

OpenSight: A Simple Open-Vocabulary Framework for LiDAR-Based Object Detection

Hu Zhang¹, Jianhua Xu⁴, Tao Tang⁵, Haiyang Sun⁶,
Xin Yu^{2*}, Zi Huang^{2*}, and Kaicheng Yu³

¹ CSIRO DATA61, Australia

² The University of Queensland, Australia

³ Westlake University

⁴ DAMO Academy, Alibaba, China

⁵ Shenzhen Campus of Sun Yat-sen University

⁶ LiAuto Inc.

Hu1.Zhang@csiro.au, xin.yu@uq.edu.au, kyu@westlake.edu.cn

<https://github.com/huzhangcs/OpenSight>

Abstract. Traditional LiDAR-based object detection research primarily focuses on closed-set scenarios, which falls short in complex real-world applications. Directly transferring existing 2D open-vocabulary models with some known LiDAR classes for open-vocabulary ability, however, tends to suffer from over-fitting problems: The obtained model will detect the known objects, even presented with a novel category. In this paper, we propose OpenSight, a more advanced 2D-3D modeling framework for LiDAR-based open-vocabulary detection. OpenSight utilizes 2D-3D geometric priors for the initial discernment and localization of generic objects, followed by a more specific semantic interpretation of the detected objects. The process begins by generating 2D boxes for generic objects from the accompanying camera images of LiDAR. These 2D boxes, together with LiDAR points, are then lifted back into the LiDAR space to estimate corresponding 3D boxes. For better generic object perception, our framework integrates both temporal and spatial-aware constraints. Temporal awareness correlates the predicted 3D boxes across consecutive timestamps, recalibrating the missed or inaccurate boxes. The spatial awareness randomly places some “precisely” estimated 3D boxes at varying distances, increasing the visibility of generic objects. To interpret the specific semantics of detected objects, we develop a cross-modal alignment and fusion module to first align 3D features with 2D image embeddings and then fuse the aligned 3D-2D features for semantic decoding. Our experiments indicate that our method establishes state-of-the-art open-vocabulary performance on widely used 3D detection benchmarks and effectively identifies objects for new categories of interest.

Keywords: OpenSight · Open-vocabulary · 3D object detection

* Corresponding author.

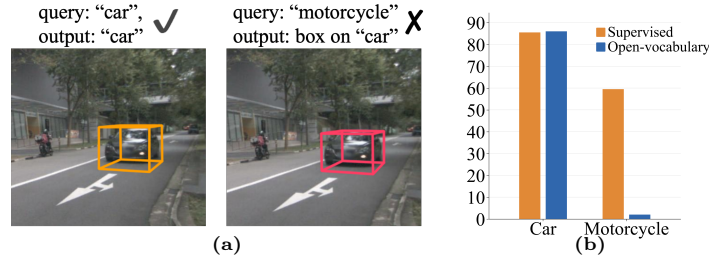


Fig. 1: (a) A model trained on known class “car” correctly detects the target object (yellow) with query “car”. However, with a new class query “motorcycle”, the model incorrectly detects the object (red). (b) The average precision of known “car” and novel “motorcycle”. The degraded performance in “motorcycle” (blue) indicates the limited generalization from the known classes.

1 Introduction

The growing prevalence of autonomous driving has significantly increased research interest in various 3D perception tasks [13,17,41,55], among which LiDAR-based object detection stands as a crucial component. Despite the advancements in this field, most existing research focuses on closed-set settings [19,25,36,48,64], where the obtained models are limited to predicting a set of predefined categories. This constraint makes these models insufficient for handling complex real-world scenarios, where diverse and previously unseen object categories may be encountered. Consequently, there is a growing demand to develop models capable of detecting novel categories of interest.

Considering the limited availability of annotated categories in existing LiDAR-based datasets [4,15,43], adapting established large-scale visual-language models (LVLs), e.g., CLIP [37] for open-vocabulary LiDAR detection appears promising. One paradigm is to co-embed known class object features with the image/text embeddings from LVLs, expecting alignment generalization to the novel classes. While effective in 2D image-based open-vocabulary detection, this paradigm exhibits significant overfitting in LiDAR applications, as models persist in detecting objects in known classes even when faced with a new category.

We consider some known classes, *e.g.*, “car”, and follow the paradigm above to illustrate this phenomenon. When given a novel category “motorcycle”, the obtained model erroneously detects the known object “car”, as depicted in Figure 1 (a). An analysis of average precision (AP) further reveals the diminished efficacy in novel classes, as shown in Figure 1 (b). These results underscore the limitation of such a paradigm, emphasizing the necessity for a tailored approach for LiDAR-based open-vocabulary problems.

In this paper, we present “OpenSight”, a more advanced 2D-3D modeling approach for LiDAR-based open-vocabulary detection. OpenSight primarily relies on 2D and 3D geometry priors to first acquire the concept and localization of generic objects. Then, it interprets the semantics of detected objects by co-embedding LiDAR features with 2D LVLs. This framework addresses all poten-

tially novel objects explicitly and circumvents the reliance on predefined known classes, thereby avoiding the overfitting problem.

Specifically, we consider both temporal-aware and spatial-aware constraints to acquire the concept of generic objects and their localization. Initially, leveraging the 2D detection model and the accompanying RGB images of LiDAR, we extract 2D geometric priors for generic objects, that is, 2D bounding boxes across successive frames. These boxes are then lifted back into 3D space with LiDAR points, resulting in a collection of 3D frustums. Point clustering is then performed in each frustum to estimate the 3D bounding boxes of objects. Temporally, we correlate predicted 3D bounding boxes across frames and project the predicted 3D boxes from previous or subsequent frames onto the current frame. The missed or inaccurate objects in the current frame are then identified through overlapping comparison. This constraint boosts the continuity and consistency of generic objects over time. Spatially, we first employ large language models (LLMs) to derive size priors for general objects and then maintain 3D boxes that closely align with these size priors. The maintained boxes are randomly selected and placed at varying distances within the current frame, with the LiDAR points inside these boxes being sampled by their distance. This spatial awareness notably increases the visibility of objects, particularly those located further from the sensor. By integrating the temporal and spatial constraints, the ability to recognize and localize generic objects is significantly augmented.

To decode the specific semantics of detected boxes in LiDAR space, we correlate objects across both LiDAR and image domains. Initially, by back-projecting 3D LiDAR bounding boxes onto the image plane, we identify the most closely matching 2D boxes from the Grounding DINO [27]. We then aim to minimize the feature discrepancy between the 3D LiDAR objects and the 2D matching box features. Subsequently, we fuse the optimized 3D-2D features and align the unified features with textual category embeddings from Grounding DINO to decode semantics. This dual 2D-3D feature approach leverages salient patterns in both 2D images and 3D geometry, significantly enhancing the robustness of semantic decoding. Additionally, we integrate size priors from LLMs before, recalibrating predictions that diverge from the average box size, thus enhancing the accuracy of our predictive model.

