







Open-world Semantic Segmentation for LIDAR Point Clouds

Jun Cen¹, Peng Yun¹, Shiwei Zhang^{2*}, Junhao Cai¹, Di Luan¹,
Mingqian Tang², Ming Liu¹, and Michael Yu Wang¹

¹ The Hong Kong University of Science and Technology
{jcenaa,pyun,jcaiaq,dluan}@connect.ust.hk {mywang,eelium}@ust.hk

² Alibaba Group
{zhangjin.zsw,mingqian.tmq}@alibaba-inc.com

Abstract. Current methods for LIDAR semantic segmentation are not robust enough for real-world applications, *e.g.*, autonomous driving, since it is *closed-set* and *static*. The closed-set assumption makes the network only able to output labels of trained classes, even for objects never seen before, while a static network cannot update its knowledge base according to what it has seen. Therefore, in this work, we propose the *open-world semantic segmentation* task for LIDAR point clouds, which aims to 1) identify both old and novel classes using open-set semantic segmentation, and 2) gradually incorporate novel objects into the existing knowledge base using incremental learning without forgetting old classes. For this purpose, we propose a **RE**dundAncy **c**lassifier (REAL) framework to provide a general architecture for both the open-set semantic segmentation and incremental learning problems. The experimental results show that REAL can simultaneously achieves state-of-the-art performance in the open-set semantic segmentation task on the SemanticKITTI and nuScenes datasets, and alleviate the catastrophic forgetting problem with a large margin during incremental learning.

Keywords: Open-world Semantic Segmentation, LIDAR Point Clouds, Open-set Semantic Segmentation, Incremental Learning

1 Introduction

3D LIDAR sensors play an important role in the perception system of autonomous vehicles. Semantic segmentation for LIDAR point clouds has grown very fast in recent years [10, 26, 42, 44], benefiting from well-annotated datasets including SemanticKITTI [2, 3, 13] and nuScenes [6]. However, existing methods for LIDAR semantic segmentation are all *closed-set* and *static*. The closed-set network regards all inputs as categories encountered during training, so it will

*Corresponding author.

Code is available at: https://github.com/Jun-CEN/Open_world_3D_semantic_segmentation

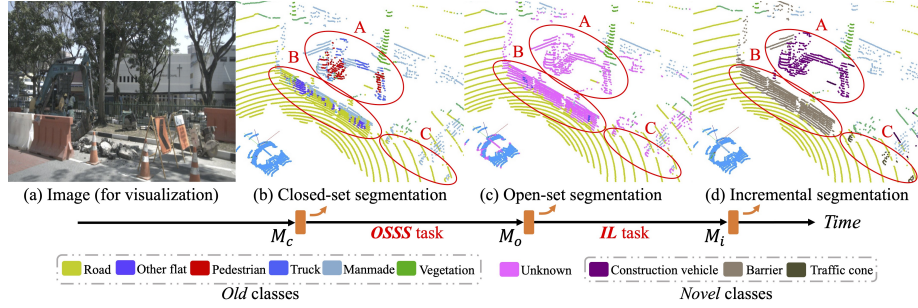


Fig. 1. Closed-set model M_c wrongly assigns the labels of old classes to novel objects (A: construction vehicle is classified as the manmade, truck, and even pedestrian; B: barrier is classified as the road, manmade and other flat; C: traffic cone is classified as the manmade). After open-set semantic segmentation (OSeg) task, the open-set model M_o can identify the novel objects and assign the label *unknown* for them. After incremental learning (IL) task, the model M_i can classify both old and novel classes.

assign the labels of old classes to novel classes by mistake, which may have disastrous consequences in safety-sensitive applications, such as autonomous driving [5]. Meanwhile, the static network is constrained to certain scenarios, as it cannot update itself to adapt to new environments. In addition, training from scratch to adapt to new scenes is extremely time-consuming, and the annotations of old classes are sometimes unavailable, due to privacy constraints.

To solve the *closed-set* and *static* problem, we propose the *open-world semantic segmentation* for LIDAR point clouds, which is composed of two tasks: 1) open-set semantic segmentation (OSeg) to assign the *unknown* label to novel classes as well as to assign the correct labels to old classes, and 2) incremental learning (IL) to gradually incorporate the novel classes into the knowledge base after labellers provide the labels of novel classes. Fig. 1 illustrates an example of open-world semantic segmentation for LIDAR point clouds.

As we are the first to study OSeg task in the 3D LIDAR point cloud domain, we refer to the existing methods in the 2D image domain, which can be divided into two types, generative network-based methods [1, 22, 39] and uncertainty-based methods [12, 15, 19], though none of them can be directly utilized. Generative network-based methods adopt a conditional generative adversarial network (cGAN) [27] to reconstruct the input based on the closed-set prediction results, and assume the novel regions have a larger difference in appearance between the reconstructed input and original input. However, cGAN is not appropriate for reconstruction of the point cloud as all information is determined by the geometry information, *i.e.*, coordinates of points, and cGAN can only reconstruct the channel information, *i.e.*, RGB values, while keeping the geometry information, including coordinates of pixels and the shape of an image, unchanged. The uncertainty-based methods also work poorly as we find the network predicts the novel classes as old classes with high confidence scores, as shown in Fig. 3 (a).

