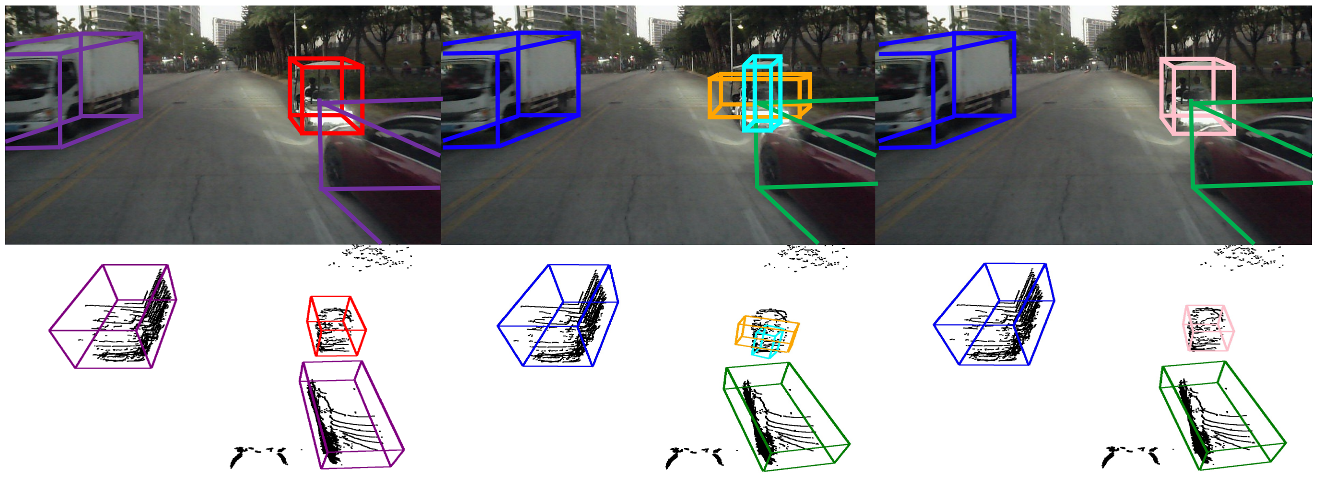
Open-set 3D Object Detection

Anonymous 3DV submission

Paper ID 32



(a) Ground truth (b) Closed-set 3D object detection (c) Open-set 3D object detection

🟪 GT of known (truck, car) 🟥 GT of unknown (golf cart) 🟧 Predicted cyclist 🟦 Predicted pedestrian

🟦 Predicted truck 🟩 Predicted car ⬜ Predicted unknown

Figure 1. Illustration of our proposed open-set 3D object detection. GT means ground truth. Classical closed-set object detection can only predict the classes involved during training, so it cannot handle unknown classes, often regarding them as known classes by mistake. The golf cart is not included in the training dataset, and the closed-set detection predicts these points as the pedestrian and cyclist (middle). By contrast, our proposed open-set object detection classifies these points as an unknown class, and gives the accurate bounding box (right).

Abstract

3D object detection has been wildly studied in recent years, especially for robot perception systems. However, existing 3D object detection is under a closed-set condition, meaning that the network can only output boxes of trained classes. Unfortunately, this closed-set condition is not robust enough for practical use, as it will identify unknown objects as known by mistake. Therefore, in this paper, we propose an open-set 3D object detector, which aims to (1) identify known objects, like the closed-set detection, and (2) identify unknown objects and give their accurate bounding boxes. Specifically, we divide the open-set 3D object detection problem into two steps: (1) finding out the regions containing the unknown objects with high probability and (2) enclosing the points of these regions with proper bounding boxes. The first step is solved by the finding that unknown objects are often classified as known objects with low confidence, and we show that the Euclidean distance sum based on metric learning is a better confidence score than the naive softmax probability to differentiate unknown objects from known objects. On this basis, unsupervised clustering is used to refine the bounding boxes of unknown objects. The proposed method combining metric learning and unsupervised clustering is called the MLUC network. Our experiments show that our MLUC network achieves state-of-the-art performance and can identify both known and unknown objects as expected.

1. Introduction

3D object detection plays an important role in many perception systems, such as autonomous driving and robotics.

LIDAR is a popular sensor to obtain the 3D point cloud for 3D object detection due to its robustness to the environment. Therefore, various deep learning-based methods [1, 2, 3, 4, 5] have been proposed to tackle this point cloud object detection problem. However, classical 3D object detection operates under a closed-set condition, meaning the network only outputs boxes of trained classes. Such a closed-set system will wrongly assign labels of known classes to unknown objects, which could have disastrous consequences in real-world applications [6]. Therefore, a method for 3D object detection with an open-set condition is needed, as illustrated in Fig. 1.

Open-set classification and semantic segmentation tasks are the foundations of the open-set object detection task. Two mainstream approaches to solve the open-set image-level and pixel-level classification problems are uncertainty estimation-based methods [7, 8, 9] and generative-based methods [10, 11, 12]. However, these methods cannot be adapted to the open-set object detection directly as classification tasks do not have to consider whether an object exists or the location of the object. Recently, open-set 2D object detection has been systematically formulated [13], and a dropout sampling-based method [14] and prototypical learning-based method [15] have been proposed to detect unknown objects. The only previous research that operates under the open-set 3D condition was proposed by Kelvin *et al*. [16], who use class-agnostic embeddings to cluster unknown objects to solve the open-set 3D instance segmentation task. Inspired by this open-set 2D object detection and 3D instance segmentation, we propose the first open-set 3D object detector, which is able to identify both known and unknown objects in 3D space, using weaker supervision (bounding boxes) instead of point-level annotations.