In the experiments, we first establish a variety of baselines for comparison. Then, we test our method on the widely-used 3D detection benchmarks, demonstrating that our approach achieves state-of-the-art open-vocabulary performance and effectively identifies objects for new categories of interest.

2 Related Works

Closed-set 3D Object Detection. Extensive research has been proposed on LiDAR-based 3D object detection in autonomous driving. One group of works performs on LiDAR-only data [39, 63], either by extracting features from raw point clouds to predict 3D bounding boxes [23, 34–36, 40, 49] or converting the irregular LiDAR point clouds into a regular feature space [14, 19, 44, 46, 48, 51, 64].

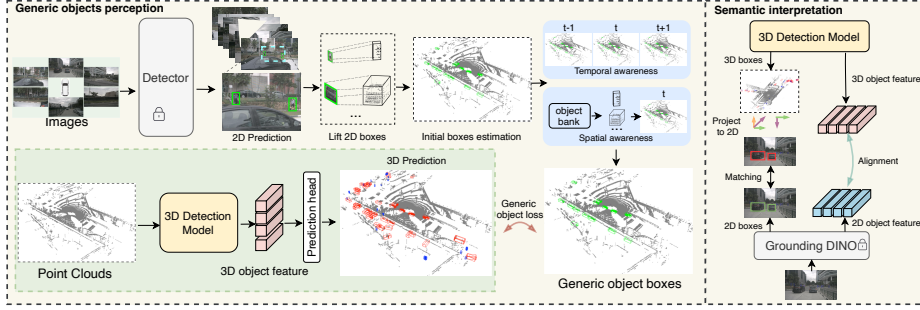


Fig. 2: An overview of our OpenSight framework. In generic object perception, 2D bounding boxes are initially lifted back into 3D space (initial boxes estimation). This is followed by integrated temporal and spatial awareness in the estimated 3D boxes to retrieve the missed objects or increase their visibility. Subsequently, we utilize these enhanced estimates to compute the generic object loss and optimize the 3D detection model. In specific semantic interpretation, we project the generated 3D boxes of the optimized 3D model into the image plane to get projected 2D boxes. These projections are then matched with the nearest 2D boxes identified via Grounding DINO, facilitating the alignment of 3D LiDAR object features with corresponding 2D image features.

Another group focuses on both LiDAR and camera modalities. Some works emphasize point-wise fusion [18, 42, 45, 52], projecting raw LiDAR points to retrieve corresponding image features and concatenating them together. In contrast, others concentrate on feature-level fusion [22, 24, 53], extracting features or proposals based on LiDAR points and querying the associated image features. This line of methods has been proven to be more efficient. For example, TransFusion [3] utilizes proposals generated by LiDAR features to query the image feature for further refinement. BEVFusion [25, 28] projects the image features into BEV space by using lift-splat-shoot (LSS) and concatenates them with the LiDAR features. More recently, LoGoNet [21] constructs Local-to-Global fusion network to perform LiDAR-camera fusion at both local and global levels. However, all these approaches target the closed-set setting, restricting the range of detected object categories.

Open-vocabulary 2D Object Detection. The advent of large-scale pre-trained models, which exhibit good text-vision alignment and zero-shot transferability, has spurred researchers to investigate the open-vocabulary problem in object detection tasks. A variety of approaches have been proposed with 2D RGB images: OVR-CNN [57] initially pre-trains a Faster R-CNN detector [38] by using image-caption pairs and text extractor, e.g., BERT [12], then fine-tunes it on downstream detection datasets. ViLD [16] distills the knowledge from a CLIP-based [37] teacher to a student detector for generalized text and image embedding alignment. Detic [62] harnesses extensive image classification data to train the classifier within a detector while localizing objects. Similar to OVR-CNN, OWL [31] first pre-trains a transformer encoder by contrastively aligning image-text pairs, then transfers it to detection tasks by attaching light-weighted object classification and localization heads. Following the DETR-based

structure, OV-DETR [56] employs CLIP-encoded text and image embeddings as queries for obtaining category-specific boxes. In contrast to the aforementioned methods, some alternatives have emerged to directly learn open-vocabulary capabilities on detection data. For example, GLIP [20] unifies the object detection and phrase grounding tasks, allowing the combined use of both detection and grounding datasets for aligning phrase semantics and regional images. Det-CLIP [50] extracts the general concepts from multiple data sources and then resorts to a description-enriched concept dictionary for enhancement. Recently, Grounding DINO [27] adopts a similar detection-grounding training strategy as GLIP but proposes a tighter text-vision fusion solution based on transformer-based detector DINO [58].

Zero-shot Learning in 3D Data. Exploration of open-vocabulary 3D object detection has been limited due to the challenges associated with acquiring large-scale point cloud-text pairs and the inherent complexity of 3D scenarios. In the confined indoor scenario, OV-3DET [29] addresses the open-vocabulary detection problem by first localizing 3D objects and then assigning labels for the objects with open-vocabulary 2D models. However, temporal and spatial constraints are overlooked. CoDA [5] proposes to use some base classes to assist in detecting novel classes. As illustrated in Figure 1, such a paradigm in LiDAR data easily overfits the base classes. Recently, UP-VL [32] also considers the open-vocabulary case in autonomous driving, however, the categories involved are rather limited (only 3 classes). Some other works [2, 6, 7, 47] tackle a related but distinct task of differentiating unknown data in testing from known data in training, known as the out-of-distribution (OOD) problem. Similarly, MODEST [54], OYSTER [59], and LiSe [61] study the unsupervised object discovery problem. Meanwhile, some studies [26, 30, 33] attempt to transfer existing 2D large-scale pre-trained models for open-vocabulary scene segmentation, while others [8–11, 60] leverage the open-set ability of pre-trained models for the 3D classification tasks.

Different from the above works, in this paper, we investigate open-vocabulary ability in LiDAR-based 3D object detection in the outdoor scenario, representing the real-world scenario and a much more complex scene.

3 OpenSight

To elaborate on the LiDAR-based open-vocabulary object detection, we first briefly introduce the necessary notations for ease of understanding. Following this, we delve into the details of generic object perception in Section 3.2. Subsequently, we present the process of specific semantic interpretation in Section 3.3. The overall framework of OpenSight is presented in Figure 2.