In addition to the challenges of the OSeg task, the catastrophic forgetting of old classes in incremental learning [25] is another problem to solve. Directly fine-tuning the network using only the labels of novel classes will make the network classify everything as novel classes. Thus a method is needed to incrementally learn novel classes while keeping the performance of the old classes.

We find that the closed-set and static properties of the traditional closed-set model is due to the fixed classifier architecture, *i.e.*, one classifier corresponds to one old class. Therefore, we propose a **REdundancy cLassifier (REAL)** framework to provide a dynamic classifier architecture to adapt the model to both the OSeg and IL tasks. For the OSeg task, we add several redundancy classifiers (RCs) on the basis of the original network to predict the probability of the unknown class. Then, during the IL task, several RCs are trained to classify the newly introduced classes, while the remaining RCs are still responsible for the unknown class, as shown in Fig. 2. We provide the training strategies for the OSeg and IL tasks under REAL, based on the unknown object synthesis, predictive distribution calibration, and pseudo label generation. We show the effectiveness of REAL and corresponding training strategies through our comprehensive experiments. In summary, our contributions are three-folds:

- We are the first to define the open-world semantic segmentation problem for LIDAR point clouds, which is composed of OSeg and IL tasks;
- We propose a REAL model to provide a general architecture for both the OSeg and IL tasks, as well as training strategies for each task, based on the unknown objects synthesis, predictive distribution calibration, and pseudo labels generation;
- We construct benchmark and evaluation protocols for OSeg and IL in the 3D LIDAR point cloud domain, based on the SemanticKITTI and nuScenes datasets, to measure the effectiveness of our training strategies under REAL.

2 Related Work

Closed-set LIDAR Semantic Segmentation: Semantic segmentation for LIDAR point clouds can be categorized into point-based and voxel-based methods. Typical point-based methods [16, 33, 38] use PointNet [29] and PointNet++ [30] to directly operate on the LIDAR point cloud. However, they have limited performance due to the varying density and large scale of the LIDAR point cloud. The other type of point-based methods convert the LIDAR point cloud to 2D grids and then apply 2D convolutional operations for semantic segmentation. SqueezeSeg [37] and its alternatives [26, 35, 42] convert the point cloud to a range image or bird’s-eye-view. However, 2D representations inevitably lose some of the 3D topology and geometric information. Cylinder3D [44] is a voxel-based method and it tackles the sparsity and varying density problems of LIDAR point clouds through cylindrical partition and asymmetrical 3D convolutional networks. Cylinder3D achieves state-of-the-art performance on SemanticKITTI [2, 3, 13] and nuScenes [6], so we adopt it as the base architecture.

Incremental learning: Neural networks tend to forget what they have learned when trained with new data, which is called catastrophic forgetting. Some researchers adopt knowledge distillation to overcome this problem. Knowledge-distillation-based methods [8, 20, 21, 28, 32, 40] retain the learnt knowledge by restricting the prediction $p(\hat{y}|x, \theta)$ close to that computed with the optimal parameter of previous tasks $p(\hat{y}|x, \theta_0^*)$. A regularization term, proportional to the distance between these two conditional distributions, is added to the original loss function. In classification, the distance is commonly measured by the Kullback-Leibler (KL) divergence [21, 31]. Shmelkov *et al.* [32] extended the knowledge distillation to the image-based 2D object detection problem and proposed an incremental learning object detector, ILOD. In recent years, there have been follow-up works [8, 20, 28, 40, 41] in 2D/3D object detection and semantic segmentation. The most similar method to the incremental learning part of our work is [23], where they adopted a pseudo-label generation approach to provide supervision of learned knowledge in image-based object detection.

Open-set 2D Classification: There are two trends of open-set 2D classification methods: uncertainty-based methods and generative model-based methods. Maximum softmax probability (MSP) [15] is the baseline of uncertainty-based methods, while Dan *et al.* [14] found that Maximum Logit (MaxLogit) is a better choice than the probability. MC-Dropout [12] and Ensembles [19] are used to approximate Bayesian inference [18, 24], which regards the network from a probabilistic view. Meanwhile, generative-based methods, including SynthCP [39] and DUIR [22], adopt conditional GAN (cGAN) [27] to reconstruct the input, and find the novel regions by comparing the reconstructed input with the original input. However, these methods cannot adapt to the 3D LIDAR point cloud domain directly, as discussed in Sec. 1. [34, 43] propose to use redundancy classifiers (RCs) to directly output the score of the unknown class, and adopt manifold mixup and a sampler based on Stochastic Gradient Langevin Dynamics (SGLD) [36] to approximate the unknown class distribution. We draw inspiration from them, and take a step further by using RCs for both OSeg and IL, as well as developing suitable training strategies for the 3D point cloud domain.