Towards the goal of recognizing unknown objects in the LIDAR point cloud, we firstly propose a naive open-set 3D object detector and analyze the difficulty of the open-set 3D object detection task. We find that the naive detector cannot handle the task well, as it mis-classifies known objects as unknown objects, generates lots of false positives of unknown objects, and places inaccurate bounding boxes for unknown objects. To solve these problems, we propose a novel perception system: the Metric learning with Unsupervised Clustering (*MLUC*) network. Specifically, we use metric learning to identify boxes with low confidence scores, and regard these regions as having a high probability of containing unknown objects. On top of this, an unsupervised clustering algorithm is used to generate the precise bounding boxes of unknown objects. In summary, our contributions are the following:

- We are the first to introduce open-set 3D object detection task, which is more suitable than closed-set for real-world applications such as autonomous driving and robotics.
- We analyze the shortcomings of the naive open-set 3D object detector, which straightforwardly extends the closed-set 3D detector to the open-set task.
- To solve the problems of the naive open-set 3D object detector, we develop the MLUC network, which combines metric learning and unsupervised clustering to identify both known and unknown 3D objects. We show that our MLUC network achieves state-of-the-art performance compared with other baselines.

2. Related Work

2.1. Closed-set 3D Object Detection

Classical closed-set 3D object detection methods can be divided into two types: grid-based methods and point-based methods.

Grid-based methods transform irregular point data to regular grids so that the data can be processed by a 2D or 3D convolutional neural network (CNN). MV3D [17] firstly projects the 3D point data to 2D bird's-eye-view grids and then applies a 2D object detection method to generate bounding boxes. Following MV3D, more efficient frameworks with a bird's-eye-view representation are proposed [18, 19]. VoxelNet [2] divides the point clouds into 3D voxels and applies a 3D CNN to extract features, and SECOND [1] introduces 3D sparse convolution [20] for efficient processing.

Point-based methods mostly rely on PointNet [3] and its variants [4, 21]. PointRCNN [22] is a typical two-stage point-based 3D object detector, while 3DSSD [23] introduces F-FPS and is the first one-stage point-based 3D object detector.

These closed-set 3D object detection methods achieve remarkable performance on autonomous driving datasets, such as the KITTI dataset [24] and Waymo open dataset [25]. However, they operate under the closed-set condition, and cannot predict unknown objects, which is different to the case in the open world.

2.2. Open-set 2D Object Detection

Dhamija *et al*. [13] systematically study the open-set performance of common 2D object detectors [26, 27, 28], and they find that unknown objects from the open world end up being incorrectly detected as known objects, often with very high confidence. Miller *et al*. [14] use Monte Carlo Dropout [7] sampling to measure the uncertainty scores of detected boxes, and regard high uncertainty boxes as unknown objects. Recently, Joseph *et al*. [15] adopted prototypical learning with contrastive clustering and an energy-based identifier to detect unknown objects. We draw inspiration from these open-set 2D object detection methods, to propose our MLUC network to address the open-set object detection problem in 3D space.

2.3. Open-set 3D Instance Segmentation

The open-set 3D instance segmentation (OSIS) network, proposed by Kelvin *et al*. [16] is the only previous research to focus on perception under open-set 3D space conditions. The authors use the embedding head to extract the class-agnostic embeddings for each point as well as prototypes for each instance of the known class. During inference, the prototypes collectively filter out points from the known classes whose embeddings are close enough to the prototypes. Then, the remaining points not assigned to any prototypes are clustered into instances of the unknown class based on their embeddings. On top of their OSIS network, we use bounding boxes to enclose the unknown instances, so that this modified OSIS method can be one of the baselines of open-set 3D object detection task.

3. Open-set 3D Object Detection

In this section, we formulate the definition of open-set 3D object detection. We define the class set of the training dataset as $\mathcal{D}^{train} = \{1, 2, ..., C\}$. In classical closed-set 3D object detection, the class set of the test dataset is the same as that of the training dataset, meaning that $\mathcal{D}^{test} = \mathcal{D}^{train} = \{1, 2, ..., C\}$. However, in the open-set 3D object detection, we assume that the test dataset contains more categories than the training dataset, which is more close to real applications. Therefore, under the open-set condition, $\mathcal{D}^{test} = \{1, 2, ..., C, C+1, C+2, ..., C+n\} \supseteq \mathcal{D}^{train}$. The label of the training dataset is the bounding boxes set $\mathcal{B}^{train} = \{\mathbf{b}_i | i = 1, 2, ..., m\}$, where m refers to the total number of bounding boxes in the training set. Each box can be represented by $\mathbf{b}_i = [c, x, y, z, w, l, h, \theta]$, where $c \in \mathcal{D}^{train}, x, y, z, w, l, h, \theta$ refer to the class, center coordinates, size and rotational angles along z axis of the box. The label of the test set is similar to that of the training set, except that the test set has a larger class set space.

The purpose of the open-set 3D object detection is to train a neural network to not only identify trained C object classes, but also assign the 'unknown' label to those classes not encountered during training that are $\{C+1, C+2, ..., C+n\}$, as well as use correct bounding boxes to enclose the corresponding points, as shown in Fig. 1.

