open-Vocabulary Object Detection. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.
- [40] Matthias Minderer, Alexey A. Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. Simple Open-Vocabulary Object Detection. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part X*, volume 13670 of *Lecture Notes in Computer Science*, pages 728–755. Springer, 2022.
- [41] Ishan Misra, Rohit Girdhar, and Armand Joulin. An End-to-End Transformer Model for 3D Object Detection. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 2886–2897. IEEE, 2021.
- [42] Mahyar Najibi, Jingwei Ji, Yin Zhou, Charles R. Qi, Xinchun Yan, Scott Ettinger, and Dragomir Anguelov. Unsupervised 3d perception with 2d vision-language distillation for autonomous driving. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 8568–8578. IEEE, 2023.
- [43] Yaqian Ning, Jie Cao, Chun Bao, and Qun Hao. DVST: Deformable Voxel Set Transformer for 3D Object Detection from Point Clouds. *Remote. Sens.*, 15(23):5612, 2023.
- [44] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 77–85. IEEE Computer Society, 2017.
- [45] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.
- [46] Filip Radenovic, Abhimanyu Dubey, Abhishek Kadian, Todor Mihaylov, Simon Vandenhende, Yash Patel, Yi Wen, Vignesh Ramanathan, and Dhruv Mahajan. Filtering, Distillation, and Hard Negatives for Vision-Language Pre-Training. In *IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023, pages 6967–6977. IEEE, 2023.

- [47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 7 2021.
- [48] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN: Point-voxel feature set abstraction for 3D object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [49] 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. *Int. J. Comput. Vis.*, 131(2):531–551, 2023.
- [50] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 770–779. Computer Vision Foundation / IEEE, 2019.
- [51] 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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 2443–2451. Computer Vision Foundation / IEEE, 2020.
- [52] Pei Sun, Mingxing Tan, Weiyue Wang, Chenxi Liu, Fei Xia, Zhaoqi Leng, and Dragomir Anguelov. SWFormer: Sparse Window Transformer for 3D Object Detection in Point Clouds. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part X*, volume 13670 of *Lecture Notes in Computer Science*, pages 426–442. Springer, 2022.

- [53] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. <https://github.com/open-mmlab/OpenPCDet>, 2020.
- [54] Qitai Wang, Yuntao Chen, Ziqi Pang, Naiyan Wang, and Zhaoxiang Zhang. Immortal Tracker: Tracklet never dies. arXiv:2111.13672, 2021.
- [55] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. *arXiv preprint arXiv:2211.05778*, 2022.
- [56] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019.
- [57] Zhenyu Wang, Yali Li, Taichi Liu, Hengshuang Zhao, and Shengjin Wang. OV-Uni3DETR: Towards Unified Open-Vocabulary 3D Object Detection via Cycle-Modality Propagation.
- [58] Yan Yan, Yuxing Mao, and Bo Li. SECOND: Sparsely Embedded Convolutional Detection. *Sensors*, 18(10):3337, 2018.
- [59] Timing Yang, Yuanliang Ju, and Li Yi. Imov3d: Learning open-vocabulary point clouds 3d object detection from only 2d images. *NeurIPS 2024*, 2024.
- [60] Lewei Yao, Jianhua Han, Youpeng Wen, Xiaodan Liang, Dan Xu, Wei Zhang, Zhenguo Li, Chunjing Xu, and Hang Xu. DetCLIP: Dictionary-Enriched Visual-Concept Paralleled Pre-training for Open-world Detection. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.
- [61] Junbo Yin, Jianbing Shen, Runnan Chen, Wei Li, Ruigang Yang, Pascal Frossard, and Wenguan Wang. IS-Fusion: Instance-Scene Collaborative Fusion for Multimodal 3D Object Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 14905–14915. IEEE, 2024.
- [62] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-Based 3D Object Detection and Tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*,

- CVPR 2021, virtual, June 19-25, 2021*, pages 11784–11793. Computer Vision Foundation / IEEE, 2021.
- [63] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Open-Vocabulary DETR with Conditional Matching. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part IX*, volume 13669 of *Lecture Notes in Computer Science*, pages 106–122. Springer, 2022.
 - [64] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-Vocabulary Object Detection Using Captions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 14393–14402. Computer Vision Foundation / IEEE, 2021.
 - [65] Dongmei Zhang, Chang Li, Renrui Zhang, Shenghao Xie, Wei Xue, Xiaodong Xie, and Shanghang Zhang. FM-OV3D: Foundation Model-Based Cross-Modal Knowledge Blending for Open-Vocabulary 3D Detection. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan, editors, *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pages 16723–16731. AAAI Press, 2024.
 - [66] Hu Zhang, Jianhua Xu, Tao Tang, Haiyang Sun, Xin Yu, Zi Huang, and Kaicheng Yu. OpenSight: A Simple Open-Vocabulary Framework for LiDAR-Based Object Detection. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXXIV*, volume 15142 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2024.
 - [67] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021.
 - [68] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting Twenty-Thousand Classes Using Image-Level Supervision. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner,

editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part IX*, volume 13669 of *Lecture Notes in Computer Science*, pages 350–368. Springer, 2022.