Open-world Classification and Detection: The open-world problem was first proposed by Abhijit *et al.* [4], who argued that the network should be able to deal with a dynamic category set which is practical in the real world. Therefore, they introduced the open-world classification pipeline: first identify both known and unknown images, and then gradually learn to classify unknown images when labels are given. They presented the Nearest Non-Outlier method to manage the open-world classification task. Joseph *et al.* [17] extended the open-world problem to the 2D object detection domain and Jun *et al.* [7] later adopted deep metric learning for open-world semantic segmentation for 2D images. We extend the open-world problem to the 3D LIDAR cloud point domain, and both sub-tasks including OSeg and IL for 3D LIDAR point clouds are not studied yet.

3 Open-world Semantic Segmentation

In this section, we formalise the definition of open-world semantic segmentation for LIDAR point clouds. Let the classes of the training set be called old classes and labeled by positive integers $\mathcal{K}_0 = \{1, 2, \dots, C\} \subset \mathbb{N}^+$. Unlike the traditional closed-set semantic segmentation where the classes of the test set are the same as the training set, some novel classes $\mathcal{U} = \{C + 1, \dots\}$ are involved in the test set in the open-world semantic segmentation problem. Let one LIDAR point cloud sample be formulated as $\mathcal{D} = \{\mathbf{P}, \mathbf{Y}\}$, where $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_M\}$ is the input LIDAR point cloud composed of M points and every point \mathbf{p} is represented by three coordinates $\mathbf{p} = (x, y, z)$. The label $\mathbf{Y} = \{y_1, y_2, \dots, y_M\}$ contains the semantic class for every point, in which $y \in \mathcal{K}_0$ for the training data and $y \in \mathcal{K}_0 \cup \mathcal{U}$ for the test data.

Suppose we already have a model \mathcal{M}_c which is trained under the closed-set condition, so its outputs are within the domain of \mathcal{K}_0 . As discussed in Sec. 1, the open-world semantic segmentation is composed of two tasks: open-set semantic segmentation (OSeg) and incremental learning (IL). For the OSeg task, the model \mathcal{M}_c will be finetuned to \mathcal{M}_o so that it can assign the correct labels for the points of old classes \mathcal{K}_0 , as well as assign the *unknown* label to the points of novel classes \mathcal{U} . For the IL task, the model \mathcal{M}_o will be further finetuned to \mathcal{M}_i when the labels of novel classes \mathcal{K}_n are given, so that its knowledge base is enlarged from \mathcal{K}_0 to $\mathcal{K}_0 \cup \mathcal{K}_n$, where $\mathcal{K}_n = \{C + 1, \dots, C + n\}$. So the classes in \mathcal{K}_n change from *unknown* to *known* for the network. We follow the classical task IL setting [9, 11, 40] that the new given labels only contain the annotation of the novel class \mathcal{K}_n , while the remaining points of old classes \mathcal{K}_0 are not annotated. Additionally, the model after IL \mathcal{M}_i still keeps the open-set property, *i.e.*, assigns the *unknown* label to the remaining novel classes $\mathcal{K}_{rn} = \{C + n + 1, \dots\}$.

4 Methodology

In this section, we introduce our strategies to solve the open-world semantic segmentation problem for LIDAR point clouds. The open-world semantic segmentation is composed of two tasks: OSeg task and IL task. We first introduce the redundancy classifier framework (REAL) in Sec. 4.1, which provides a general network architecture for both the OSeg task and IL task. Then, we introduce the training strategies and inference procedures for the OSeg task and IL task in Sec. 4.2 and Sec. 4.3 respectively.

4.1 Redundancy Classifier Framework (REAL)

The overall view of REAL is shown in Fig. 2. The trained closed-set model \mathcal{M}_c , which can well classify old classes \mathcal{K}_0 , is composed of a feature extractor f and normal classifiers $g_{nm} = \{g_{nm}^1, g_{nm}^2, \dots, g_{nm}^C\}$. For a certain input $\mathbf{P} \in \mathbb{R}^{M \times 3}$, the output of the model \mathcal{M}_c is

$$\mathcal{M}_c(\mathbf{P}) = [y^{old}] = [g_{nm}(f(\mathbf{P}))] \in \mathbb{R}^{M \times C}. \quad (1)$$

