Unsupervised 3D Perception with 2D Vision-Language Distillation for Autonomous Driving

Mahyar Najibi* Jingwei Ji* Yin Zhou† Charles R. Qi Xinchen Yan
Scott Ettinger Dragomir Anguelov
Waymo LLC

Abstract

Closed-set 3D perception models trained on only a pre-defined set of object categories can be inadequate for safety critical applications such as autonomous driving where new object types can be encountered after deployment. In this paper, we present a multi-modal auto labeling pipeline capable of generating amodal 3D bounding boxes and tracklets for training models on open-set categories without 3D human labels. Our pipeline exploits motion cues inherent in point cloud sequences in combination with the freely available 2D image-text pairs to identify and track all traffic participants. Compared to the recent studies in this domain, which can only provide class-agnostic auto labels limited to moving objects, our method can handle both static and moving objects in the unsupervised manner and is able to output open-vocabulary semantic labels thanks to the proposed vision-language knowledge distillation. Experiments on the Waymo Open Dataset show that our approach outperforms the prior work by significant margins on various unsupervised 3D perception tasks.

1. Introduction

In autonomous driving, most existing 3D detection models [62, 22, 42] have been developed with the prior assumption that all possible categories of interest should be known and annotated during training. While significant progress has been made in this supervised closed-set setting, these methods still struggle to fully address the safety concerns that arise in high-stakes applications. Specifically, in the dynamic real-world environment, it is unacceptable for autonomous vehicles to fail to handle a category that is not present in the training data. To address this safety concern, a recent development by Najibi *et al*. [35] proposed an unsupervised auto labeling pipeline that uses motion cues from point cloud sequences to localize 3D objects. However, by design, this method does not localize static objects which constitute a significant portion of traffic participants. Moreover, it only models the problem in a class-agnostic way and fails to provide semantic labels for scene understanding. This is suboptimal as semantic information is essential for downstream tasks such as motion planning, where category-specific safety protocols are deliberately added to navigate through various traffic participants.

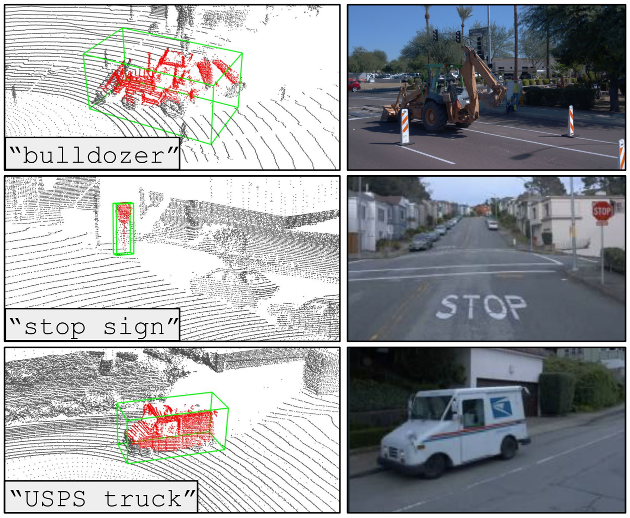

Figure 1. An illustration of three interesting urban scene examples of open-vocabulary perception. Left: our method can faithfully detect objects based on user-provided text queries during inference, without the need for 3D human supervision. Red points are points matched with the text queries. Right: camera images for readers' reference. Note that the inference process solely relies on LiDAR points and does not require camera images.

Recently, models trained with large-scale image-text datasets have demonstrated robust flexibility and generalization capabilities for open-vocabulary image-based classification [38, 19, 33], detection [20, 12, 24, 59] and semantic segmentation [23, 11] tasks. Yet, open-vocabulary recognition in the 3D domain [9, 15, 40] is in its early stages. In the context of autonomous driving it is even more underexplored. In this work, we fill this gap by leveraging a pre-

[*] Equal contribution
[†] Corresponding author

trained vision-language model to realize open-vocabulary 3D perception in the wild.

We propose a novel paradigm of Unsupervised 3D Perception with 2D Vision-Language distillation (UP-VL). Specifically, by incorporating a pre-trained vision-language model, UP-VL can generate auto labels with substantially higher quality for objects in arbitrary motion states, compared to the latest work by Najibi *et al.* [35].

With our auto labels, we propose to co-train a 3D object detector with a knowledge distillation task, which can achieve two goals simultaneously, *i.e.* improving detection quality and transferring semantic features from 2D image pixels to 3D LiDAR points. The perception model therefore is capable of detecting all traffic participants and thanks to the distilled open-vocabulary features, we can flexibly query the detector's output embedding with text prompts, for preserving specific types of objects at inference time (see Figure 1 for some examples).

We summarize the contributions of UP-VL as follows:

- UP-VL achieves state-of-the-art performance on unsupervised 3D perception (detection and tracking) of moving objects for autonomous driving.
- UP-VL introduces semantic-aware unsupervised detection for objects in any motion state, a first in the field of autonomous driving. This breakthrough eliminates the information bottleneck that has plagued previous work [35], where class-agnostic auto labels were used, covering only moving objects with a speed above a predetermined threshold.
- UP-VL enables 3D open-vocabulary detection of novel objects in the wild, with queries specified by users at inference time, therefore removing the need to re-collect data or re-train models.

2. Related works

Vision-language training. Contrastive vision language training on billions of image-text training pairs resulted in impressive improvements in the tasks of open-set and zero-shot image classification and language related applications [38, 19, 57]. More recently, open-set object localization in 2D images has been shown to benefit from such abundant image-text data as well. Specifically, [20, 12, 24, 59, 60, 32] used image-text training to improve the open-set capability of 2D object detectors and [23, 11] explored the use of large-scale scene-level vision-language data for the task of open-set 2D semantic segmentation. Recent research [30, 21, 50, 13, 17] has begun to explore the application of 2D vision-language pre-training in 3D perception tasks. However, these studies focused on static indoor scenario where the scene is small-scale and the RGB-D data is captured in high-resolution. Here we design a multi-modal pipeline that leverages vision-language pre-training for unsupervised open-set 3D perception in complex, sparse, and occlusion-rich environments for autonomous driving.