4. Naive Open-set 3D Object Detector

Suppose the output of the classical closed-set 3D object detector is $\hat{\mathcal{B}}^{close} = \left\{\hat{\mathbf{b}}_i | i = 1, 2, ..., \hat{m}\right\}$, and each $\hat{\mathbf{b}}_i = [\hat{c}, \hat{s}, \hat{x}, \hat{y}, \hat{z}, \hat{w}, \hat{l}, \hat{h}, \hat{\theta}]$, where $\hat{c}, \hat{s}, \hat{x}, \hat{y}, \hat{z}, \hat{w}, \hat{l}, \hat{h}, \hat{\theta}$ refer to predicted class, confidence score, center coordinates, size and rotational angle of the predicted box. Specifically, the confidence score $\hat{s}$ is determined by:

$$\hat{s} = max\left\{\hat{p}_i | i = 1, 2, ..., C\right\}, \qquad (1)$$

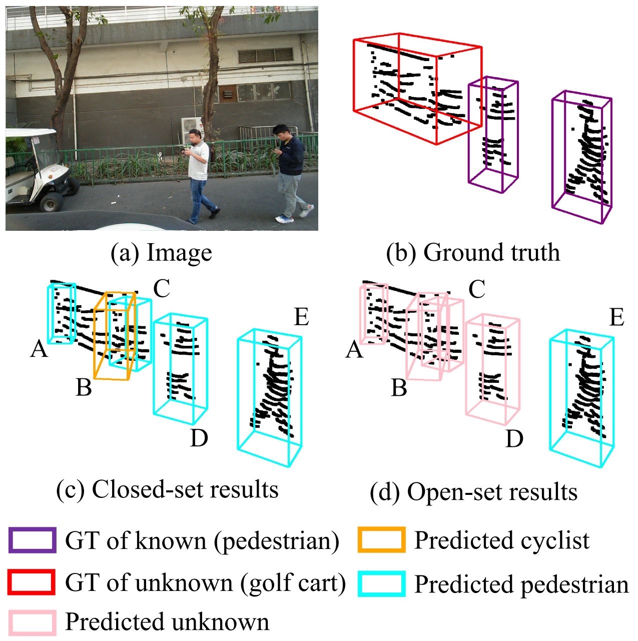


Figure 2. Prediction example of the naive open-set 3D object detector. Objects A, B, C, and D have low confidence scores, so they are classified as unknown objects by the naive open-set 3D object detector.

where p_i refers to the softmax probability of a certain class.

The most natural way to solve the open-set 3D object detection problem is to regard those boxes whose confidence scores are smaller than a threshold ($\hat{s} < \lambda_{naive}$) as unknown objects, which we call the naive open-set 3D object detector. We use this method as one of the baselines. One visualization of the naive open-set 3D object detector is shown in Fig. 2. We can see that although this method is straightforward and simple enough, it has several problems.

Mis-classifies known objects as unknown objects. It is possible to mis-classify some known objects as unknowns [7, 29, 9]. The reasons include two folds. On one hand, the prediction of unknowns highly depends on closed-set predictions. Since the whole probability space is divided for known objects and there is no remaining space for unknown objects, a foreground object might get mis-classified as unknowns due to a relatively low confidence score or high threshold. On the other hand, only one class, which owns the maximum softmax probability, gets considered in estimating unknown objects. The naive method does not exploit the information of the other classes. In Fig. 2, the known object D is classified as unknown by mistake.

Generates many false positive unknown objects. In our experiments, we find that the 3D detector tends to classify unknown objects to multiple known objects. For example, the golf cart in Fig. 2 is divided into two pedestrians and one cyclist. If we regard all of these objects as unknown objects, more false positive unknown objects will