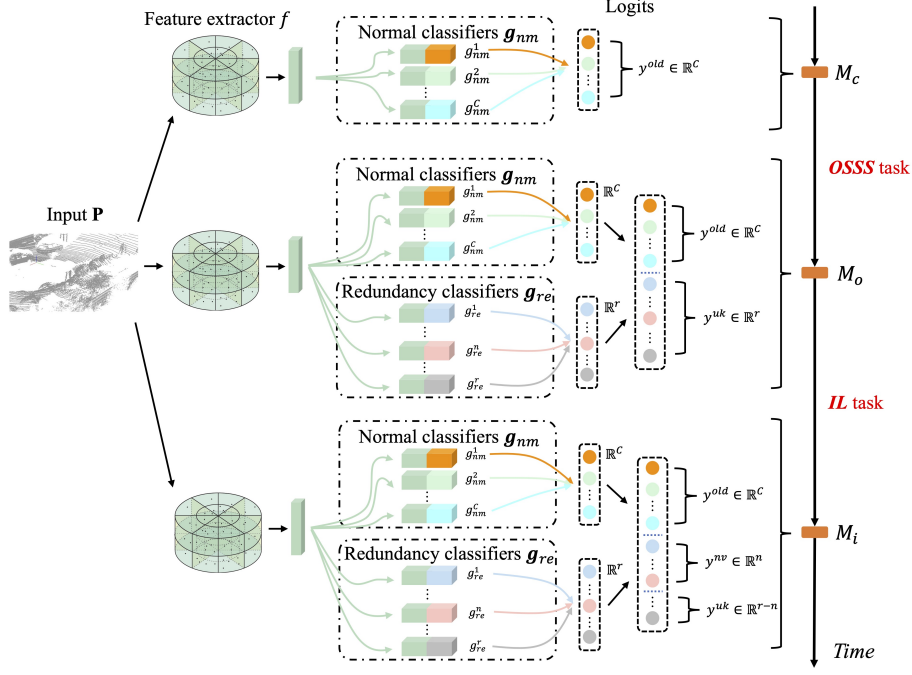


Fig. 2. Redundancy classifier framework (REAL). Closed-set model \mathcal{M}_c can only output logits for old classes y^{old} . Redundancy Classifiers g_{re} are added on top of the original framework in our REAL. All g_{re} in \mathcal{M}_o are used to output the scores y^{uk} for the unknown class. After the IL task, part of g_{re} are used to output logits for the newly introduced classes y^{nv} , while the remaining are still for the unknown class y^{uk} .

OSeg task: The OSeg task is to adapt closed-set model \mathcal{M}_c to open-set model \mathcal{M}_o so that \mathcal{M}_o can identify novel classes \mathcal{U} as *unknown*. To achieve this goal, we add r redundancy classifiers (RCs) $g_{re} = \{g_{re}^1, g_{re}^2, \dots, g_{re}^r\}$ on top of the original feature extractor f , as shown in Fig. 2 \mathcal{M}_o . All RCs in \mathcal{M}_o are used to predict the scores y^{uk} for the unknown class. We let the maximum response of y^{uk} be the score of the unknown class, which is represented by class 0. In this way, the output of the open-set model \mathcal{M}_o is

$$\mathcal{M}_o(\mathbf{P}) = [\max y^{uk}, y^{old}] = [\max g_{re}(f(\mathbf{P})), g_{nm}(f(\mathbf{P}))] \in \mathbb{R}^{M \times (1+C)}. \quad (2)$$

IL task: The IL task is to train open-set model \mathcal{M}_o to \mathcal{M}_i so that newly introduced classes \mathcal{K}_n change from *unknown* to *known*. \mathcal{M}_i is still open-set, *i.e.*, it can classify remaining novel classes \mathcal{K}_{rn} as *unknown*. In this task, among all RCs g_{re} , some of the RCs $g_{re}^{nv} = \{g_{re}^1, g_{re}^2, \dots, g_{re}^n\}$ are used to classify newly introduced classes \mathcal{K}_n , *i.e.*, y^{nv} in Fig. 2 \mathcal{M}_i , and the remaining RCs $g_{re}^{uk} = \{g_{re}^{n+1}, g_{re}^{n+2}, \dots, g_{re}^r\}$ are kept for the unknown class \mathcal{K}_{rn} , *i.e.*, y^{uk} in Fig. 2 \mathcal{M}_i .

In this way, the output of \mathcal{M}_i can be represented as

$$\mathcal{M}_i(\mathbf{P}) = [\max y^{uk}, y^{old}, y^{nv}] = [\max g_{re}^{uk}(f(\mathbf{P})), g_{nm}(f(\mathbf{P})), g_{re}^{nv}(f(\mathbf{P}))]. \quad (3)$$

where $\mathcal{M}_i(\mathbf{P}) \in \mathbb{R}^{M \times (1+C+n)}$.

4.2 Open-set Semantic Segmentation (OSeg)

The OSeg task is to train the closed-set model \mathcal{M}_c to the open-set model \mathcal{M}_o which can identify novel classes \mathcal{U} as *unknown*, as shown in Fig. 1 (c). The network architecture of \mathcal{M}_o is shown in Fig. 2 \mathcal{M}_o . We introduce two training methods including *Unknown Object Synthesis* and *Predictive Distribution Calibration* as well as inference procedure in this section.

Unknown Object Synthesis: We synthesize pseudo unknown objects in the LIDAR point cloud to approximate the distribution of real novel objects. The synthesis process should meet two requirements: 1) the synthesized object should share some invariant basic geometry features with existing objects, such as curved and flat surfaces, so that it can be regarded as an *object* rather than noise and possibly have a similar appearance to real unknown objects; 2) the synthesis process should be as quick as possible.