Unsupervised 3D object detection. Unsupervised 3D object detection from LiDAR data is largely under-explored [7, 53, 49, 27, 35]. Dewan *et al.* [7] proposed a model-free method to detect and track the visible part of objects, by using the motion cues from LiDAR sequences. However, this approach is incapable of generating amodal bounding boxes which is essential for autonomous driving. Cen *et al.* [3] relied on a supervised detector to produce proposals of unknown categories. However, this approach requires full supervision to train the base detector and has limited generalization capability to only semantically similar categories. Wong *et al.* [53] identified unknown instances via supervised segmentation and clustering, which by design cannot generate amodal boxes from partial observations. Most recently, Najibi *et al.* [35] developed an unsupervised auto meta labeling pipeline to generate pseudo labels for moving objects, which can be used to train real-time 3D detection models. This approach fails to provide semantics to detection boxes and ignores static objects, which limits its practical utility. Compared to all previous efforts, we realize open-vocabulary unsupervised 3D detection for both static and moving objects, by leveraging vision-language pre-training, and benchmark our system on the realistic and challenging scenario of autonomous driving. While utilizing 2D vision-language models that may have been pre-trained with human annotations, we avoid the need for any additional 3D labels within our paradigm, thereby creating a pragmatically unsupervised setting.

LiDAR 3D object detection. Most previous works focused on developing performant model architectures in the fully supervised setting, without considering the generalization capability to long-tail cases and unknown object types that are prevalent in the dynamic real world. These methods can be categorized into point based [42, 37, 55, 43, 34, 25], voxelization based [8, 51, 45, 36, 54, 44, 62, 22, 52, 56, 58, 5, 29], perspective projection based [31, 2, 10], and feature fusion [48, 6, 61, 14, 41]. Recent research also explore transferring knowledge from image for 3D point cloud understanding [39, 28, 18, 4]. Our method is compatible with any 3D detector, extending it to handle the open-set settings.

3. Method

We present UP-VL, a new approach for unsupervised open-vocabulary 3D detection and tracking of traffic participants. UP-VL advances the previous state-of-the-art [35] which was limited to *class-agnostic* detection of *moving-only* objects in two main directions: 1) It enables *class-aware* open-set 3D detection by incorporating open-vocabulary text queries at inference time, and 2) It is able

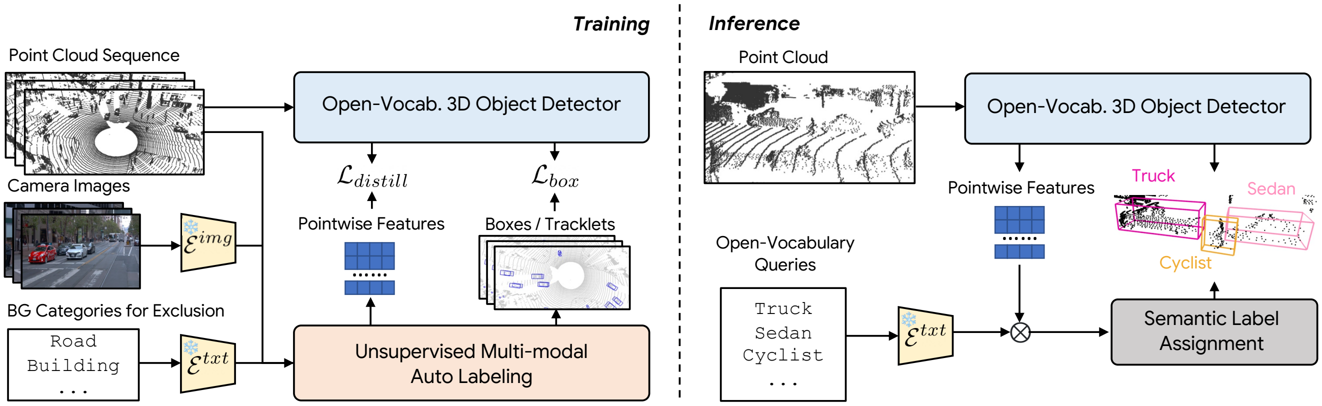

Figure 2. Overview of the proposed UP-VL framework. During training (left), our method taps into multi-modal inputs (LiDAR, camera, text) and produces high-quality auto supervisions, via Unsupervised Multi-modal Auto Labeling, including 3D point-level features, 3D object-level bounding boxes and tracklets. Our auto labels are then used to supervise a class-agnostic open-vocabulary 3D detector. Besides, our 3D detector distills the features extracted from a pre-trained 2D vision-language model. At inference time (right), our trained 3D detector produces class-agnostic boxes and per-point features in the embedding space of the pre-trained vision-language model. We then use the text encoder to map queries to the embedding space and compute the per-point similarity scores between the predicted feature and the text embeddings (⊗ refers to cosine similarity). These per-point scores are then aggregated to assign semantic labels to boxes.

to detect objects in *all motion states* as opposed to moving-only objects in the previous study. To achieve these goals, we deploy a multi-modal approach and combine intrinsic motion cues [35] available from the LiDAR sequences with the semantics captured by a vision-language model [11] trained on generic image-text pairs from the Internet. An overview of our approach is shown in Figure 2. As illustrated on the left, our training pipeline involves two main stages. First, our auto labeling method uses these motion and semantic cues to automatically label the raw sensor data, yielding class-agnostic 3D bounding boxes and tracklets as well as point-wise semantic features. Then, in the second stage, we use these auto labels to train open-vocabulary 3D perception models. The right side of the figure illustrates our inference pipeline where given raw LiDAR point clouds, our detector is able to perform open-vocabulary 3D detection given a set of text queries.