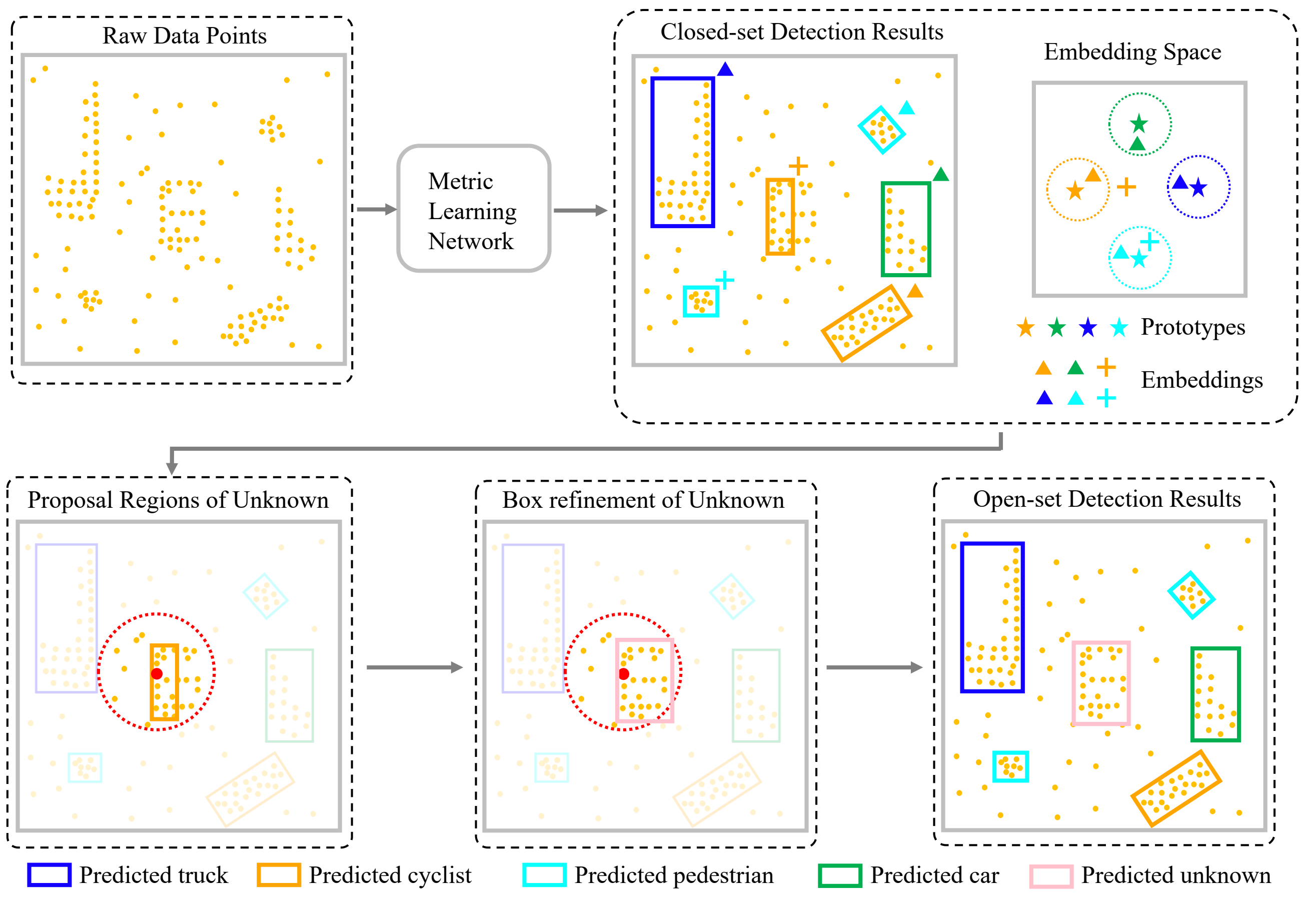


Figure 3. The pipeline of our MLUC network. A metric learning network is applied for the raw data points to obtain the closed-set detection boxes and corresponding embeddings. Those boxes whose embedding is located at the center of the embedding space are regarded as low-confidence-score boxes. We randomly pick one point from each low-confidence-score box and obtain the proposal regions of unknown objects using picked points as centers. Then unsupervised clustering is used to refine the boxes of unknown objects. In this way, our MLUC network can identify both known and unknown objects to fulfill the open-set 3D object detection task.

be induced. If we filter these bounding boxes using non-maximum-suppression (NMS) with negative softmax probability score, it will make the bounding boxes inaccurate, which will be discussed in the next paragraph.

Generates inaccurate bounding boxes for unknown objects. The size of the predicted boxes from the naive method is very close to the pre-defined anchor size. Therefore, if we only change the labels of some boxes to 'unknown' but keep the size unchanged, the corresponding boxes cannot enclose the points of unknown objects very well. For example, in Fig. 2, the boxes of the pedestrians and cyclists do not match the size of the golf cart. This is why open-set object detection is more difficult than open-set classification.

5. MLUC Network

To solve the problems of the naive open-set 3D object detector, we propose the MLUC network. We use the Euclidean distance sum in metric space to measure the uncertainty, and we show that it is a better confidence score compared with the naive maximum softmax probability. Therefore, metric learning helps us find more reliable regions containing unknown objects. On this basis, unsupervised clustering is applied on each proposal region of unknown objects, so that the bounding boxes of unknowns get well refined. Finally, bounding boxes generated for unknown objects will be processed by NMS to filter out overlapping results. In this way, we can obtain more accurate bounding boxes for unknown objects and suppress false positives. The pipeline of the MLUC network is shown in Fig. 3.

5.1. Deep Metric Learning

The classification branch of a typical object detection network is composed of two parts: a feature extractor to obtain the high-dimensional features and a classifier to generate the decision hyper-plane. However, this feature ex-

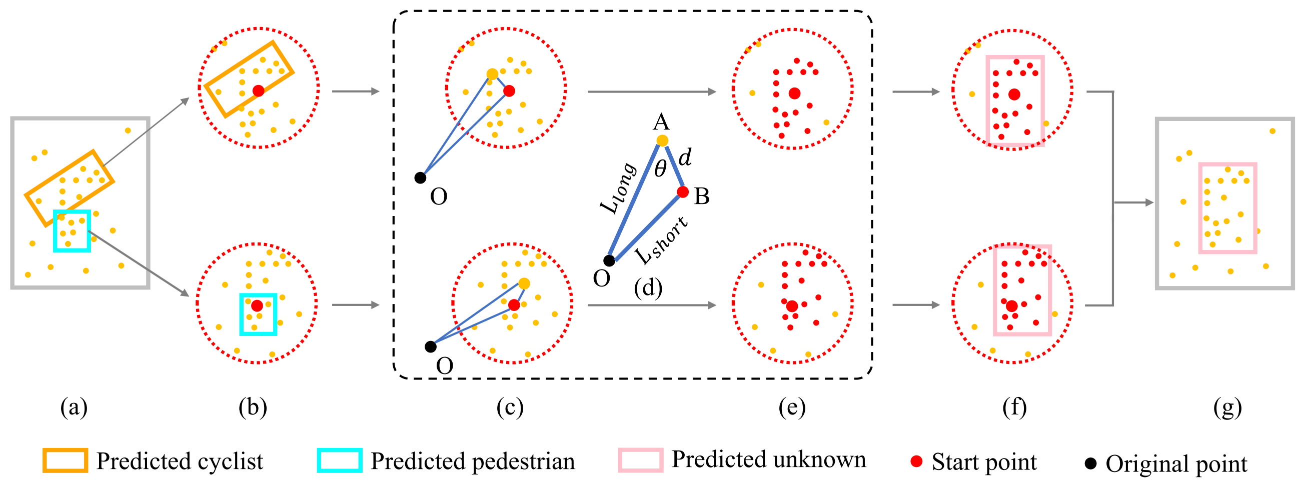


Figure 4. Illustration of our unsupervised clustering algorithm. (a): Closed-set detection results. (b): Proposal regions of unknown objects. (c): Depth clustering from the start point. (d) Depth clustering mechanism. (e) Depth clustering results. (f) Using a new bounding box to enclose the clustering results. (g) Predicted unknown objects.