3.1 Notation and Preliminaries

The LiDAR dataset is denoted as $\mathcal{D} = \{P_i, \{y_k\}_{k=1}^K\}_{i=1}^{|\mathcal{D}|}$, where P_i corresponds to the LiDAR point clouds in i -th frame, basically composed of (x_i, y_i, z_i) coordinates. y_k represents the combination of each 3D bounding box b_k and its

semantic label c_k . In open-vocabulary scenarios, P_i is visible while y_k is not. Generally, in the context of autonomous driving, each LiDAR frame is accompanied by m images captured from varying perspectives. We define the image dataset as $\mathcal{I} = \{(I_{ij})_{j=1}^m\}_{i=1}^{|\mathcal{D}|}$ here. Image I_{ij} in j -th view and LiDAR points P_i could be connected by the intrinsic matrix $\mathbf{K}_j$ and extrinsic matrix $\mathbf{E}_j$. For clarity and brevity in the rest of the paper, we utilize a single LiDAR frame as an example and abbreviate I_{ij} as I_j , respectively. N denotes the number of 3D objects, with f^{3D} and f^{2D} denotes the extracted feature in 3D and 2D.

3.2 Generic Objects Perception

Discerning and localizing generic objects are fundamental in LiDAR-based open-vocabulary detection when ground-truth boxes and labels are absent. The model is expected to perceive the existence of a generic object before assigning specific labels. This section details how we leverage 2D and 3D geometry priors to enable this perception, including initial box estimation, and temporal-spatial awareness.

Initial Box Estimation. In practical applications, each LiDAR frame is typically paired with m images $\{I_j\}_{j=1}^m$, captured from various perspectives. Each image is processed through the pre-trained detection model $\mathcal{I}$, yielding a set of 2D bounding boxes $\{b_{jk}\}_{k=1}^{V_j}$, where V_j represents the number of 2D objects identified in the j -th viewpoint. By integrating the data from all viewpoints, we compile the collective set of detected boxes $\{(b_{jk}^{2D})_{k=1}^{V_j}\}_{j=1}^m$.

With the obtained 2D boxes of generic objects, we try to lift them back in the LiDAR space. Specifically, we project the LiDAR point P_i into each image view by:

$$\hat{P}_{ij} = \mathbf{K}_j \cdot \mathbf{E}_j \cdot P_i, \quad j = 1, \dots, m \quad (1)$$

where $\mathbf{E}_j$ and $\mathbf{K}_j$ are extrinsic and intrinsic matrix, respectively. The points whose projection falls within each of the predicted 2D bounding boxes b_{jk}^{2D} essentially constitute the 3D frustum, a conical region perceivable from a given perspective. We then apply the region growing algorithm [1] to isolate the cluster with the densest LiDAR points. Such an algorithm first randomly selects one point in the frustum and then iteratively finds its adjacent points.

To estimate 3D bounding boxes from LiDAR point clusters, we first determine the extremal points along each axis by calculating the minimum and maximum coordinates. Take the X-axis for example, this is achieved by performing $x_{\min} = \min(\{x_i\}_{i=0}^{|N_c|})$ and $x_{\max} = \max(\{x_i\}_{i=0}^{|N_c|})$, where $|N_c|$ is the number of points in the obtained cluster. These extremal points, identified similarly for the Y and Z axis, define the vertices of the bounding box. From these vertices, we construct the maximal bounding box b_i^{3D} , enclosing all points in the cluster. For the heading of the bounding box, we traverse different angles. In each angle, we can obtain the XYZ boundary. We then shrink the boundary by a scale of 0.95, some points will be out of the obtained bounding boxes. The angle with the most outside points is regarded as the angle of the object. It is based on the assumption that for the ‘‘tightest’’ bounding box, a slight change in the angle will incur a big change in the encompassed points. The 3D boxes for generic

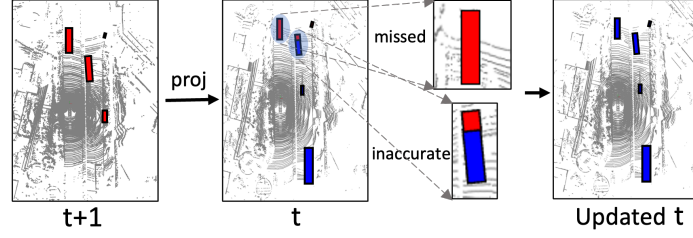


Fig. 3: Temporal awareness: Project boxes from frame $t + 1$ to t . By comparing with projections, we refine missed or inaccurately estimated boxes in frame t .

objects at timestamp t are then finalized as $\mathcal{B}^t = \{b_i^{3D}\}_{i=0}^{N_t}$, by aggregating all boxes from m perspectives.

Temporal Awareness. For the LiDAR point cloud, point density decreases sharply as distance increases, leading to less reliable estimations for distant objects. However, in autonomous driving scenarios, objects that are distant in the current frame may move closer in subsequent frames. To exploit this, we leverage the temporal continuity between frames by correlating 3D bounding box predictions across successive timestamps. We take frames at timestamp t and $t + 1$ as an example. Specifically, given the predicted boxes $\mathcal{B}^t = \{b_i\}_{i=0}^{N_t}$ and $\mathcal{B}^{t+1} = \{b_i\}_{i=0}^{N_{t+1}}$, where N_{t+1} and N_t represent the number of boxes in $t + 1$ and t , respectively, we project each 3D box b_i in $\mathcal{B}^{t+1}$ back to frame t using the transformation:

$$\hat{b}_i = \mathbf{E}_{\text{lidar} \leftarrow \text{ego}_t} \cdot \mathbf{E}_{\text{ego}_t \leftarrow \text{glb}} \cdot \mathbf{E}_{\text{glb} \leftarrow \text{ego}_{t+1}} \cdot \mathbf{E}_{\text{ego}_{t+1} \leftarrow \text{lidar}} \cdot b_i,$$

where $\mathbf{E}_{\text{lidar} \leftarrow \text{ego}_t}$ and $\mathbf{E}_{\text{ego}_t \leftarrow \text{glb}}$ are the extrinsic matrices for the ego to LiDAR and global to ego transformations at timestamp t , and $\mathbf{E}_{\text{glb} \leftarrow \text{ego}_{t+1}}$, $\mathbf{E}_{\text{ego}_{t+1} \leftarrow \text{lidar}}$ denotes ego to global and LiDAR to ego at timestamp $t + 1$, respectively.

With all boxes in $\mathcal{B}^{t+1}$ considered, we obtain their projections as $\hat{\mathcal{B}}^{t+1}$ and construct the overlap matrix $M \in \mathbb{R}^{N_{t+1} \times N_t}$ by calculating their Intersection over Union (IoU) with boxes in $\mathcal{B}^t$. We address two key scenarios: If a box $\hat{b}_i$ has no overlap with boxes in $\mathcal{B}^t$, its counterpart b_i in timestamp $t + 1$ is considered a missed detection in the current frame and is added to the prediction at timestamp t . Conversely, if box $\hat{b}_i$ overlaps with boxes in $\mathcal{B}^t$ and its distance exceeds a certain threshold, we perform Non-Maximum Suppression (NMS) and retain the union of the two boxes to address potentially incomplete estimations at greater distances (See Figure 3).