We find that resizing the existing objects with a proper factor is a simple but effective way to conduct the synthesis process, as it keeps the geometric shape of an object, but the different size determines it is a new object. For instance, a car, truck, bus, and construction vehicle have similar local geometric features, such as the shape of the body and tires, but their size can be different. Therefore, we pick up objects of specific old classes \mathcal{K}_{syn} with a probability p_{syn} and resize them from 0.25 to 0.5 times or 1.5 to 3 times as pseudo unknown objects, such as B in Fig. 4 (c) and (d). In this way, the input \mathbf{P} is divided into two parts: $\mathbf{P} = \mathbf{P}_{syn} \cup \mathbf{P}_{nm}$, where \mathbf{P}_{syn} and \mathbf{P}_{nm} represent the points of synthesized objects and unchanged normal objects respectively. For the points of synthesized objects \mathbf{P}_{syn} , the synthesis loss \mathcal{L}_{syn} is

$$\mathcal{L}_{syn} = \ell(\mathcal{M}(\mathbf{P}_{syn}), \mathbf{0}), \quad (4)$$

where ℓ is the cross-entropy loss. The ground truth labels of synthesized objects are set to be the unknown class 0, so the first term in Eq. 2 is trained to give high scores to objects never seen before.

Predictive Distribution Calibration: We find that in the closed-set prediction, the novel objects are classified as old classes with high probability, as shown in Fig. 3 (a). We intend to alleviate this problem by probability calibration, and the calibrated scores of the unknown class are shown as Fig. 3 (b).

We force every point of old classes to have the largest score on its original class, and have the second largest score on the unknown class [34]. By this design, the network is supposed to output high probability scores on the unknown class for the novel objects as they do not belong to any one of the old classes. Therefore, for the points of unchanged normal objects \mathbf{P}_{nm} , the calibration loss is designed as

$$\mathcal{L}_{cal} = \mathcal{L}_{cal}^{ori} + \lambda_{cal} \mathcal{L}_{cal}^{uk}, \quad (5)$$

where \mathcal{L}_{cal}^{ori} and \mathcal{L}_{cal}^{uk} are defined as

$$\mathcal{L}_{cal}^{ori} = \ell(\mathcal{M}(\mathbf{P}_{nm}), \mathbf{Y}_{nm}), \quad (6)$$

$$\mathcal{L}_{cal}^{uk} = \ell(\mathcal{M}(\mathbf{P}_{nm}) \setminus \mathbf{Y}_{nm}, \mathbf{0}), \quad (7)$$

where \mathbf{Y}_{nm} is the ground truth of \mathbf{P}_{nm} . $\mathcal{M}(\mathbf{P}_{nm}) \setminus \mathbf{Y}_{nm}$ means to remove the response of the corresponding ground truth old class. \mathcal{L}_{cal}^{ori} is to ensure the good closed-set prediction, while \mathcal{L}_{cal}^{uk} is to make every point have the second largest probability on the unknown class.

Loss Function: The overall loss function to train the model \mathcal{M}_c to \mathcal{M}_o is

$$\mathcal{L}^{OSeg} = \mathcal{L}_{cal}^{OSeg} + \lambda_{syn} \mathcal{L}_{syn}^{OSeg}, \quad (8)$$

where \mathcal{L}_{cal}^{OSeg} is determined by Eq. 5, Eq. 6, and Eq. 7, while \mathcal{L}_{syn}^{OSeg} is determined by Eq. 4. All \mathcal{M} in the related terms are \mathcal{M}_o in the OSeg task.

Inference: Both the closed-set and open-set performance of the finetuned model \mathcal{M}_o will be evaluated. For the closed-set prediction, the inference result $\hat{\mathbf{Y}}_{close}$ is defined as

$$\hat{\mathbf{Y}}_{close} = \arg \max_{i=1,2,\dots,C} g_{nm}(f(\mathbf{P})). \quad (9)$$

For the open-set prediction, we have to classify both old classes and the novel class, so the inference result $\hat{\mathbf{Y}}_{open}$ is defined as:

$$\hat{\mathbf{Y}}_{open} = \begin{cases} \arg \max_{i=1,2,\dots,C} g_{nm}(f(\mathbf{P})) & \lambda_{conf} < \lambda_{th} \\ 0 & otherwise, \end{cases} \quad (10)$$

where $\lambda_{conf} = \max g_{re}(f(\mathbf{P}))$ is the confidence score of the unknown class, and λ_{th} is the threshold. The unknown class is represented by class 0.

4.3 Incremental Learning (IL)

The IL task is to train \mathcal{M}_o to \mathcal{M}_i when the labels of novel classes \mathcal{K}_n are available. \mathcal{M}_i can classify both newly introduced classes \mathcal{K}_n and old classes \mathcal{K}_0 , as well as identify remaining novel classes \mathcal{K}_{rn} as *unknown*. The inference example is shown in Fig. 1 (d) and the architecture is shown in Fig. 2 \mathcal{M}_i .

As mentioned in Sec. 3, only the labels of introduced novel classes \mathcal{K}_n are given in this task. Therefore, we divide the unchanged normal points \mathbf{P}_{nm} into

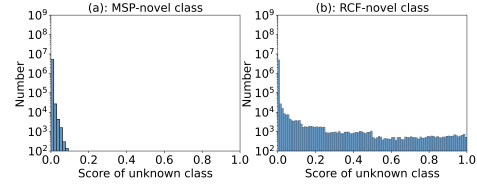


Fig. 3. Distribution of scores of the unknown class for Maximum Softmax Probability (MSP) and our REAL method. The scores of the unknown class for novel classes are low in MSP (a), meaning the closed-set prediction classifies novel classes as old classes with high confidence.