tractor and classifier structure is not suitable for unknown objects detection, as the network assumes that all feature space is assigned for known objects, and there is no space left for unknown objects [30]. Therefore, we replace the classifier by the Euclidean distance representation with all prototypes $\mathcal{M}_{in} = \{\mathbf{m}_t \in \mathbb{R}^{1\times C} | t \in \{1, 2, ..., C\}\}$, where $\mathbf{m}_t$ refers to the prototype of class $\mathcal{D}_t^{train}$. The feature extractor $f(\mathbf{X})$ is designed to learn the embedding vector in the metric space of each input box $\mathbf{X}$. In this way, the probability of box $\mathbf{X}$ classified as the class $\mathcal{D}_t^{train}$ is formulated as:

$$p_t(\mathbf{X}) = \frac{exp(-\|f(\mathbf{X}) - \mathbf{m}_t\|^2)}{\sum_{t'=1}^{C} exp(-\|f(\mathbf{X}) - \mathbf{m}_{t'}\|^2)}. \tag{2}$$

Then we can apply this Euclidean distance-based probability to define the loss function:

$$\mathcal{L} = \sum -log(\frac{exp(-\|f(\mathbf{X}) - \mathbf{m}_\mathbf{Y}\|^2)}{\sum_{k=1}^{C} exp(-\|f(\mathbf{X}) - \mathbf{m}_k\|^2)}), \tag{3}$$

where $\mathbf{Y}$ is the ground truth class of the input box $\mathbf{X}$. This loss function has two effects on the learned embedding vectors: (1) The embedding vector will be attracted by the prototype of the same class, which is affected by the numerator of the loss function. (2) The embedding vector will be repelled by the prototypes of other classes, which is affected by the denominator of the loss function. In this way, the embeddings of known objects will be close to the corresponding prototypes of the same class, while the embeddings of unknown objects will be distributed in the center of the embedding space as they are repelled by all known prototypes, as illustrated in Fig. 3.

Based on this metric learning framework, we propose to use the Euclidean distance sum (EDS) to measure the uncertainty. The EDS is defined as:

$$EDS = \sum_{t=1}^{C} \|f(\mathbf{X}) - \mathbf{m}_t\|^2. \tag{4}$$

Unknown objects are supposed to have smaller a EDS as they are in the center of the embedding space. Compared to the maximum softmax probability, this EDS considers all classes explicitly. The boxes whose EDS score is smaller than a threshold $(EDS < \lambda_{EDS})$ are considered to contain unknown objects. As EDS is class-independent, all prototypes of known classes are designed to be evenly distributed in the embedding space and stable during training. We define the prototype in a one-hot vector form: only the t^{th} element of $\mathbf{m}_t$ is C, while others remain zero, where $t \in \{1, 2, ..., C\}$ [31].

5.2. Unsupervised Clustering

After we obtain the low-confidence-score bounding boxes, we regard them as the proposal regions of unknown objects and apply unsupervised clustering to refine them. Fig. 4 shows the process of our unsupervised clustering algorithm.

The first step is to find the proposal regions of unknown objects based on the obtained EDS of each predicted box. Those boxes whose EDS is smaller than a threshold λ_{EDS} are considered to have high a probability of containing unknown objects. Then we randomly pick one point from each low EDS box, and set up the proposal regions of unknown objects with the cylinders whose centers are picked points and radius is r, as shown in Fig. 4 (a) and (b).

The second step is to cluster points of unknown objects in the proposal regions. We execute the depth clustering [32] algorithm to find the points of unknown objects.

The principle of depth clustering is illustrated in Fig. 4 (d). O is the original point, which is also the location of the LIDAR. A and B are two points for which it is to be decided whether they belong to the same object or not. The point which is far away from O is A and the other is B. If θ is smaller than a threshold λ_θ, A and B are considered to belong to one object. We start iterations from the center of the proposal regions, and the points which are seen to belong to the same object with the start point will be the start point of the next iteration, until all points in the proposal region are decided. The red points in Fig. 4 (e) are considered to be the points of one unknown object.

The third step is to use a tight bounding box to enclose the points of unknown objects, and then use NMS to post-process all obtained bounding boxes of unknown objects, with larger bounding boxes having higher priority, as shown in Fig. 4 (f) and (g).

We summarize our unsupervised clustering method in Algorithm 1. $\hat{\mathcal{B}}^{close} = \left\{\hat{\mathfrak{b}}_i | i = 1, 2, ..., \hat{m}\right\}$ represents the output closed-set bounding boxes of the metric learning, and $\hat{\mathfrak{b}}_i[\hat{s}]$ is the corresponding EDS value.

6. Experiments

6.1. Experimental Setup

Datasets: We evaluate the performance of our MLUC network on two datasets.

UDI dataset is a self-driving dataset with LIDAR point clouds collected from an industrial park. Six classes including car, pedestrian, cyclist, truck, golf cart, and forklift, are annotated in the UDI dataset. In our experiments, four classes, car, pedestrian, cyclist, and truck, are treated as known objects, while the other two classes, golf cart, and forklift, are regarded as unknown objects and not involved during training. There are a total 200k known objects in the training set and 12k known objects as well as 600 unknown objects in the test set.

KITTI dataset [24] is one of the most popular open-source datasets of 3D object detection for autonomous driving. Three common classes, car, pedestrian, and cyclist, are used as known objects, while the van and truck are used as the unknown classes and not included during training. There are 8690 training objects and 6100 test objects in our experiments, with the test objects composed of 4845 known objects and 1255 unknown objects.