This process updates the predicted boxes of the current frame to $\mathcal{B}_{\text{update}}^t = \mathcal{B}^t \cup \mathcal{B}_{\text{temporal}}^t$, which combines the original and temporally adjusted boxes. In practice, for the current frame t , we consider both its subsequent frame at timestamp $t + 1$ and its precedent frame at timestamp $t - 1$. That means two adjacent frames are considered. Through such a constraint, we are likely to boost the continuity and consistency of generic objects over time.

Spatial Awareness. Considering the sparse object distribution in 3D LiDAR scans, where vast areas of the point cloud are often devoid of meaningful objects,

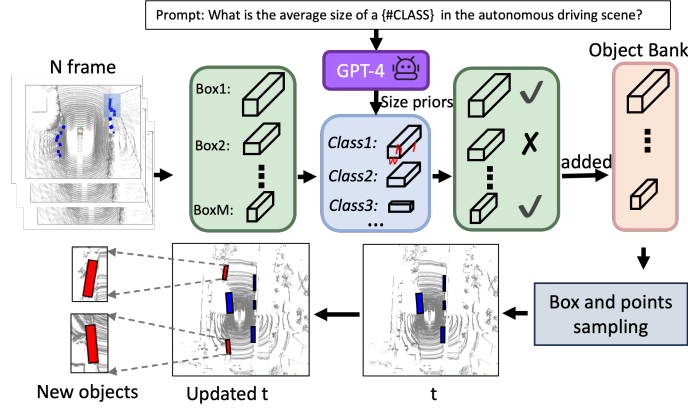


Fig. 4: Spatial awareness: Size priors are first generated from GPT-4 to select “precisely” estimated boxes and build the object bank. Randomly select objects and points from the objects bank and place them in the current frame t .

we also focus on the systematic integration of “precisely” estimated objects in each frame to enhance the detectability and representation of generic objects. To obtain these precise boxes from $\mathcal{B}^t$ in the absence of any ground truth, we utilize large language models (LLMs), such as GPT-4, for extracting useful size priors of objects. GPT-4, when prompted appropriately, can provide valuable insights into the typical sizes of objects within a specific class, *e.g.*, the width, length, and height of a class. For instance, prompts like “What is the average size of a {#CLASS} in the autonomous driving scene?” enable us to collect size statistics for common classes (Details in the Appendix). Moreover, leveraging the ability of GPT-4 to process images, we also input images of the relevant objects to validate these size priors (See Figure 4).

Utilizing the extracted size priors, we selectively retain 3D boxes in $\mathcal{B}^t$ that closely align with these priors. A box is reserved if its dimensions w , l , and h fall within the specified range:

$$\begin{aligned} w &\in [(1 - \tau)w_{prior}, (1 + \tau)w_{prior}], \\ l &\in [(1 - \tau)l_{prior}, (1 + \tau)l_{prior}], \\ h &\in [(1 - \tau)h_{prior}, (1 + \tau)h_{prior}], \end{aligned} \quad (2)$$

where τ represents the error margin. w_{prior} , l_{prior} and h_{prior} are from LLMs-derived priors. This process results in an object bank $O = \{(b_i^{3D}, P_i)\}_{i=0}^{N_o}$, where b_i^{3D} is the reserved box, P_i comprises the LiDAR points within each box and N_o is the total number of reserved boxes.

For the current frame, we randomly select boxes from bank O and integrate them at varying distances from the ego vehicle, ensuring diverse spatial distribution. The LiDAR points within each box are sampled according to their distance, with the sampling ratio defined as:

$$r_{sampling} = \min(1, \frac{d - d_{new}}{d - d_{ori}}), \quad (3)$$

where d denotes the total range in the point cloud, d_{ori} is the original distance of the object, d_{new} is the distance post-placement. The selected boxes are denoted as $\mathcal{B}_{\text{spatial}}^t$ and the collective set of boxes in the current frame is represented as $\mathcal{B}_{\text{final}}^t = \mathcal{B}^t \cup \mathcal{B}_{\text{temporal}}^t \cup \mathcal{B}_{\text{spatial}}^t$. This strategy aims to substantially enhance the visibility of generic objects in LiDAR data and the perception ability of the obtained model.

Loss for Generic Object Perception. With the obtained 3D boxes $\mathcal{B}_{\text{final}}^t = \{b_k^{3D}\}_{k=1}^{V_{\text{final}}}$, we formulate the loss for generic object concept acquisition and localization. Utilizing a DETR-like structure, we extract features from LiDAR points and apply bipartite matching to determine the optimal permutation $\hat{\sigma}$, the loss is thus expressed as:

$$\mathcal{L} = \sum_{k=1}^N [\mathcal{L}_{\text{focal}}(s_{\hat{\sigma}_i(k)}) + \mathcal{L}_{L_1}(b_{\hat{\sigma}_i(k)}, b_k^{3D})], \quad (4)$$

where $s_{\hat{\sigma}_i(k)}$ denotes whether it is an object or not, $\hat{\sigma}_i(k)$ is the element in $\hat{\sigma}$. $\mathcal{L}_{\text{focal}}$ constrains the level of ‘‘objectness’’ of generated 3D boxes, $\mathcal{L}_{\text{box}}$ considers the constraint on boxes. With this composite loss optimized, we obtain a model with the ability to perceive generic objects.

3.3 Specific Semantics Interpretation

To interpret the specific semantics of detected generic objects in LiDAR, we correlate objects across both LiDAR and image domains. We apply the model after generic object perception to a given LiDAR frame, producing boxes $\{b_i^{3D}\}_{i=0}^N$ and their corresponding features $\{f_i^{3D}\}_{i=0}^N$. Instead of using the image-level LVLs, *e.g.*, CLIP, we opt for object-level LVLs, such as Grounding DINO [27], which provides 2D boxes $\{b_{i=0}^{2D}\}_{i=0}^{N_{2D}}$, 2D object features $\{f_i^{2D}\}_{i=0}^{N_{2D}}$ and text embeddings $\{t_i\}_{i=0}^{N_{2D}}$ simultaneously.

For each detected 3D box b_i^{3D} , we project it to each view image with the corresponding intrinsic matrix K_j and extrinsic matrix E_j , formulated as:

$$\hat{b}_{ij}^{2D} = \mathbf{K}_j \cdot \mathbf{E}_j \cdot b_i^{3D}, \quad j = 1, \dots, m \quad (5)$$

The $\hat{b}_{ij}^{2D}$ is further filtered and only those lying in the image are reserved. For the reserved box $\hat{b}_{ij}^{2D}$, we compute the IoU with the 2D boxes $\{b_i^{2D}\}_{i=0}^{N_{2D}}$ from Grounding DINO and find the most close one b_j^{2D} . The corresponding text embeddings t_j^{2D} feature is retrieved. We then try to minimize the discrepancy between f_i^{3D} and all 2D features that have the same text embeddings t_j^{2D} . Mathematically, it is expressed as:

$$\ell_{\text{align}} = -\frac{1}{N} \sum_{i=1}^N \log \frac{\sum_{j=0}^{N_i} e^{f_i^{3D} \cdot f_j^{2D} / \delta}}{\sum_{k=0}^{N_{2D}} e^{f_i^{3D} \cdot f_k^{2D} / \delta}}, \quad (6)$$

where N_i denotes all 2D features that have the same text embeddings t_j^{2D} for b_i^{3D} . Once the object-level alignment is finished, we can directly perform open-vocabulary LiDAR detection with the LiDAR input.