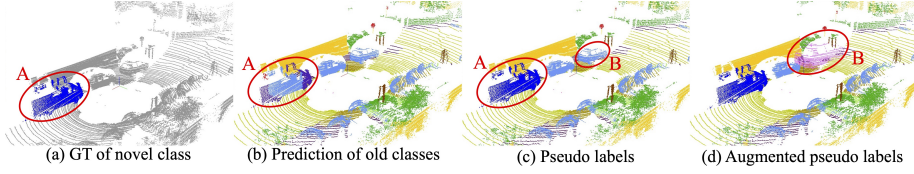


Fig. 4. Pseudo labels generating process for incremental learning. Ground truth (a) only contains the label of the novel class (A: other-vehicle). So we combine the prediction results of \mathcal{M}_o (b) to generate the pseudo labels (c). Then we resize objects of old classes as the synthesized objects in (d) (B: resized car).

two parts, \mathbf{P}_{nm}^{old} , which belongs to old classes \mathcal{K}_0 , and \mathbf{P}_{nm}^{nv} , which belongs to newly introduced classes \mathcal{K}_n , so that $\mathbf{P}_{nm} = \mathbf{P}_{nm}^{old} \cup \mathbf{P}_{nm}^{nv}$. The labels of points \mathbf{P}_{nm}^{nv} are given as \mathbf{Y}_{nm}^{nv} , *e.g.*, labels of A in Fig. 4 (a), but labels of \mathbf{P}_{nm}^{old} are not given, *e.g.*, gray points in Fig. 4 (a). If we only use \mathbf{Y}_{nm}^{nv} to directly finetune the model, it will classify all points as the newly introduced class as there is only one kind of class in the training process. This is called the catastrophic forgetting and we use *Pseudo Label Generation* to solve this problem.

Pseudo Label Generation: We use model \mathcal{M}_o to predict the pseudo labels \mathbf{pY}_{nm}^{old} for \mathbf{P}_{nm}^{old} [7,9], as shown in Fig. 4 (b). In this way, the learned knowledge of old classes is preserved in \mathbf{pY}_{nm}^{old} to alleviate the catastrophic forgetting problem. Then we combine \mathbf{pY}_{nm}^{old} with \mathbf{Y}_{nm}^{nv} to generate the pseudo labels of the whole point cloud \mathbf{Y}_{nm} , such as in Fig. 4 (c).

Loss Function: Note that we keep the open-set property after IL, so the methods in OSeg task including *Unknown Object Synthesis* and *Predictive Distribution Calibration* are still used in IL task. The overall loss function to train the model \mathcal{M}_o from \mathcal{M}_i is

$$\mathcal{L}^{il} = \mathcal{L}_{cal}^{il} + \lambda_{syn} \mathcal{L}_{syn}^{il}, \quad (11)$$

where \mathcal{L}_{cal}^{il} and \mathcal{L}_{syn}^{il} are determined by Eq. 5, Eq. 6, Eq. 7, and Eq. 4. All \mathcal{M} in the related terms are \mathcal{M}_i . Note that \mathbf{Y}_{nm} in Eq. 6 and Eq. 7 are generated as

$$\mathbf{Y}_{nm} = \mathbf{pY}_{nm}^{old} \cup \mathbf{Y}_{nm}^{nv}, \quad (12)$$

where \mathbf{Y}_{nm}^{nv} is the ground truth label of newly introduced classes \mathcal{K}_n and \mathbf{pY}_{nm}^{old} is the pseudo labels of old classes \mathcal{K}_0 generated by \mathcal{M}_o ,

$$\mathbf{pY}_{nm}^{old} = \mathcal{M}_o(\mathbf{P}_{nm}^{old}). \quad (13)$$

The \mathbf{Y}_{nm} in Eq. 12 contains both newly introduced classes \mathcal{K}_n and old classes \mathcal{K}_0 , so \mathcal{M}_i can learn new classes without forgetting old classes.

Inference: To evaluate the performance of IL, we only calculate the closed-set prediction results. This is because, for incremental learning we care about how well the catastrophic forgetting problem is alleviated and the new classes are learned, while the ability to classify the unknown class is already evaluated by Eq. 10 in OSeg task, although after IL the model \mathcal{M}_i can still classify the

Table 1. Benchmark of open-set semantic segmentation for LIDAR point clouds. Results are evaluated on the validation set.

Dataset	SemanticKITTI			nuScenes		
Methods	AUPR	AUROC	mIoU _{old}	AUPR	AUROC	mIoU _{old}
Closed-set	0	0	58.0	0	0	58.7
Upper bound	73.6	97.1	63.5	86.1	99.3	73.8
MSP	6.7	74.0	58.0	4.3	76.7	58.7
MaxLogit	7.6	70.5	58.0	8.3	79.4	58.7
MC-Dropout	7.4	74.7	58.0	14.9	82.6	58.7
REAL	20.8	84.9	57.8	21.2	84.5	56.8

unknown class \mathcal{K}_{rn} . The closed-set inference result $\hat{\mathbf{Y}}'_{close}$ is defined as

$$\hat{\mathbf{Y}}'_{close} = \arg \max_{i=1,2,\dots,C+n} [g_{nm}(f(\mathbf{P}), g_{re}^{nv}(f(\mathbf{P}))]. \quad (14)$$

5 Experiments

We conduct experiments for both tasks of the open-world semantic segmentation, including OSeg and IL tasks. We evaluate our proposed method on two large-scale datasets, SemanticKITTI and nuScenes.