3.1. Background

The key challenges in unsupervised 3D perception are twofold: 1) generating high-quality 3D amodal bounding boxes and consistent tracklets for all open-set traffic participants, and 2) inferring per-object semantics. Najibi *et al*. [35] developed an auto labeling technique to address the first challenge partially. Their approach focuses on moving objects only. Specifically, their method takes LiDAR sequences as input, and removes ground points. It then breaks down the scene into individual connected components (*i.e*. point clusters). Next, it calculates local flow between pairs of point clusters from two adjacent frames and retains only clusters with speed above a predefined threshold. It then tracks each cluster across frames and aggregates points to obtain a more comprehensive view of the object, which enables the derivation of a faithful 3D amodal bounding box. Finally, the resulting 3D amodal boxes and tracklets can serve as auto labels for training 3D perception models.

While the previous work [35] has shown promising results, it suffers from significant limitations: 1) it can only deal with moving objects; and 2) it is unable to output semantics. These limitations hinder its practical utility for safety-critical applications such as autonomous driving.

3.2. Unsupervised Multi-modal Auto Labeling

In contrast to the traditional way of training a detection model by presenting box geometries and closed-set semantics, our unsupervised multi-modal auto labeling approach produces box geometries and point-wise semantic feature embeddings, where the former teaches the detector to localize all traffic participants and the latter informs the model to preserve certain types of objects based on the inference-time text queries.

Figure 3 shows an overview of the auto labeling pipeline and Algorithm 1 presents its details. Specifically, our system leverages multiple modalities as input, namely camera images, LiDAR point sequences, and natural language. It also employs a pre-trained vision-language model [11] to extract feature embeddings from images and texts, which naturally complements the 3D depth information and motion cues with rich semantics, compared to [35]. We begin by detailing the feature extraction process. We then describe how we utilize the extracted vision-language information in combination with the inherent motion cues from LiDAR sequences to generate auto labels in an unsupervised manner.

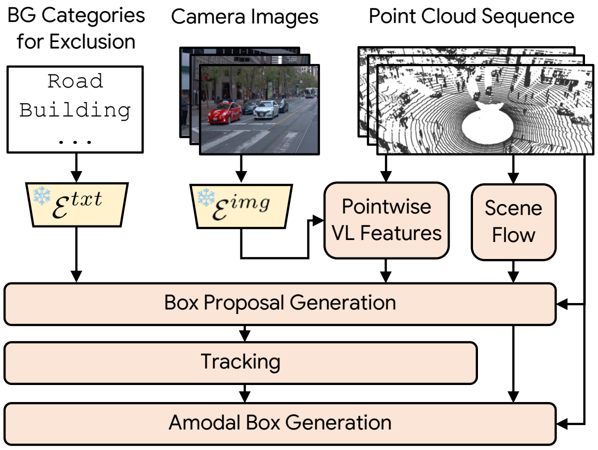

Figure 3. Overview of our unsupervised multi-modal auto labeling approach. This pipeline first extracts vision-language and motion features from multiple modalities, then proposes, tracks and completes bounding boxes of objects. The resulting pointwise VL features, 3D bounding boxes and tracklets will serve as automatic supervisions to train the perception model.

Feature Extraction

As the first step to our approach, we start by extracting open-vocabulary features from all available cameras and then transfer these 2D features to 3D LiDAR points using known sensor calibrations. Specifically, at each time t, we have a set of images $\{\mathbf{I}_t^k \in \mathbb{R}^{H_k \times W_k \times 3}\}_t$ captured by K cameras, where H_k and W_k are image dimensions of the camera k. We also have a collection of point cloud, $\{\mathbf{P}_t \in \mathbb{R}^{N_t \times 3}\}$, captured over time using LiDAR sensors. Here, N_t denotes the number of points at time t. We use a pre-trained open-vocabulary 2D image encoder $\mathcal{E}^{img}$ to extract the pixel-wise visual features for each image, denoted as $\{\mathbf{V}_t^k \in \mathbb{R}^{H_k \times W_k \times D}\}$, where D represents the feature dimension. Next, we build the mapping between 3D LiDAR points and their corresponding image pixels using the camera and LiDAR calibration information. Once this mapping is created, we can associate each 3D point with its corresponding image feature vector. As a result, we obtain vision-language features for all the 3D points as $\mathbf{F}_t^{vl} \in \mathbb{R}^{N_t \times D}$, where N_t is the number of points at time t.

Additionally, we leverage motion signals as another crucial representation that can substantially aid in deducing the concept of objectness for moving instances in the open-set environment. Specifically, we employ the NSFP++ algorithm [35] to compute the scene flow $\mathbf{F}_t^{sf} \in \mathbb{R}^{N_t \times 3}$ of points at each time t, which is a set of flow vectors corresponding to each point in $\mathbf{P}_t$.

Bounding Box Proposal Generation

At each time step, we generate initial bounding box proposals $\{\mathbf{B}_t^{vis} \in \mathbb{R}^{M_t \times 7}\}$ by clustering the points, where M_t is the number of boxes at time t, and each box is parameterized as (center x, center y, center z, length, width, height, heading). Note that vis indicates that each box only covers the visible portion of an object. To cluster each point, we leverage a set of features which includes the point locations $\mathbf{P_t}$, scene flow $\mathbf{F}_t^{sf}$, and the vision-language features $\mathbf{F}_t^{vl}$.

Algorithm 1 Unsupervised multi-modal auto labeling.