2D-3D Integration. Considering that paired RGB images with LiDAR are always available in autonomous driving, we thus can perform 2D-3D feature integration. For each detected object, we have optimized 3D features f_i^{3D} and identified the best matched 2D boxes feature f_i^{2D} , the similarity between 3D-2D features with the text embeddings in $\{t_i\}_{i=0}^{N^{2D}}$ are computed as follows:

$$\begin{aligned} s_j^{3D} &= \cos(f_i^{3D}, t_j), \\ s_j^{2D} &= \cos(f_i^{2D}, t_j), \quad j = 0, \dots, N^{2D} \end{aligned} \quad (7)$$

We obtain the maximum value for 2D and 3D feature by $s^{3D} = \max_j \{s_j^{3D}\}_{j=0}^{N^{2D}}$, $s^{2D} = \max_j \{s_j^{2D}\}_{j=0}^{N^{2D}}$, along with their semantic labels c^{3D} and c^{2D} . The final label for object i is determined by comparing s_i^{2D} and s_i^{3D} , selecting the label associated with the larger value.

Large Language Model (LLMs) Post-processing. The size priors from LLMs can be used to assist in recalibrating predictions. By matching the 3D IoU of predicted boxes with these size priors, we can determine the most matching prior and the corresponding semantic label. We then incorporate the predictions from size prior with results from 2D-3D integration by selecting label with the highest similarity.

3.4 Inference

we first input the LiDAR point cloud to the generic object detector and extract 3D object features. We then align these features with text embeddings from Grounding DINO [27] to determine semantic labels. Additionally, when RGB images are available, we further improve the alignment ability by integrating 3D and 2D features or use the LLMs prior (size priors from GPT-4) to calibrate the predicted labels, thereby improving the accuracy of obtained semantic labels.

4 Experiments

4.1 Experimental settings

Dataset. We conduct the open-vocabulary experiments on nuScenes [4], a large-scale, outdoor dataset featuring multimodal data modalities. Each frame in the dataset contains data from one 32-beam LiDAR and six cameras, offering a comprehensive view of the surroundings. The dataset comprises 1,000 scenes in total, with up to 1.4 million annotated 3D bounding boxes spanning 10 classes. We reserve all the classes as novel to evaluate the effectiveness of our method.

Evaluation Metrics. For quantitative evaluation, we initially employ Average Precision (AP) on each class and mean Average Precision (mAP) across all novel classes. Considering that LiDAR point density varies with the distance, we introduce distance-based metrics: mAP-near, mAP-midrange, and mAP-far. Similarly, acknowledging the size disparities among objects in different classes (e.g., ‘bus’ vs. ‘traffic_cone’), we categorize classes into ‘large’, ‘medium’, and ‘small’ and propose mAP-large, mAP-medium, and mAP-small. More details are in the Appendix.

Table 1: Results on NuScenes dataset. All classes are reserved as novel classes. AP of each class and mAP across all classes are shown. “Car” in CoDA is the known class and is fully supervised (‘Cone’: ‘Traffic_cone’, ‘C.V’: ‘Construction_vehicle’, ‘Motor.’: ‘Motorcycle’, ‘Ped.’: ‘Pedestrian’).

Method	mAP	NDS	Car	Truck	C.V.	Bus	Trailer	Barrier	Bicycle	Cone	Motor.	Ped.
Supervised (box+label)	57.3	61.7	86.0	38.8	18.3	63.3	34.2	70.5	46.7	70.2	59.5	85.2
Supervised (box only)	34.2	39.0	53.2	20.1	7.4	20.5	0.7	50.1	37.8	52.6	39.1	60.8
Detic-3D [62]	5.8	12.7	15.1	2.9	2.1	4.2	0.0	0.1	6.2	0.8	6.1	21.1
OWL-ViT-3D [31]	5.4	12.4	14.2	2.5	1.8	4.0	0.0	0.1	5.2	1.1	5.5	19.4
OV-PointCLIP [60]	0.1	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.2	0.0	0.1
OV-PointCLIPv2 [65]	2.8	10.9	0.0	2.7	0.0	0.0	0.0	22.0	0.0	0.2	0.0	3.2
CLIP-3D [37]	7.2	14.2	9.6	2.8	2.1	2.6	0.0	0.1	11.5	14.4	9.8	18.9
OV-3DET [29]	5.7	12.0	16.1	2.9	0.3	1.3	0.0	0.3	1.8	11.0	2.0	21.1
SeCondPoint [26]	8.2	15.0	15.1	2.3	2.0	2.0	0.0	0.0	10.9	14.9	15.2	19.3
3DGenZ [30]	7.5	14.9	14.6	3.5	2.3	3.2	0.0	0.0	10.5	14.2	8.7	17.9
Openscene [33]	8.8	15.4	17.2	3.9	2.2	3.0	0.1	0.2	12.8	14.0	10.8	23.4
UP-VL [32]	11.3	17.2	19.4	7.9	4.3	3.1	0.4	0.5	14.1	15.6	16.8	30.8
CoDA [5]	10.3	16.1	85.5	1.9	0.0	2.0	0.0	0.0	3.9	0.0	5.9	3.5
OpenSight (Ours)	22.9	23.3	25.3	11.6	8.7	5.1	0.8	42.2	26.1	25.6	30.4	52.5

Table 2: Performance *w.r.t* mAP-near, mAP-medium, mAP-far.

Method	mAP-near	mAP-midrange	mAP-far
Detic-3D [62]	18.5	9.3	5.8
OWL-ViT-3D [31]	17.3	8.2	5.4
OV-PointCLIP [60]	0.2	0.1	0.1
OV-PointCLIPv2 [65]	4.5	3.6	2.8
CLIP-3D [37]	18.8	11.2	7.2
OV-3DET [29]	16.1	9.9	5.7
CoDA [5]	18.2	16.6	10.3
OpenSight (Ours)	28.0	26.5	22.9

4.2 Baselines

As there are no direct baselines addressing open-vocabulary LiDAR-based 3D detection, we primarily construct the baselines based on well-known works [5, 29, 37, 62, 65] and compare with OpenSight. Specifically, they are:

Fully-supervised training. Initially, we perform fully supervised training across all categories, establishing an upper bound for the open-vocabulary detection performance. We consider using both ground-truth boxes and labels (‘Supervised (box+label)’ or just using ground-truth boxes (‘Supervised (box only)’).