Evaluation metrics: For known objects, we use the 3D *mean average precision* (mAP_{known}) to evaluate the performance, while for unknown objects, we report the $recall_{unknown}$ and 3D *average precision* ($AP_{unknown}$). Then, the metric mAP_{harm} is used to comprehensively evaluate the overall performance [33]:

$$mAP_{harm} = \frac{2 * mAP_{known} * AP_{unknown}}{mAP_{known} + AP_{unknown}}, \quad (5)$$

Algorithm 1: Unsupervised Clustering Method

```
Input: $\hat{\mathcal{B}}^{close} = \left\{\hat{\mathfrak{b}}_i | i = 1, 2, ..., \hat{m}\right\}$
Output: Bounding boxes of unknown objects $\hat{\mathcal{B}}^{unknown}$
1  $P = [\,]$;
2  for $\hat{\mathfrak{b}}_i$ in $\hat{\mathcal{B}}^{close}$ do
3      if $\hat{\mathfrak{b}}_i[\hat{s}] < \lambda_{EDS}$ then
4          randomly pick one point $p$ inside $\hat{\mathfrak{b}}_i$;
5          $P$.append($p$);
6  end
7  for $p$ in $P$ do
8      $Q = [p]$; $\bar{Q} = R = [\,]$;
9      while $Q$ is not empty do
10         $t$=$Q$.top(); $Q$.pop(); $\bar{Q}$.append($t$); $R$.append($t$);
11         $N_t$ = the neighbor point set of $t$;
12         for $s$ in $N_t$ do
13             if $s$ in $\bar{Q}$ or $sp > r$ then
14                 continue
15             $L_{long} = max(Os, Ot)$;
16             $L_{short} = min(Os, Ot)$;
17             $d = st$;
18             $\theta = arccos(\frac{L_{long}^2 + d^2 - L_{short}^2}{2dL_{long}})$;
19             if $\theta < \lambda_\theta$ then
20                 $Q$.append($s$)
21             else
22                 $\bar{Q}$.append($s$)
23             end
24         end
25     end
26     $\hat{\mathfrak{b}}^{unknown}$ = the bounding box enclosing $R$ tightly;
27     $\hat{\mathcal{B}}^{unknown}$.append($\hat{\mathfrak{b}}^{unknown}$);
28 end
29 $\hat{\mathcal{B}}^{unknown}$=NMS($\hat{\mathcal{B}}^{unknown}$);
```

Baselines: As we are the first to propose the open-set 3D object detection task, there are no formal baselines from other research. Therefore, we adopt the naive open-set 3D object detector and modified OSIS network, which have been discussed in Sections 4 and 2.3 respectively, as our baselines.

Implementation details: We adopt *SECOND* [1] as our 3D object detector for both the UDI and KITTI dataset.

For the UDI dataset, the detection range of the point clouds is $[-51.2, 51.2]$ m for both the X and Y axis, and $[-5, 3]$ m for the Z axis. We use the ADAM [34] optimizer with learning rate 0.003 and momentum 0.9. The network is trained on one NVIDIA 2080Ti for 20 epochs with a batch size 4. The IoU threshold during evaluation is 0.5 for car,

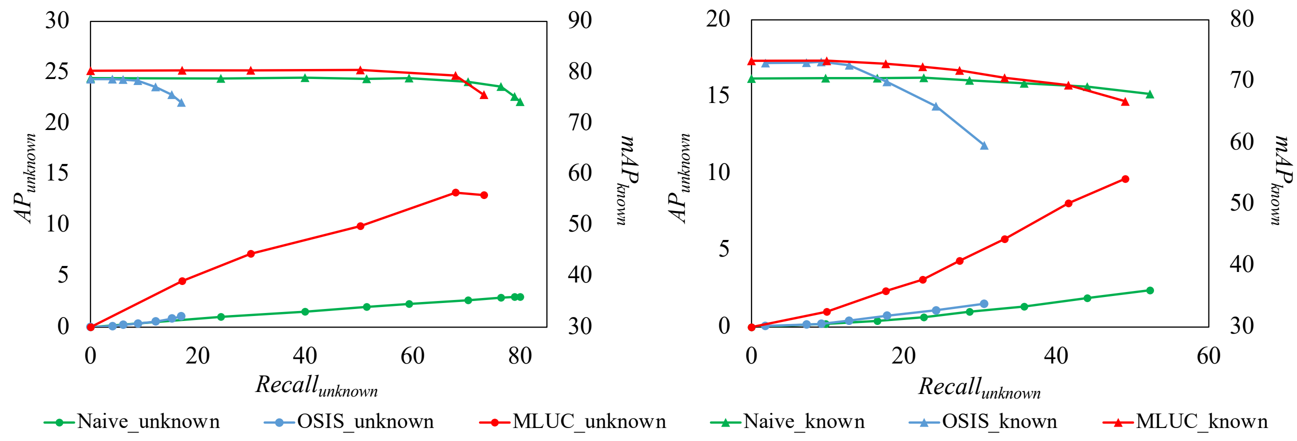


Figure 5. The performance of Naive, OSIS and MLUC methods on UDI (left) and KITTI (right) dataset. For each confidence score threshold, we can obtain a data point $(Recall_{unknown}, AP_{unknown}, mAP_{known})$. The two graphs contain the results of $AP_{unknown}$ and mAP_{known} with regard to $Recall_{unknown}$.

pedestrian, and cyclist, 0.7 for truck, and 0.1 for the unknown objects, i.e., golf cart and forklift. The evaluation difficulty is easy.

For the KITTI dataset, the detection range is $[0, 70.4]\ m$ for the X axis, $[-40, 40]\ m$ for the Y axis, and $[-3, 1]\ m$ for the Z axis. The learning process setting is the same as the UDI dataset except that we train the model for 80 epochs on the KITTI dataset. The IoU threshold during evaluation is 0.7 for car and truck, 0.5 for pedestrian and cyclist, and 0.1 for the unknown objects, i.e., van and truck. The evaluation difficulty is moderate.