Input: A sequence of images across T frames for each of the K cameras $\{\mathbf{I}_t^k\}$; a sequence of LiDAR point locations $\{\mathbf{P}_t\}$.
Requires: Cosine similarity threshold for background categories ϵ^{bg}; minimum scene flow magnitude ϵ^{sf}; maximum ratio of background points within a box r^{bg}; a set of a prior background categories $\mathbf{C}^{bg}$; a pre-trained open-vocabulary model with image encoder $\mathcal{E}^{img}$ and text encoder $\mathcal{E}^{txt}$.
Output: Amodal 3D bounding boxes $\{\mathbf{B}_t\}$ and their track IDs $\{\mathbf{T}_t\}$; point-wise open-vocabulary features $\{\mathbf{F}_t^{vl}\}$.
Function:

```
for $t = 1$ to $T$ do
    $\{\mathbf{V}_t^k\} \leftarrow \mathcal{E}^{img}(\{\mathbf{I}_t^k\})$    ▷ 2D VL features
    $\mathbf{F}_t^{vl} \leftarrow \text{Unprojection}(\{\mathbf{V}_t^k\}, \mathbf{P}_t)$    ▷ 3D VL features
    if $t \neq T$ then
        $\mathbf{F}_t^{sf} \leftarrow \text{NSFP++}(\mathbf{P}_t, \mathbf{P}_{t+1})$    ▷ Scene flow
    else
        $\mathbf{F}_t^{sf} \leftarrow -\text{NSFP++}(\mathbf{P}_t, \mathbf{P}_{t-1})$
    for $i = 1$ to $N_t$ do
        $(\mathbf{M}_t^{sf})_i \leftarrow \mathbb{1}(\|(\mathbf{F}_t^{sf})_i\| \geq \epsilon^{sf})$
        $(\mathbf{M}_t^{bg})_i \leftarrow \mathbb{1}(\max_{c \in \mathbf{C}^{\text{bg}}} \frac{(\mathbf{F}_t^{vl})_i \cdot \mathcal{E}^{txt}(c)}{\|(\mathbf{F}_t^{vl})_i\| \|\mathcal{E}^{txt}(c)\|} \geq \epsilon^{bg})$
    $\widetilde{\mathbf{P}}_t, \widetilde{\mathbf{F}}_t^{sf} \leftarrow \mathbf{P}_t[\mathbf{M}_t^{sf}], \mathbf{F}_t^{sf}[\mathbf{M}_t^{sf}]$
    $\mathbf{B}_t^{vis} \leftarrow \text{InitialBoxProposal}(\widetilde{\mathbf{P}}_t, \widetilde{\mathbf{F}}_t^{sf}, \mathbf{M}_t^{bg}; r^{bg})$
$\{\mathbf{T}_t\} \leftarrow \text{Tracking}(\{\mathbf{B}_t^{vis}\})$
$\{\mathbf{B}_t\} \leftarrow \text{AmodalBoxGeneration}(\{\mathbf{B}_t^{vis}\}, \{\mathbf{T}_t\}, \{\mathbf{P}_t\})\}$
return $\{\mathbf{B}_t\}$, $\{\mathbf{T}_t\}$, $\{\mathbf{F}_t^{vl}\}$
```

We design our pipeline to flexibly generate auto labels for objects in desired motion states. Given scene flow $\mathbf{F}_t^{sf}$, we introduce a velocity threshold ϵ^{sf} to select points whose speed is greater than or equal to the threshold (*e.g.*, 1.0 m/s). To capture objects in all motion states, we set $\epsilon^{sf} = 0$.

One major challenge of auto labeling objects in all motion states is how to automatically distinguish traffic participants (*e.g.*, vehicles, pedestrians, *etc.*) from irrelevant scene elements (*e.g.*, street, fence, *etc.*). We propose to leverage an a priori list of *background* object categories to exclude irrelevant scene elements from labeling. Specifically, we use the text encoder, $\mathcal{E}^{txt}$, from the pre-trained 2D vision-language model [11], to encode each background category name c into its feature embedding $\mathcal{E}^{txt}(c) \in \mathbb{R}^D$. We further define a per-point binary background mask, denoted as $\mathbf{M}_t^{bg} \in \{0, 1\}^{N_t}$, that takes on a value of 1 if a point is assigned to one of the a priori background categories, or 0 otherwise. See Algorithm 1 for the definition of $\mathbf{M}_t^{bg}$, where $(\cdot)_i$ denotes the i-th row of a matrix and $\mathbb{1}(\cdot)$ represents the indicator function. We use this background mask to mark scene elements which are not of interest.

We then proceed to cluster the point cloud into neighboring regions using a spatio-temporal clustering algorithm, modified from [35], followed by calculating the tightest bounding box around each cluster. In addition to clustering points by their locations and motions, we also use $\mathbf{M}_t^{bg}$ to eliminate bounding boxes which are likely to be background. To be precise, we discard any bounding box in which the ratio of background points exceeds a threshold of r^{bg} (which is set to 99%). This process results in the initial set of bounding box proposals $\{\mathbf{B}_t^{vis}\}$. Note that in this step, the box dimensions are determined based on the *visible* portion of each object, which can be significantly underestimated compared to the human labeled amodal box, due to ubiquitous occlusions and sparsity.

Amodal Auto Labeling

In autonomous driving, perception downstream tasks desire *amodal* boxes that encompass both the visible and occluded parts of the objects. To transform our visible-only proposals to amodal auto labels, we follow [35] by adopting a tracking-by-detection paradigm with Kalman filter state updates to link all proposals over time. We then perform shape registration for each object track of $\{\mathbf{T}_t\}$ using ICP [1]. Within each track, we leverage the intuition that different viewpoints contain complementary information and temporal aggregation of the registered points from proposals would allow us to obtain a complete shape of the object. Hence, we fit a new box to the aggregated points to yield the amodal box. Finally, we undo the registration from aggregated points to individual frames and replace the original visible box proposal at each time step with the amodal box, which produces auto labeled 3D boxes and the tracklet.

In practice, background point filtering, point cloud registration and temporal aggregation may contain noise, leading to spurious boxes, *e.g.*, tiny and sizable boxes and overlapping boxes. We apply non-maximum suppression (NMS) to clean the auto label boxes. This final set of unsupervised amodal auto labels $\{\mathbf{B}_t\}$, their track IDs $\{\mathbf{T}_t\}$, together with the extracted vision-language embeddings $\{\mathbf{F}_t^{vl}\}$, are then used to train open-vocabulary 3D object detection model as described in Sec. 3.3.

3.3. Open-vocabulary 3D Object Detection

In this subsection, we describe how the unsupervised auto labels, can be used to train a 3D object detector capable of localizing open-set objects and assigning open-vocabulary semantics to them, all without using any 3D human annotations during training.