2D open-vocabulary methods. Detic [62] is a pioneer work in the realm of 2D open-vocabulary object detection tasks and can detect 21K classes. It is thus desirable for our study here. OWL-ViT [31] decodes the object labels and boxes directly from the pre-trained features and shows good robustness in real-world applications. With the 2D detection results of either Detic or OWL-ViT, we adopt the same procedure in Section 3.2 to obtain the corresponding 3D bounding boxes in LiDAR space. These two baselines are denoted as Detic-3D and OWL-ViT-3D.

3D open-vocabulary methods. We then consider the latest open-vocabulary point cloud classification methods and a variety of indoor open-vocabulary works to evaluate their performance in our tasks here. Specifically, we utilize a fully-supervised method to generate pseudo boxes and employ PointCLIP [60], PointCLIPv2 [65], denoted as “OV-PointCLIP”, “OV-PointCLIPv2”. We also combine

Table 3: Performance *w.r.t* mAP-large, mAP-medium, mAP-small.

Method	mAP-large	mAP-medium	mAP-small
Detic-3D [62]	5.3	4.1	11.0
OWL-ViT-3D [31]	4.9	3.7	10.3
OV-PointCLIP [60]	0.0	0.3	0.2
OV-PointCLIPv2 [65]	0.6	7.3	1.6
CLIP-3D [37]	3.4	7.1	16.7
OV-3DET [29]	4.1	1.4	16.0
CoDA [5]	17.9	4.9	1.8
OpenSight (Ours)	11.5	32.9	39.0

Table 4: Results with temporal and spatial awareness *w.r.t* mAP- $\{\text{near, medium, far}\}$.

#	Temporal	Spatial	mAP-near	mAP-midrange	mAP-far
(1)			27.1	24.8	22.1
(2)	✓		27.4	25.9	22.5
(3)		✓	27.5	26.0	22.7
(4)	✓	✓	28.0	26.5	22.9

our predicted boxes and CLIP [37] for this task (“CLIP-3D”). Meanwhile, we adapt the latest works OV-3DET [29] and CoDA [5] in indoor settings and UP-VL [32] to our task here. We also consider the segmentation-based methods, *e.g.*, SeCondPoint [26], 3DGenZ [30], and OpenScene [33], for comparison.

4.3 Main results

In this section, we discuss the performance of our model on novel classes, as summarized in Table 1. Despite the lack of 3D ground truth supervision, our framework demonstrates strong overall open-vocabulary ability across classes. Notably, it achieves exceptional performance in some classes, for example, 52.5 AP in ‘pedestrian’ and 42.2 AP in ‘Barrier’. Unfortunately, we also notice the limited performance in ‘Trailer’, with 0.8 AP. To analyze this, we compare with ‘Supervised (box only)’, where groundtruth boxes are used and only 0.7 AP is achieved in ‘Trailer’, we thus attribute the reason to label alignment with Grounding DINO rather than box inaccuracy. Grounding DINO offers limited help in fine-grained classes. Our model also shows superior performance when compared with the baselines. Specifically, we observe that PointCLIP is very limited in detecting novel objects and CLIP-3D achieves the best overall performance with our detected boxes. CoDA, while effective in known classes like ‘car’, falls short in novel class adaptation, averaging only 1.9% accuracy across novel categories.

Our more detailed evaluations, using distance-based metrics and size-based metrics, are presented in Tables 2 and 3, respectively. The evaluation in Table 2 reveals a trend where detection accuracy is higher for objects closer to the sensor. This is attributed to their denser point clouds and more defined geometric features. Conversely, detection accuracy decreases with increasing distance, likely due to sparser data and less distinct object outlines. Regarding size-based performance (Table 3), our model shows superior results for medium and small-sized objects. Large objects present lower detection rates, mainly due to their

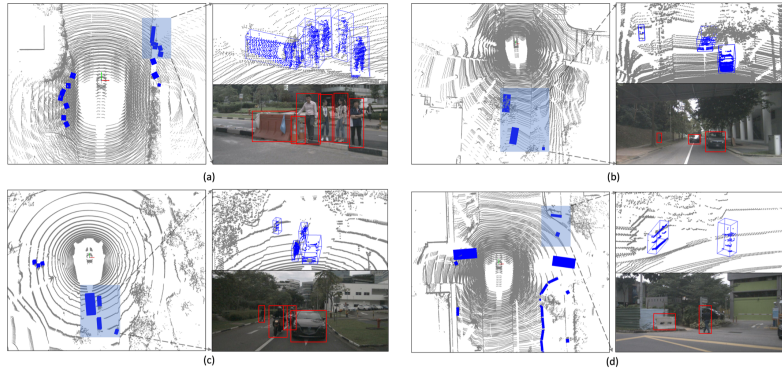


Fig. 5: Visualization of detected novel classes in different LiDAR frames. (a) Detected ‘barrier’, ‘traffic_cone’, and ‘pedestrian’; (b) Detected ‘car’ and ‘pedestrian’; (c) Detected ‘car’ and ‘motorcycle’; (d) Detected ‘barrier’ and ‘pedestrian’.

Table 5: Results with direct alignment (‘D.A.’), 3D-2D feature integration (‘I.G.’), and LLMs priors.

#	D.A	I.G.	Priors	mAP-large	mAP-medium	mAP-small
(1)	✓			9.4	31.8	38.1
(2)	✓	✓		10.9	32.7	38.8
(3)	✓	✓	✓	11.5	32.9	39.0

incomplete LiDAR scans. Additionally, the fine-grained categorization of large vehicles, such as ‘Construction_vehicle’, complicates their accurate detection.

4.4 Qualitative results on novel classes

As shown in Figure 5, our framework demonstrates a strong ability to detect novel objects by inputting the interested categories. We visualize the results from both BEV and a specific perspective. For example, our method successfully identifies the ‘pedestrian’, ‘barrier’, and ‘traffic_cone’ in example (a), and the ‘car’ and ‘motorcycle’ in (c). These results also indicate that our method exhibits superior open-vocabulary detection ability across different scenes.

4.5 Ablation

In this part, we present a comprehensive analysis of OpenSight and examine the effectiveness of each component.

Components in Generic Object Perception. We decouple the components in generic object perception and record the results in Table 4. The overall improvement may not be so obvious, however, we improve more for mAP-medium and mAP-far with these techniques. For example, temporal improves 1.1 AP for mAP-medium while 0.3 AP for mAP-near. Both modules interact positively with each other and further improve the results when jointly introduced.