- [69] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4490–4499. Computer Vision Foundation / IEEE Computer Society, 2018.
- [70] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced Grouping and Sampling for Point Cloud 3D Object Detection. *CoRR*, abs/1908.09492, 2019. arXiv: 1908.09492.

APPENDICES

Appendix A

Additional Implementation Details

A.1 Pseudocode for Cross-Modal Association

The goal of cross-modal association is to match each 2D novel object proposal from the image domain with one or more candidate 3D object clusters obtained from LiDAR. As shown in Algorithm 1, each 2D proposal is first projected into a 3D frustum using intrinsic and extrinsic calibration parameters. The center ray of the frustum is computed, and the algorithm identifies nearby 3D clusters whose points lie within a distance threshold τ_{match} from this ray. All matched pairs are stored as potential cross-modal associations for subsequent processing.

A.2 Pseudocode for Cross-Modal Target Preparation

Once candidate (2D proposal, 3D cluster) pairs are obtained, cross-modal target preparation filters associations to retain those suitable for training novel object discovery. As shown in Algorithm 2, the algorithm fits a 3D bounding box to each cluster and applies filters such as occlusion, resolution, and multi-view alignment. A semantic embedding from the 2D proposal, extracted via a vision-language model (e.g., CLIP), is used to assign a high-level class label. Each resulting target includes the 3D box, image embedding, class, and a boolean for alignment quality. To resolve ambiguities when a 2D proposal matches multiple clusters, the best-fitting candidate is selected. Duplicate proposals from overlapping views are removed via 3D NMS. Finalized targets update the novel object memory bank.

Algorithm 1 Cross-Modal Association

```
def cross_modal_association(3D_object_clusters,
                           2D_novel_object_proposals,
                           calibrations):
    """
    Input:
    - 3D_object_clusters: Set of 3D object clusters
    - 2D_novel_object_proposals: Set of 2D novel
                                object proposals
    - calibrations: Intrinsic and extrinsic
                    calibration parameters

    Output:
    - cross_modal_proposals:
      List of (2D proposal, 3D cluster) pairs
    """
    cross_modal_proposals = []

    for proposal in 2D_novel_object_proposals:
        frustum = get_frustum(proposal, calibration, d_min, d_max)
        frustum_center_ray = get_frustum_center_ray(frustum)
        matched_clusters = get_clusters_near_ray(
            3D_object_clusters,
            frustum_center_ray,
            tau_match
        )
        for cluster in matched_clusters:
            cross_modal_proposals.append((proposal, cluster))

    return cross_modal_proposals
```

Algorithm 2 Cross-Modal Target Preparation

```
def cross_modal_target_preparation(cross_modal_proposals,
                                   novel_object_bank,
                                   calibrations):
    """
    Input:
    - cross_modal_proposals:
        List of (2D proposal, 3D cluster) pairs
    - novel_object_bank:
        Set of novel objects for training
    - calibrations: Intrinsic and extrinsic
                    calibration parameters
    """
    cross_modal_targets = []
    box_search_costs = []

    for pair in cross_modal_proposals:
        2d_proposal, 3d_object_cluster = pair
        3D_box_params, box_search_cost = adaptive_3D_box_search(
            3d_object_cluster, 2d_proposal, PS0_params
        )

        instance_mask = get_instance_mask(2d_proposal)
        is_not_occluded = occlusion_filter(2d_proposal, instance_mask)
        is_high_res = resolution_filter(2d_proposal)
        is_aligned_mv = multi_view_alignment_filter(
            2d_proposal, 3D_box_params, calibrations
        )
        fit_for_alignment = is_not_occluded & is_high_res & is_aligned_mv

        2D_image_embed = CLIP(2d_proposal)
        high_level_novel_class = classify(
            2D_image_embed, set_of_novel_classes
        )

        novel_object_target = (
            3D_box_params,
            2D_image_embed,
            high_level_novel_class,
            fit_for_alignment
        )

        cross_modal_targets.append(novel_object_target)
        box_search_costs.append(box_search_cost)

    cross_modal_targets = resolve_CMA_conflicts(
        cross_modal_targets, box_search_costs
    )
    cross_modal_targets = NMS(cross_modal_targets, box_search_costs)
    update(novel_object_bank, cross_modal_targets)
    return
```