3.3.1 Model Architecture

Our design, as depicted in Figure 2, is based on decoupling object detection into class-agnostic object localization and semantic label assignment. For class-agnostic bounding box prediction, we add a branch to a 3D point cloud encoder backbone to generate 3D bounding box center, dimensions, and heading. This branch accompanies a binary classification branch which outputs foreground / background class-agnostic per box objectness score. To supervise these two branches, we treat our unsupervised auto labels (see Sec. 3.2) as ground-truth and add bounding box regression and classification losses to our learning objective. We would like to highlight that our pipeline is independent of a specific 3D point-cloud encoder [22, 61, 47] and the detection paradigm (either anchor-based or anchor-free detection). Here, we adopt an anchor-based PointPillar backbone [22] with Huber loss for box residual regression and Focal Loss [26] for objectness classification to have a fair comparison with prior works [35]. Besides predicting 3D bounding boxes, we also perform text query-based open-vocabulary semantic assignment by distilling knowledge from pre-trained 2D vision-language models using an extra branch which is described in the next subsections.

3.3.2 Vision-Language Knowledge Distillation

Besides class agnostic bounding box generation, our 3D detector pipeline also distills the semantic knowledge from the per-point vision-language features provided by our auto labeling pipeline (*i.e.* $\{\mathbf{F}_t^{vl}\}$, introduced in the vision-language feature extraction in Sec. 3.2). In our method, we directly distill these features, which as will be discussed in the next subsection, unlocks text query-based open-vocabulary category assignment at inference time. More precisely, as shown in the left side of Figure 2, we add a new linear branch to the model to predict per-point D dimensional features (here D is the dimensionality of the vision-language embedding space). As the input to this branch, we scatter the computed voxelized features in our backbone back into the points and concatenate them with the available per-point input features (*i.e.* 3D point locations and LiDAR intensity and elongation features). We then train the network to predict the feature vector $\mathbf{f}_p^{vl} \in \mathbf{F}_t^{vl}$ for any point $\mathbf{p}$ visible in the camera images and add the following loss to the training objective:

$$\mathcal{L}_{\text{distill}}(\mathbf{p}) = \text{CosineDist}(\mathbf{y}_p, \mathbf{f}_p^{vl}) \tag{1}$$

where $\mathbf{y}_p$ is the distillation prediction by the model for point $\mathbf{p}$. This together with the bounding box regression and the objectness classification losses (based on our auto labels as discussed in Sec. 3.3.1) form our final training objective.

3.3.3 Open-Vocabulary Inference

So far, we have introduced how to train a detector to simultaneously localize all objects in a class-angnostic manner and predict vision-language features for all LiDAR points. Here, we discuss how we assign open-vocabulary semantics to the predicted boxes during inference. This process is depicted in the right side of Figure 2. The pre-trained 2D vision language model [11] contains an image encoder and a text encoder, which are jointly trained to map text and image data to a shared embedding space. As described in Sec. 3.3.2, we add a feature distillation branch that maps 3D input point clouds to the 2D image encoder embedding space, which essentially bridges the gap between point clouds and semantic text queries. As a result, at the inference time we can encode arbitrary open-vocabulary categories presented as text queries and compute their similarities with the observed 3D points. This can be achieved by computing the cosine similarity between the text query embeddings and the vision-language features predicted by our model for each 3D point. Finally, we assign open-vocabulary categories to boxes based on majority voting. Specifically, we associate each point the category with the highest computed cosine similarity, and then assign to each box the most common category of its enclosing points.

We would like to emphasize that our approach does not need to process images at inference time, since we have distilled image encoder features to the point cloud. Therefore, the only added computation is a simple linear layer for predicting per-point vision-language embeddings, which is negligible compared to the rest of the detector architecture.

4. Experiments

Our UP-VL approach advances the previous state-of-the-art in unsupervised 3D perception for autonomous driving [35] in two main important directions: 1) enabling open-vocabulary category semantics and 2) detecting objects in all motion states (as opposed to moving-only objects in the previous study). In this section, we perform extensive evaluations with respect to each of these innovations. Note that unsupervised open-set 3D detection is still at early stage in the research community with few published works. Therefore to fairly compare with the state-of-the-art [35], we perform our detection experiments first following the same setting as [35] (*i.e.* detecting class-agnostic moving objects) and then showcasing our new capabilities (*i.e.* detecting objects in any motion states with semantics).

Sec. 4.2 studies the performance of our system in the class-agnostic setting. This allows us to compare our approach with the existing state-of-the-art method on detecting moving-only objects, showing large improvements. Sec. 4.3 moves the needle beyond the capability of the previous class-agnostic state-of-the-art methods and reports results under open-vocabulary class-aware setting for detecting moving-only objects (Sec. 4.3.1) and the most challenging setting of open-vocabulary detection of objects in all motion states (Sec. 4.3.2). Finally, Sec. 4.4 reports the open-set tracking quality of our auto labels and Sec. 4.5 presents qualitative results. See supplementary materials for more ablation studies and error analyses.

Table 1. Comparison of the methods on class-agnostic unsupervised 3D detection of *moving* objects. Top: Auto label boxes. Bottom: Detection boxes.

Method	Box Type	3D AP@0.4		3D AP@0.5	
		L1	L2	L1	L2
MI-UP [35]	Auto labels	36.9	35.5	27.4	26.4
UP-VL (ours)		**39.9**	**38.4**	**34.2**	**32.0**
MI-UP [35]	Detections	42.1	40.4	29.6	28.4
UP-VL (ours)		**49.9**	**48.1**	**38.4**	**36.9**

4.1. Experimental Setting

We evaluate our framework using the challenging Waymo Open Dataset (WOD) [46], which provides a large collection of run segments captured by multi-modal sensors in diverse environment conditions. To define moving-only objects in Sec. 4.2, we follow [35] and apply a threshold of 1.0 m/s (*i.e.* $\epsilon^{sf} = 1.0$). We set the cosine similarity threshold for background categories at $\epsilon^{bg} = 0.02$ to achieve best performance in practice. The background categories C^{bg} we exclude from auto labeling are "vegetation", "road", "street", "sky", "tree", "building", "house", "skyscaper", "wall", "fence", and "sidewalk". The WOD [46] has three common object categories, *i.e.* vehicle, pedestrian, and cyclist. In the class-aware 3D detection experiments (Sec. 4.3), we follow [35] and combine pedestrian and cyclist into one VRU (vulnerable road users) category, which contains a similar number of labels as the vehicle category. As in [35], we also train and evaluate the detectors on a 100m × 40m rectangular region around the ego vehicle. We use the popular PointPillars detector [22] for all our detection experiments and set an intersection over union, IoU=0.4, for evaluations unless noted otherwise. Please refer to Sec. 1 of supplementary materials for a more detailed description of all experimental settings.