Components in Semantic Interpretation. We analyze the modules of direct alignment, 3D-2D feature integration, and LLMs priors in semantic interpretation and record the results in Table 5. The performance is consistently improved

Table 6: Results of image-level and object-level models.

mAP	-near	-midrange	-far	-large	-medium	-small
CLIP	19.0	11.9	8.2	3.9	8.2	18.4
Grounding DINO	28.0	26.5	22.9	11.5	32.9	39.0

Table 7: AP and AR of some classes under different threshold.

#CLASS	AP-0.5	AP-1	AP-2	AR-0.5	AR-1	AR-2
Car	4.7	18.3	45.9	40.0	69.0	84.4
Truck	0.2	3.0	13.3	13.2	30.0	51.1
Pedestrian	38.8	51.6	57.1	58.4	69.3	74.2

with integration and priors introduced. Compared with LLMs priors, 3D-2D integration brings more benefits, for example, 0.9 *vs.* 0.2 in mAP-medium.

Image-level *vs.* Object-level. We also investigate the image-level CLIP [37] model in semantic interpretation. The results are recorded in Table 6. Compared with our adopted object-level Grounding DINO, we notice that object-level alignment shows much better results from either a distance perspective or a size perspective. For example, 28.0 AP of near objects is achieved in object-level alignment, 9 points higher than image-level alignment. We analyze that CLIP requires more accurate bounding box prediction, in comparison with object-level alignment which only requires a nearest box.

Importance of Bounding Boxes Existence. Existing open-vocabulary methods focus more on the alignment between 3D features and 2D features. However, in LiDAR-based object detection, we emphasize that accurate bounding boxes are essential before alignment. We select some classes and investigate both average precision (AP) and average recall (AR) under different thresholds. The results are recorded in Table 7. Compared with AP, we observe a higher recall value in different threshold, indicating that although we can recall the correct objects, more objects with incorrect sizes are also recalled. Such results motivate that if we can obtain a more precise estimation of objects, the results will be improved further.

5 Conclusion

We present OpenSight, a simple and versatile open-vocabulary framework for LiDAR-Based object detection. In the absence of ground-truth boxes and semantic labels, OpenSight first learns to identify and localize a wide range of generic objects. To boost the ability of generic object perception, we incorporate temporal and spatial awareness techniques that recalibrate and improve the visibility of estimated 3D boxes. We also align LiDAR object features with the 2D features from the Grounding DINO for semantic interpretation. The extensive experiments demonstrate the effectiveness and generalization ability of the proposed framework, paving the way for further exploration in LiDAR-based open-vocabulary challenges.

Acknowledgement

This research is funded in part by ARC-Discovery grant (DP220100800 to XY) and ARC-DECRA grant (DE230100477 to XY). We thank all anonymous reviewers and ACs for their constructive suggestions.

References

1. Adams, R., Bischof, L.: Seeded region growing. *IEEE Transactions on pattern analysis and machine intelligence* **16**(6), 641–647 (1994)
2. Alliegro, A., Cappio Borlino, F., Tommasi, T.: 3dos: Towards 3d open set learning-benchmarking and understanding semantic novelty detection on point clouds. *Advances in Neural Information Processing Systems* **35**, 21228–21240 (2022)
3. Bai, X., Hu, Z., Zhu, X., Huang, Q., Chen, Y., Fu, H., Tai, C.L.: Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 1090–1099 (2022)
4. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: *CVPR* (2020)
5. Cao, Y., Zeng, Y., Xu, H., Xu, D.: Coda: Collaborative novel box discovery and cross-modal alignment for open-vocabulary 3d object detection. *arXiv preprint arXiv:2310.02960* (2023)
6. Cen, J., Yun, P., Cai, J., Wang, M.Y., Liu, M.: Open-set 3d object detection. In: *2021 International Conference on 3D Vision (3DV)*. pp. 869–878. IEEE (2021)
7. Cen, J., Yun, P., Zhang, S., Cai, J., Luan, D., Tang, M., Liu, M., Yu Wang, M.: Open-world semantic segmentation for lidar point clouds. In: *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVIII*. pp. 318–334. Springer (2022)
8. Cheraghian, A., Rahman, S., Campbell, D., Petersson, L.: Mitigating the hubness problem for zero-shot learning of 3d objects. *arXiv preprint arXiv:1907.06371* (2019)
9. Cheraghian, A., Rahman, S., Campbell, D., Petersson, L.: Transductive zero-shot learning for 3d point cloud classification. In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. pp. 923–933 (2020)
10. Cheraghian, A., Rahman, S., Chowdhury, T.F., Campbell, D., Petersson, L.: Zero-shot learning on 3d point cloud objects and beyond. *International Journal of Computer Vision* **130**(10), 2364–2384 (2022)
11. Cheraghian, A., Rahman, S., Petersson, L.: Zero-shot learning of 3d point cloud objects. In: *2019 16th International Conference on Machine Vision Applications (MVA)*. pp. 1–6. IEEE (2019)
12. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018)
13. Douillard, B., Underwood, J., Kuntz, N., Vlaskine, V., Quadros, A., Morton, P., Frenkel, A.: On the segmentation of 3d lidar point clouds. In: *2011 IEEE International Conference on Robotics and Automation*. pp. 2798–2805. IEEE (2011)
14. Fan, L., Xiong, X., Wang, F., Wang, N., Zhang, Z.: Rangedet: In defense of range view for lidar-based 3d object detection. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 2918–2927 (2021)

15. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
16. Gu, X., Lin, T.Y., Kuo, W., Cui, Y.: Open-vocabulary object detection via vision and language knowledge distillation. In: International Conference on Learning Representations (2022), <https://openreview.net/forum?id=1L3lnMbR4WU>
17. Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M.: Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence* **43**(12), 4338–4364 (2020)
18. Huang, T., Liu, Z., Chen, X., Bai, X.: Epnet: Enhancing point features with image semantics for 3d object detection. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16. pp. 35–52. Springer (2020)
19. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12697–12705 (2019)
20. Li, L.H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.N., et al.: Grounded language-image pre-training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10965–10975 (2022)
21. Li, X., Ma, T., Hou, Y., Shi, B., Yang, Y., Liu, Y., Wu, X., Chen, Q., Li, Y., Qiao, Y., et al.: Logonet: Towards accurate 3d object detection with local-to-global cross-modal fusion. arXiv preprint arXiv:2303.03595 (2023)
22. Li, Y., Yu, A.W., Meng, T., Caine, B., Ngiam, J., Peng, D., Shen, J., Lu, Y., Zhou, D., Le, Q.V., et al.: Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17182–17191 (2022)
23. Li, Z., Wang, F., Wang, N.: Lidar r-cnn: An efficient and universal 3d object detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7546–7555 (2021)
24. Liang, M., Yang, B., Wang, S., Urtasun, R.: Deep continuous fusion for multi-sensor 3d object detection. In: Proceedings of the European conference on computer vision (ECCV). pp. 641–656 (2018)
25. Liang, T., Xie, H., Yu, K., Xia, Z., Lin, Z., Wang, Y., Tang, T., Wang, B., Tang, Z.: Bevfusion: A simple and robust lidar-camera fusion framework. arXiv preprint arXiv:2205.13790 (2022)
26. Liu, B., Deng, S., Dong, Q., Hu, Z.: Language-level semantics conditioned 3d point cloud segmentation. arXiv preprint arXiv:2107.00430 (2021)
27. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023)
28. Liu, Z., Tang, H., Amini, A., Yang, X., Mao, H., Rus, D., Han, S.: Bevfusion: Multi-task multi-sensor fusion with unified bird’s-eye view representation. arXiv preprint arXiv:2205.13542 (2022)
29. Lu, Y., Xu, C., Wei, X., Xie, X., Tomizuka, M., Keutzer, K., Zhang, S.: Open-vocabulary point-cloud object detection without 3d annotation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1190–1199 (2023)
30. Michele, B., Boulch, A., Puy, G., Bucher, M., Marlet, R.: Generative zero-shot learning for semantic segmentation of 3d point clouds. In: 2021 International Conference on 3D Vision (3DV). pp. 992–1002. IEEE (2021)