The λ_θ in the unsupervised clustering is $65°$, and r is $4\ m$ and $5\ m$ for the UDI and KITTI dataset respectively.

6.2. Results

The specific open-set 3D object detection results are related to the confidence thresholds (λ_{naive} for Naive method and OSIS method, and λ_{EDS} for MLUC method). Therefore, we plot the results for the UDI and KITTI dataset of $AP_{unknown}$ and mAP_{known} with regard to $Recall_{unknown}$ in Fig. 5, where each data point is the result of one specific threshold. Fig. 5 shows the mAP_{known} reduces with the growth of $Recall_{unknown}$, because more known objects are regarded as unknown objects with the growth of the confidence score threshold. So we only record the points whose mAP_{known} is not reduced by 10% to ensure the high performance of the known classes. Then we pick the result with the best mAP_{harm} to represent the optimal performance of each method, and these are recorded in Table 2.

Fig. 5 and Table 2 show that our MLUC method has the best $AP_{unknown}$ and mAP_{harm} compared with the Naive method and OSIS method. From Fig. 5, we also find that the Naive method and MLUC method have a larger $Recall_{unknown}$ than the OSIS method when mAP_{known} decreases within 10%. This is because an unknown object is often classified as several overlapping known objects, as shown in Fig. 1 and 2, and in the OSIS method, points will only be considered to be part of unknown objects when all boxes that include them have higher confidence scores than the threshold. In contrast, the Naive method and MLUC method only require the lowest score of these overlapping boxes to be higher than the threshold. Two visualization results are shown in Fig. 6.

To validate the effectiveness of the metric learning and unsupervised clustering in the MLUC method, we conduct ablation experiments and show the results in Table 1. It shows both metric learning and unsupervised clustering make a contribution to the better open-set detection performance.

UDI dataset				
ML	UC	mAP_{known}	$AP_{unknown}$	mAP_{harm}
×	×	75.3	3.0	5.7
✓	×	78.7	8.3	15.1
✓	✓	**79.4**	**13.2**	**22.6**
KITTI dataset				
ML	UC	mAP_{known}	$AP_{unknown}$	mAP_{harm}
×	×	63.8	3.9	7.3
✓	×	66.1	5.8	10.6
✓	✓	**66.8**	**9.7**	**16.9**

Table 1. Ablation experiment results of MLUC method. ML and UC refer to metric learning and unsupervised clustering respectively.

7. Conclusion

In this paper, we propose open-set 3D object detection to identify both known and unknown objects in 3D LIDAR point clouds. We show that our metric learning

Dataset	UDI			KITTI		
Methods	mAP_{known}	$AP_{unknown}$	mAP_{harm}	mAP_{known}	$AP_{unknown}$	mAP_{harm}
Closed-set	78.9	0	0	70.5	0	0
Supervised	78.4	61.7	69.1	76.6	80.6	78.5
Naive	75.3	3.0	5.7	63.8	3.9	7.3
OSIS	74.1	1.1	2.1	65.9	1.1	2.2
MLUC	**79.4**	**13.2**	**22.6**	**66.8**	**9.7**	**16.9**

Table 2. Optimal performance of open-set 3D object detection. Supervised method means we include the unknown classes in the training set and retrain the model, so it is the upper bound of the open-set detection performance. We show that our MLUC method achieves the best performance among all baselines.