4.2. Class-agnostic Unsupervised 3D Detection of Moving Objects

For fair comparison, we follow the same setting as [35] and tailor our approach to class-agnostic moving-only 3D detection. Specifically, we perform auto labeling as introduced in 3.2 with speed threshold $\epsilon^{sf} = 1.0$m/s and train a class-agnostic detector with feature distillation as described in 3.3.1. However, we disable text queries at inference time. Note that [35] only considered detection of moving objects.

We leave the study of more challenging settings to Sec. 4.3.

Table 1 shows our result and compares it with MI-UP [35]. The top part of the table compares the auto labeling quality. The bottom part compares the detector performance between our UP-VL approach and MI-UP. We use the exact same detection backbone and hyper-parameters to ensure a fair comparison. When evaluating at IoU=0.4 as suggested by [35], UP-VL significantly outperforms MI-UP, both in terms of the auto label as well as the detection performance. To better demonstrate our improved auto label quality, we also evaluate with a higher localization criterion at IoU=0.5, where our improvement becomes even more pronounced. We should also point out that in both methods, the final detection quality is superior to the auto label quality. We hypothesize that this is due to the network being able to learn a better objectness scoring function for ranking as well as its ability to denoise the auto labels given the inductive bias of the model [16].

4.3. Class-aware Unsupervised Open-vocabulary 3D Detection

In this section, we evaluate the capability of our UP-VL pipeline in class-aware open-vocabulary 3D detection of objects in different motion states. Please note that we don't use any 3D human annotations during training and only use the available human labeled categories for evaluation. Moreover, it should be noted that the previous state-of-the-art [35], as a class-agnostic approach, falls short in this new setting, making comparisons not possible. In all experiments in this section, we assign labels to boxes by querying category names as text at inference time in an open-vocabulary fashion as described in Sec. 3.3.3 (see Sec. 1 of supplementary for a detailed list of text queries used).

4.3.1 Moving-only Objects

Table 2 reports the class-aware open-vocabulary 3D detection results on the moving-only objects. Since [35] is no longer applicable in this setting, we construct two baselines for comparison: *i.e*. geometric clustering [35] which additionally uses our extracted scene flow features ($\mathbf{F}_t^{sf}$) and its variant which leverages both the scene flow features and the vision-language features ($\mathbf{F}_t^{vl}$). 3D point-wise semantics for the baselines are extracted directly by projecting the 2D image features of the pre-trained vision-language model. We report per-category AP as well as the mAP of these baselines in the top two rows of Table 2. The bottom of the table presents the results for our unsupervised auto labels and our final UP-VL detections. Our auto labels and UP-VL detector both outperform baselines constructed from prior approaches. As discussed in Sec. 3.3.2, unlike the baselines that requires applying the image encoder to all camera images at inference time, our detector directly predicts image features extracted by our auto labeling pipeline for 3D point clouds and consequently is more efficient.

4.3.2 Objects in All Motion States

Finally in this section, we report results on the most challenging setting: unsupervised class-aware open-vocabulary 3D detection for all objects with arbitrary motion states. Like Sec. 4.3.1, since [35] falls short in this setting, we construct three clustering baselines using different combinations of our features. More specifically, the first row only uses point locations ($\mathbf{P}_t$), the second row uses both point locations and our vision-language features ($\mathbf{F}_t^{vl}$), and the third row leverages all the features including our scene flow features ($\mathbf{F}_t^{sf}$). As an ablation on the effectiveness of the introduced feature distillation in UP-VL, we also add a baseline called "Our detector w/o feature distillation", where we remove the distillation head and its loss from our detector, and like the baselines in the first three rows, we directly project the vision-language features from camera images to the point cloud for semantic label assignment. As summarized in Table 3, our auto labels significantly outperform other baselines listed in the first three rows. Moreover, comparing the last two rows, we observe that the proposed vision-language feature distillation leads to significant performance improvement aross all metrics. For example, our approach with feature distillation outperforms the counterpart without distillation by more than 8 points in mAP.

4.4. Tracking

The UP-VL exhibits a high performance not only in detection, but also in tracking - a critical task in autonomous driving. We employ the motion-based tracker from [35], and conduct experiments in the tracking-by-detection manner. We evaluate tracking performance for moving objects and compare our UP-VL detector trained with feature distillation as outlined in Table 2 against two baselines: MI-UP detector from Table 1 and another open-set baseline from Table 2. To measure the effectiveness of our model, we employ the widely used MOTA and MOTP metrics, both in the class-agnostic and class-aware open-vocabulary settings. Our experimental results (Table 4) demonstrate that UP-VL outperforms both baselines by a significant margin.

4.5. Qualitative Results

Our UP-VL enables open-vocabulary detection of arbitrary object types beyond the few human annotated categories in the autonomous driving datasets. Figure 4 illustrates some examples. In each row, we present the camera image on the right for readers' reference. On the left, we show the corresponding 3D point cloud and the predicted 3D bounding box by our model based on the open-vocabulary text query provided at inference time.

Table 2. Comparison of methods on unsupervised class-aware *moving* object detection. (*since semantics are not available, we report class agnostic AP for the first row, given that vehicle and VRU contain similar number of samples.)

Method	Representations		Box type	3D AP		mAP
	Motion	Vision-Language		Veh	VRU	
Clustering [35]	✓		visible	N/A	N/A	32.4*
Clustering [35] + OpenSeg [11]	✓	✓	visible	47.8	21.5	34.7
Our auto labels	✓	✓	amodal	57.5	**29.8**	43.7
Our UP-VL detector w. feature distillation	✓	✓	amodal	**76.9**	28.6	**52.8**

Table 3. Comparison of methods on unsupervised class-aware detection of objects in *all motion states*. (*since semantics are not available, we report class-agnostic AP for the first row, given that vehicle and VRU contain similar number of samples.)