31. Minderer, M., Gritsenko, A., Stone, A., Neumann, M., Weissenborn, D., Dosovitskiy, A., Mahendran, A., Arnab, A., Dehghani, M., Shen, Z., et al.: Simple open-vocabulary object detection with vision transformers. arXiv preprint arXiv:2205.06230 (2022)
32. Najibi, M., Ji, J., Zhou, Y., Qi, C.R., Yan, X., Ettinger, S., Anguelov, D.: Unsupervised 3d perception with 2d vision-language distillation for autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8602–8612 (2023)
33. Peng, S., Genova, K., Jiang, C., Tagliasacchi, A., Pollefeys, M., Funkhouser, T., et al.: Openscene: 3d scene understanding with open vocabularies. arXiv preprint arXiv:2211.15654 (2022)
34. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum pointnets for 3d object detection from rgb-d data. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 918–927 (2018)
35. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660 (2017)
36. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems* **30** (2017)
37. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
38. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* **28** (2015)
39. Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H.: Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10529–10538 (2020)
40. Shi, S., Wang, X., Li, H.: Pointtrcn: 3d object proposal generation and detection from point cloud. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 770–779 (2019)
41. Simon, M., Amende, K., Kraus, A., Honer, J., Samann, T., Kaulbersch, H., Milz, S., Michael Gross, H.: Complexer-yolo: Real-time 3d object detection and tracking on semantic point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)
42. Sindagi, V.A., Zhou, Y., Tuzel, O.: Mvx-net: Multimodal voxelnet for 3d object detection. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 7276–7282. IEEE (2019)
43. Sun, P., Kretschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2446–2454 (2020)
44. Sun, P., Wang, W., Chai, Y., Elsayed, G., Bewley, A., Zhang, X., Sminchisescu, C., Anguelov, D.: Rsn: Range sparse net for efficient, accurate lidar 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5725–5734 (2021)
45. Wang, C., Ma, C., Zhu, M., Yang, X.: Pointaugmenting: Cross-modal augmentation for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11794–11803 (2021)

46. Wang, Y., Fathi, A., Kundu, A., Ross, D.A., Pantofaru, C., Funkhouser, T., Solomon, J.: Pillar-based object detection for autonomous driving. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16. pp. 18–34. Springer (2020)
47. Wong, K., Wang, S., Ren, M., Liang, M., Urtasun, R.: Identifying unknown instances for autonomous driving. In: Conference on Robot Learning. pp. 384–393. PMLR (2020)
48. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. *Sensors* **18**(10), 3337 (2018)
49. Yang, Z., Sun, Y., Liu, S., Jia, J.: 3dssd: Point-based 3d single stage object detector. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11040–11048 (2020)
50. Yao, L., Han, J., Wen, Y., Liang, X., Xu, D., Zhang, W., Li, Z., Xu, C., Xu, H.: Detclip: Dictionary-enriched visual-concept paralleled pre-training for open-world detection. arXiv preprint arXiv:2209.09407 (2022)
51. Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3d object detection and tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11784–11793 (2021)
52. Yin, T., Zhou, X., Krähenbühl, P.: Multimodal virtual point 3d detection. *Advances in Neural Information Processing Systems* **34**, 16494–16507 (2021)
53. Yoo, J.H., Kim, Y., Kim, J., Choi, J.W.: 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16. pp. 720–736. Springer (2020)
54. You, Y., Luo, K., Phoo, C.P., Chao, W.L., Sun, W., Hariharan, B., Campbell, M., Weinberger, K.Q.: Learning to detect mobile objects from lidar scans without labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1130–1140 (2022)
55. Yurtsever, E., Lambert, J., Carballo, A., Takeda, K.: A survey of autonomous driving: Common practices and emerging technologies. *IEEE access* **8**, 58443–58469 (2020)
56. Zang, Y., Li, W., Zhou, K., Huang, C., Loy, C.C.: Open-vocabulary detr with conditional matching. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX. pp. 106–122. Springer (2022)
57. Zareian, A., Rosa, K.D., Hu, D.H., Chang, S.F.: Open-vocabulary object detection using captions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14393–14402 (2021)
58. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L.M., Shum, H.Y.: Dino: Detr with improved denoising anchor boxes for end-to-end object detection. arXiv preprint arXiv:2203.03605 (2022)
59. Zhang, L., Yang, A.J., Xiong, Y., Casas, S., Yang, B., Ren, M., Urtasun, R.: Towards unsupervised object detection from lidar point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9317–9328 (2023)
60. Zhang, R., Guo, Z., Zhang, W., Li, K., Miao, X., Cui, B., Qiao, Y., Gao, P., Li, H.: Pointclip: Point cloud understanding by clip. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8552–8562 (2022)
61. Zhang, R., Zhang, H., Yu, H., Zheng, Z.: Approaching outside: Scaling unsupervised 3d object detection from 2d scene. In: Proceedings of the European conference on computer vision (ECCV) (2024)

- 62. Zhou, X., Girdhar, R., Joulin, A., Krähenbühl, P., Misra, I.: Detecting twenty-thousand classes using image-level supervision. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX. pp. 350–368. Springer (2022)
- 63. Zhou, Y., Sun, P., Zhang, Y., Anguelov, D., Gao, J., Ouyang, T., Guo, J., Ngiam, J., Vasudevan, V.: End-to-end multi-view fusion for 3d object detection in lidar point clouds. In: Conference on Robot Learning. pp. 923–932. PMLR (2020)
- 64. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4490–4499 (2018)
- 65. Zhu, X., Zhang, R., He, B., Guo, Z., Zeng, Z., Qin, Z., Zhang, S., Gao, P.: Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. arXiv preprint arXiv:2211.11682 (2022)