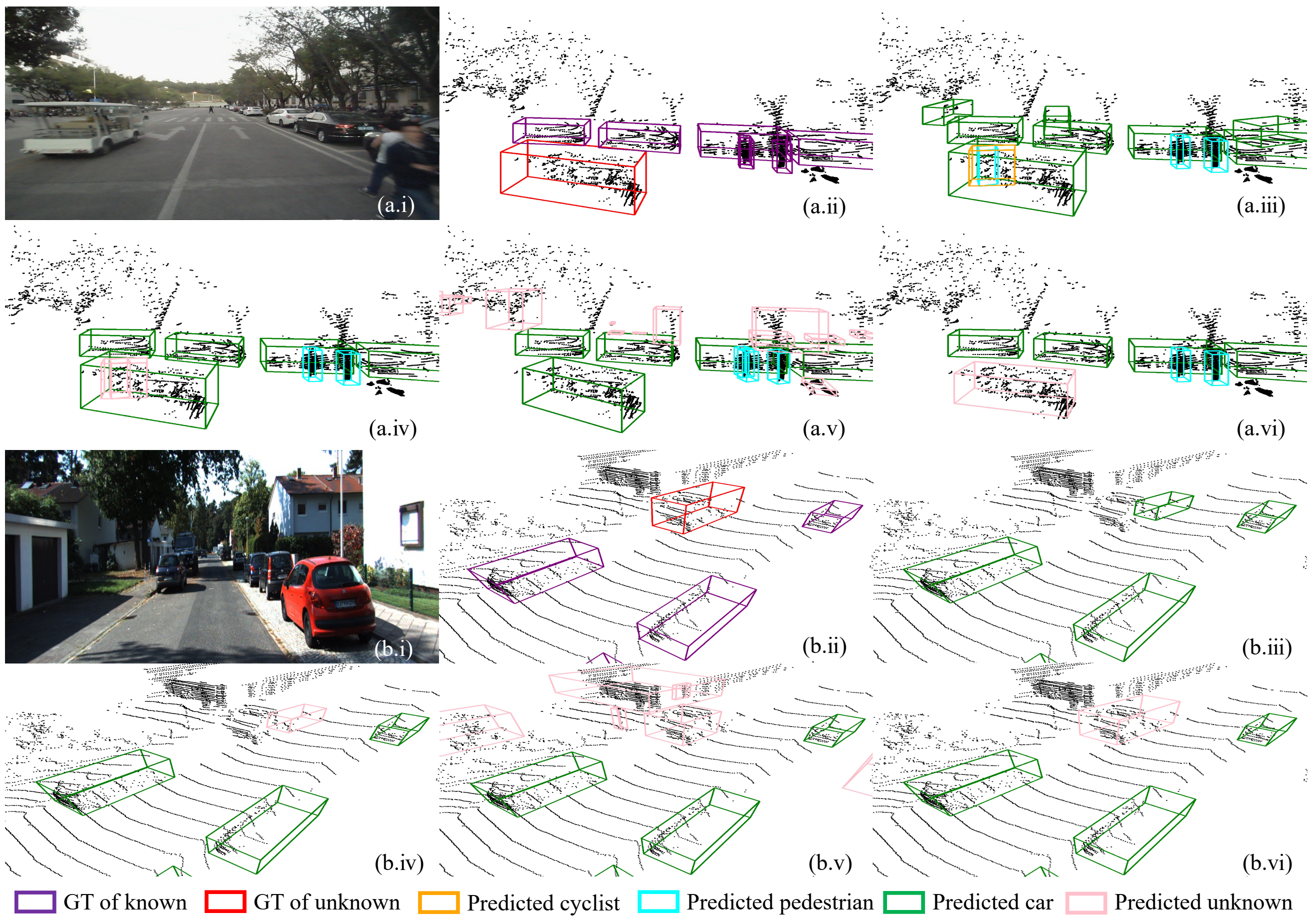


Figure 6. Visualization of some qualitative results. (a) UDI dataset; (b) KITTI dataset; (i) Image; (ii) Ground truth; (iii) Closed-set results; (iv) Naive method; (v) OSIS method; (vi) MLUC method. Golf cart and truck are unknown objects for UDI and KITTI dataset respectively. (a.iii) and (b.iii) show that unknown objects are classified as known objects by mistake. The Naive method cannot provide accurate bounding boxes for unknown objects according to (a.iv) and (b.iv). The OSIS method regards other stuff things including trees and walls as unknown objects, as shown in (a.v) and (b.v). Our MLUC method can provide accurate locations and boxes of unknown objects, as indicated in (a.vi) and (b.vi).

with unsupervised clustering (MLUC) method achieves the best performance compared with the Naive method and OSIS method on the UDI and KITTI datasets. The performance gap between our MLUC method and supervised upper bound indicates that this open-set 3D object detection problem can be further studied. We hope our work can draw more researchers to contribute to this practically valuable research direction.

References

[1] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 2, 6

[2] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4490–4499, 2018. 2

[3] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660, 2017. 2

[4] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 30. Curran Associates, Inc., 2017. 2

[5] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–779, 2019. 2

[6] Darko Bozhinoski, Davide Di Ruscio, Ivano Malavolta, Patrizio Pelliccione, and Ivica Crnkovic. Safety for mobile robotic systems: A systematic mapping study from a software engineering perspective. *Journal of Systems and Software*, 151:150–179, 2019. 2

[7] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*, pages 1050–1059, 2016. 2, 3

[8] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations (ICLR)*, 2017. 2

[9] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6402–6413, 2017. 2, 3

[10] Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. In *International MICCAI Brainlesion Workshop*, pages 161–169, 2018. 2

[11] Krzysztof Lis, Krishna Nakka, Pascal Fua, and Mathieu Salzmann. Detecting the unexpected via image resynthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2152–2161, 2019. 2

[12] Yingda Xia, Yi Zhang, Fengze Liu, Wei Shen, and Alan L Yuille. Synthesize then compare: Detecting failures and anomalies for semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 145–161. Springer, 2020. 2

[13] Akshay Raj Dhamija, Manuel Günther, Jonathan Ventura, and Terrance E. Boult. The overlooked elephant of object detection: Open set. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1010–1019, 2020. 2

[14] Dimity Miller, Lachlan Nicholson, Feras Dayoub, and Niko Sünderhauf. Dropout sampling for robust object detection in open-set conditions. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3243–3249. IEEE, 2018. 2

[15] KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5830–5840, 2021. 2

[16] Kelvin Wong, Shenlong Wang, Mengye Ren, Ming Liang, and Raquel Urtasun. Identifying unknown instances for autonomous driving. In *Conference on Robot Learning (CoRL)*, pages 384–393. PMLR, 2020. 2, 3

[17] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1907–1915, 2017. 2

[18] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7652–7660, 2018. 2

[19] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12697–12705, 2019. 2

[20] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9224–9232, 2018. 2

[21] Teppei Suzuki, Keisuke Ozawa, and Yusuke Sekikawa. Rethinking pointnet embedding for faster and compact model. In *2020 International Conference on 3D Vision (3DV)*, pages 791–800, 2020. 2

[22] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–779, 2019. 2

[23] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11040–11048, 2020. 2

[24] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research (IJRR)*, 32(11):1231–1237, 2013. 2, 6

[25] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2446–2454, 2020. 2

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems (NIPS)*, 28:91–99, 2015. 2

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2980–2988, 2017. 2

[28] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7263–7271, 2017. 2

[29] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. *arXiv preprint arXiv:1911.11132*, 2019. 3

[30] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3474–3482, 2018. 5

[31] Dimity Miller, Niko Sunderhauf, Michael Milford, and Feras Dayoub. Class anchor clustering: A loss for distance-based open set recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3570–3578, 2021. 5

[32] Igor Bogoslavskyi and Cyrill Stachniss. Fast range image-based segmentation of sparse 3d laser scans for online operation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 163–169, 2016. 5

[33] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network for zero- and few-label semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 6

[34] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 6