Method	Representations		Box type	3D AP		mAP
	Motion	Vision-Language		Veh	VRU	
Clustering [35]			visible	N/A	N/A	11.6*
Clustering [35] + OpenSeg [11]		✓	visible	15.8	9.9	12.9
Clustering [35] + OpenSeg [11]	✓	✓	visible	16.1	10.0	13.1
Our auto labels	✓	✓	amodal	30.2	14.7	22.4
Our detector w/o feature distillation	✓	✓	amodal	40.0	15.2	27.6
Our UP-VL detector w. feature distillation	✓	✓	amodal	**52.0**	**19.7**	**35.8**

Table 4. Comparison of tracking methods for moving objects with evaluations in class-agnostic (Cls. ag.) and class-aware settings. "MI-UP-C" refers to class-agnostic MI-UP clustering approach, which is unable to be evaluated in the class-aware setting.

Method	MOTA (↑) / MOTP (↓)		
	Veh	VRU	Cls. ag.
MI-UP [35]	N/A	N/A	12.8/45.5
MI-UP-C [35] + OpenSeg [11]	39.6/37.4	13.5/53.7	22.8/43.4
UP-VL detector	**65.3/31.0**	**24.0/46.8**	**41.3/37.4**

5. Conclusions

In this paper, we study the problem of unsupervised 3D object detection and tracking in the context of autonomous driving. We present a cost-efficient pipeline using multi-sensor information and an off-the-shelf vision-language model pre-trained on image-text pairs. Core to our approach is a multi-modal auto labeling pipeline, capable of generating class-agnostic amodal box annotations, tracklets, and per-point semantic features extracted from vision-language models. By combining the semantic information and motion cues observed from the LiDAR point clouds, our auto labeling pipeline can identify and track open-set traffic participants based on the raw sensory inputs. We have evaluated our auto labels by training a 3D open-vocabulary object detection model on the Waymo Open Dataset without any 3D human annotations. Strong results have been demonstrated on the task of open-vocabulary 3D detection with categories specified during inference by text queries which we believe opens up new directions towards more scalable software stacks for autonomous driving.

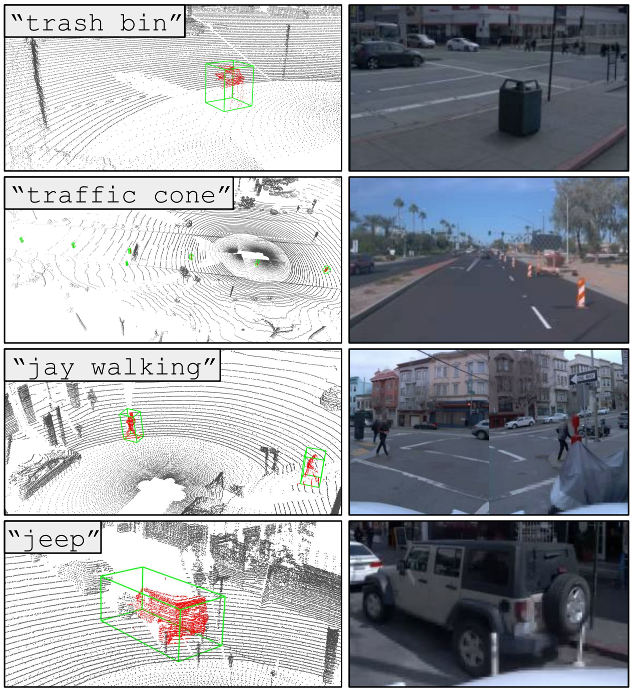

Figure 4. Open-vocabulary detection of both static and moving objects via user-provided text queries. Note that in the open-vocabulary setting, the text queries of interested object types are not given in either auto labeling or model training.

References

[1] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992. 5

[2] Alex Bewley, Pei Sun, Thomas Mensink, Dragomir Anguelov, and Cristian Sminchisescu. Range conditioned dilated convolutions for scale invariant 3d object detection, 2020. 2

[3] Jun Cen, Peng Yun, Junhao Cai, Michael Yu Wang, and Ming Liu. Open-set 3d object detection. In *3DV*, 2021. 2

[4] Runnan Chen, Youquan Liu, Lingdong Kong, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, Yu Qiao, and Wenping Wang. Clip2scene: Towards label-efficient 3d scene understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7020–7030, 2023. 2

[5] Yukang Chen, Yanwei Li, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Focal sparse convolutional networks for 3d object detection. In *CVPR*, 2022. 2

[6] Y. Chen, S. Liu, X. Shen, and J. Jia. Fast point r-cnn. In *ICCV*, 2019. 2

[7] Ayush Dewan, Tim Caselitz, Gian Diego Tipaldi, and Wolfram Burgard. Motion-based detection and tracking in 3d lidar scans. In *ICRA*, 2016. 2

[8] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. In *ICRA*, 2017. 2

[9] Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. In *NeurIPS*, 2022. 1

[10] Lue Fan, Xuan Xiong, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Rangedet: In defense of range view for lidar-based 3d object detection. In *ICCV*, 2021. 2

[11] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *ECCV*, 2022. 1, 2, 3, 4, 6, 8

[12] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *ICLR*, 2022. 1, 2

[13] Huy Ha and Shuran Song. Semantic abstraction: Open-world 3d scene understanding from 2d vision-language models. In *CoRL*, 2022. 2

[14] Chenhang He, Hui Zeng, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Structure aware single-stage 3d object detection from point cloud. In *CVPR*, June 2020. 2

[15] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. In *CoRL*, 2022. 1

[16] Minyoung Huh, Hossein Mobahi, Richard Zhang, Brian Cheung, Pulkit Agrawal, and Phillip Isola. The low-rank simplicity bias in deep networks, 2021. 7