5.1 Open-world Evaluation Protocol

Data Split: We set the novel classes of SemanticKITTI \mathcal{K}_n^{sk} and nuScenes \mathcal{K}_n^{ns} as:

$$\mathcal{K}_n^{sk} = \{other-vehicle\}$$

$$\mathcal{K}_n^{ns} = \{barrier, construction-vehicle, traffic-cone, trailer\}$$

All remaining classes are included in the old class set \mathcal{K}_0^{sk} and \mathcal{K}_0^{ns} . During training of the closed-set model \mathcal{M}_c and open-set model \mathcal{M}_o , we set the labels of novel classes \mathcal{K}_n^{sk} and \mathcal{K}_n^{ns} to be void and ignore them. During incremental learning, we gradually introduce the labels of novel classes \mathcal{K}_n^{sk} and \mathcal{K}_n^{ns} one by one, and set the labels of old classes \mathcal{K}_0^{sk} and \mathcal{K}_0^{ns} to be void.

Evaluation Metrics: To evaluate the performance of the open-set semantic segmentation model \mathcal{M}_o , we consider both the closed-set and open-set segmentation ability. The closed-set ability is measured by mIoU_{close}, while the open-set evaluation is regarded as a binary classification problem between the known class and unknown class, which is measured by area under the ROC curve (AUROC) and area under the precision-recall curve (AUPR) [14].

To evaluate the performance of the model \mathcal{M}_i after incremental learning, we calculate the performance of the old classes mIoU_{old} and newly introduced classes mIoU_{novel} respectively, and also the mIoU of all classes.

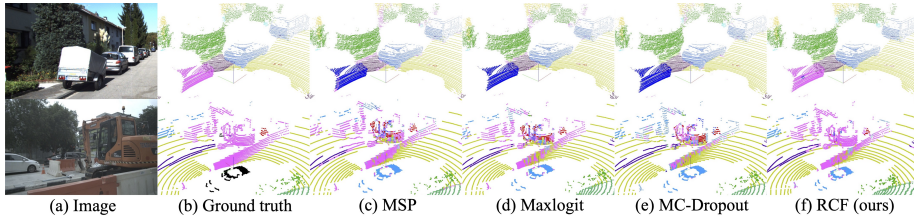


Fig. 5. Qualitative results of OSeg task. Novel classes are in pink (other-vehicle in SemanticKITTI (top), and construction-vehicle and barrier in nuScenes (bottom)). The results show that our method has a better performance in distinguishing the novel class from old classes than all the baselines. Best viewed in zoom.

5.2 Open-set Semantic Segmentation (OSeg)

Implementation: We adopt Cylinder3D as the base network and train the traditional closed-set model \mathcal{M}_c following the training settings in [44] using the labels of old classes \mathcal{K}_0^{sk} and \mathcal{K}_0^{ns} . Then we add several redundancy classifiers on top of the \mathcal{M}_0 and finetune the model \mathcal{M}_c to \mathcal{M}_o based on the training strategies described in Sec. 4.2. The old classes used to synthesize novel objects \mathcal{K}_{syn} are *car* for SemanticKITTI and *car*, *bus*, and *truck* for nuScenes. The probability of resizing these objects p_{syn} is set to 0.5. The unknown object synthesis time is 0.5-4 *ms* based on our experiments, which is sufficiently quick.

Baselines and Upper Bound: We refer to several methods from the open-set 2D semantic segmentation domain and implement them in our 3D LIDAR points domain as our baselines, including MSP, Maxlogit, and MC-Dropout, as discussed in Sec. 2. The upper bound is to use labels of all classes $\mathcal{K}_0 \cup \mathcal{K}_n$ to train the network and regard the softmax probability of the classes \mathcal{K}_n as the confidence score.

Quantitative results: The quantitative results of OSeg are shown in Tab. 1. The closed-set method does not consider the unknown class at all, so the open-set evaluation metrics are 0. Among all open-set semantic segmentation baselines, our REAL achieves remarkably better results on the open-set evaluation metrics. The closed-set mIoU_{old} shows that our method does not sacrifice the ability to classify old classes. The upper bound naturally achieves the best performance as it is conducted in a supervised manner, while the information of the unknown class is not provided for other open-set methods.

Qualitative results: Fig. 5 contains the qualitative results from SemanticKITTI and nuScenes respectively. Fig. 5 top row shows that our method can identify the other-vehicle as the novel class, while all baselines consider it as the truck. In Fig. 5 bottom row, the baselines classify the construction-vehicle as the truck, pedestrian, and manmade, while our method distinguishes it as the novel object.