[17] Ayush Jain, Nikolaos Gkanatsios, Ishita Mediratta, and Katerina Fragkiadaki. Bottom up top down detection transformers for language grounding in images and point clouds. In *ECCV*, 2022. 2

[18] Andrej Janda, Brandon Wagstaff, Edwin G Ng, and Jonathan Kelly. Self-supervised pre-training of 3d point cloud networks with image data. *arXiv preprint arXiv:2211.11801*, 2022. 2

[19] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 1, 2

[20] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *ICCV*, 2021. 1, 2

[21] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. 2022. 2

[22] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, 2019. 1, 2, 5, 6

[23] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. 2022. 1, 2

[24] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *CVPR*, 2022. 1, 2

[25] Zhichao Li, Feng Wang, and Naiyan Wang. Lidar r-cnn: An efficient and universal 3d object detector. In *CVPR*, 2021. 2

[26] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 5

[27] Yang Liu, Idil Esen Zulfikar, Jonathon Luiten, Achal Dave, Aljoša Ošep, Deva Ramanan, Bastian Leibe, and Laura Leal-Taixé. Opening up open-world tracking. In *CVPR*, 2022. 2

[28] Yueh-Cheng Liu, Yu-Kai Huang, Hung-Yueh Chiang, Hung-Ting Su, Zhe-Yu Liu, Chin-Tang Chen, Ching-Yu Tseng, and Winston H Hsu. Learning from 2d: Contrastive pixel-to-point knowledge transfer for 3d pretraining. *arXiv preprint arXiv:2104.04687*, 2021. 2

[29] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. 2023. 2

[30] Yuheng Lu, Chenfeng Xu, Xiaobao Wei, Xiaodong Xie, Masayoshi Tomizuka, Kurt Keutzer, and Shanghang Zhang. Open-vocabulary 3d detection via image-level class and debiased cross-modal contrastive learning. *arXiv preprint arXiv:2207.01987*, 2022. 2

[31] Gregory P Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, and Carl K Wellington. Lasernet: An efficient probabilistic 3d object detector for autonomous driving. In *CVPR*, 2019. 2

[32] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection with vision transformers. 2022. 2

[33] Yifei Ming, Ziyang Cai, Jiuxiang Gu, Yiyou Sun, Wei Li, and Yixuan Li. Delving into out-of-distribution detection with vision-language representations. In *NeurIPS*, 2022. 1

[34] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *ICCV*, 2021. 2

[35] Mahyar Najibi, Jingwei Ji, Yin Zhou, Charles R Qi, Xinchen Yan, Scott Ettinger, and Dragomir Anguelov. Motion inspired unsupervised perception and prediction in autonomous driving. In *ECCV*, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[36] Mahyar Najibi, Guangda Lai, Abhijit Kundu, Zhichao Lu, Vivek Rathod, Thomas Funkhouser, Caroline Pantofaru, David Ross, Larry S Davis, and Alireza Fathi. Dops: Learning to detect 3d objects and predict their 3d shapes. In *CVPR*, 2020. 2

[37] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *ICCV*, 2019. 2

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2

[39] Corentin Sautier, Gilles Puy, Spyros Gidaris, Alexandre Boulch, Andrei Bursuc, and Renaud Marlet. Image-to-lidar self-supervised distillation for autonomous driving data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9891–9901, 2022. 2

[40] Nur Muhammad Mahi Shafiullah, Chris Paxton, Lerrel Pinto, Soumith Chintala, and Arthur Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. *arXiv preprint arXiv:2210.05663*, 2022. 1

[41] 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 *CVPR*, 2020. 2

[42] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, 2019. 1, 2

[43] Weijing Shi and Ragunathan (Raj) Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *CVPR*, 2020. 2

[44] Martin Simony, Stefan Milzy, Karl Amendey, and Horst-Michael Gross. Complex-yolo: an euler-region-proposal for real-time 3d object detection on point clouds. In *ECCV*, 2018. 2

[45] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *CVPR*, 2016. 2

[46] 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 *CVPR*, 2020. 6

[47] 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 Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *ECCV*, 2022. 5

[48] Pei Sun, Weiyue Wang, Yuning Chai, Gamaleldin Elsayed, Alex Bewley, Xiao Zhang, Cristian Sminchisescu, and Dragomir Anguelov. Rsn: Range sparse net for efficient, accurate lidar 3d object detection. In *CVPR*, pages 5725–5734, 2021. 2

[49] Hao Tian, Yuntao Chen, Jifeng Dai, Zhaoxiang Zhang, and Xizhou Zhu. Unsupervised object detection with lidar clues. In *CVPR*, 2021. 2

[50] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. 2022. 2

[51] Dominic Zeng Wang and Ingmar Posner. Voting for voting in online point cloud object detection. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015. 2

[52] Yue Wang, Alireza Fathi, Abhijit Kundu, David Ross, Caroline Pantofaru, Thomas Funkhouser, and Justin Solomon. Pillar-based object detection for autonomous driving. In *ECCV*, 2020. 2

[53] Kelvin Wong, Shenlong Wang, Mengye Ren, Ming Liang, and Raquel Urtasun. Identifying unknown instances for autonomous driving. In *CoRL*, 2020. 2

[54] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *CVPR*, 2018. 2

[55] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *CVPR*, 2020. 2

[56] M. Ye, S. Xu, and T. Cao. Hvnet: Hybrid voxel network for lidar based 3d object detection. In *CVPR*, 2020. 2

[57] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *CVPR*, 2022. 2

[58] Wu Zheng, Weiliang Tang, Li Jiang, and Chi-Wing Fu. Se-ssd: Self-ensembling single-stage object detector from point cloud. In *CVPR*, 2021. 2

[59] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *CVPR*, 2022. 1, 2

[60] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Phillip Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. 2022. 2

[61] Yin Zhou, Pei Sun, Yu Zhang, Dragomir Anguelov, Jiyang Gao, Tom Ouyang, James Guo, Jiquan Ngiam, and Vijay Vasudevan. End-to-end multi-view fusion for 3d object detection in lidar point clouds. In *CoRL*, 2020. 2, 5

[62] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, 2018. 1, 2