Ablation experiments: We carefully conduct ablation experiments on the SemanticKITTI dataset to verify the effectiveness of our proposed components. According to the results of Row ID 2 in Tab. 2, using the calibration loss alone can already outperform all baselines in Tab. 1. Furthermore, the result of Row

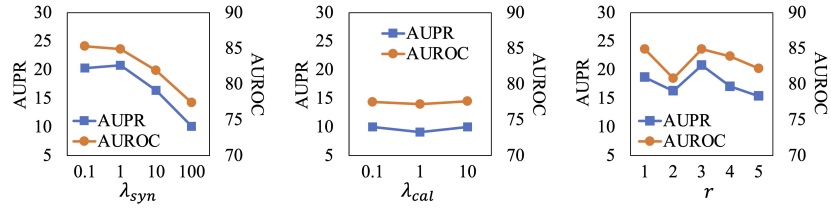


Fig. 6. Ablation experiments of coefficient λ_{syn} , λ_{cal} and number of redundancy classifiers r for OSeg task on SemanticKITTI.

Table 2. Ablation study results of \mathcal{L}_{cal} and \mathcal{L}_{syn} for OSeg task on SemanticKITTI.

Row ID	\mathcal{L}_{cal}	\mathcal{L}_{syn}	AUPR	AUROC	mIoU _{old}
1	×	×	0	0	58.0
2	✓	×	10.0	77.5	58.1
3	✓	✓	20.8	84.9	57.8

ID 3 illustrates that resizing the objects of existing classes with a proper factor is a simple but useful way to imitate novel objects. λ_{syn} and r are set to be 1 and 3 according to Fig. 6. λ_{cal} is 0.1, and it does not influence the result with a large margin based on Fig. 6.

5.3 Incremental Learning

Implementation: We adopt the training strategies described in Sec. 4.3 to finetune the model \mathcal{M}_o to \mathcal{M}_i . The old classes used for synthesis \mathcal{K}_{syn} are the same as the set during training from \mathcal{M}_c to \mathcal{M}_o .

Baselines and upper bound: We adopt direct finetuning of \mathcal{M}_o to \mathcal{M}_i using only the labels of novel classes \mathcal{K}_n^{sk} and \mathcal{K}_n^{ns} to illustrate the catastrophic forgetting problem. Two methods including Feature Extraction and Learning without Forgetting (LwF) [21] using \mathcal{K}_n^{sk} and \mathcal{K}_n^{ns} are regarded as the baselines. The upper bound is the same as the upper bound in the open-set semantic segmentation task, which uses all labels $\mathcal{K}_0 \cup \mathcal{K}_n$ to train the network.

Quantitative results: Tab. 3 and Tab. 4 show the IL performance of SemanticKITTI and nuScenes dataset respectively. Directly finetuning the model \mathcal{M}_o to \mathcal{M}_i only using labels of the novel class incurs the catastrophic forgetting problem, *i.e.*, the network classifies all points as the new class. mIoU_{old} becomes 0 as there is no prediction results in old classes. mIoU_{novel} is also close to 0 as newly introduced class only counts a little portion in the whole point cloud. In contrast, mIoU_{old} in our method is similar with the closed-set, meaning our method can learn the new classes one by one without forgetting the old classes. Our methods has better performance compared to two baselines, showing that using the unlabeled background points \mathbf{Y}_{nm}^{old} is extremely helpful to preserve

Table 3. Incremental learning results on SemanticKITTI 18 + 1 (other-vehicle) setting.

SemanticKITTI 18+1	Validation set			Test set		
Method	mIoU	mIoU _{novel}	mIoU _{old}	mIoU	mIoU _{novel}	mIoU _{old}
Closed-set	58.0	0	61.2	61.8	0	65.3
Upper bound	63.5	44.1	64.6	62.2	40.1	63.5
Finetune	0	0.5	0	0	0	0
Feature extraction	6.8	0.6	7.1	6.9	0.4	7.3
LwF	21.6	1.7	22.7	20.2	0.9	21.3
REAL	64.3	51.5	65.0	61.1	25.3	63.1

Table 4. Incremental learning results on nuScenes for 12 + 4 (barrier, construction-vehicle, traffic-cone, and trailer) setting.

nuScenes 12+4	Validation set			Test set		
Method	mIoU	mIoU _{novel}	mIoU _{old}	mIoU	mIoU _{novel}	mIoU _{old}
Closed-set	58.7	0	78.3	55.8	0	74.4
Upper bound	73.8	62.5	77.6	73.8	70.4	74.8
Finetune	0	0	0	0	0	0
Feature extraction	5.5	2.1	6.6	5.3	1.9	6.4
LwF	6.1	2.4	7.3	5.6	2.5	6.6
REAL	74.9	62.2	79.1	74.2	71.9	75.0

the old knowledge. Compared to the upper bound, our method only needs the ground truth of newly introduced classes \mathcal{K}_n and consumes much less time in training (5 epochs v.s. 35 epochs), while keeping the similar performance.

We show the performance of the model on the nuScenes dataset during IL in Fig. 7. Fig. 7 (a) shows during IL the model are gradually learning novel classes while keeping the performance of old classes. Fig. 7 (b) illustrates the model starts from the closed-set model and finally achieves the comparable performance with the upper bound.

5.4 Open-world Semantic Segmentation

We illustrate the whole open-world semantic segmentation system in Fig. 8. Traditional closed-set model \mathcal{M}_c classifies objects of novel classes \mathcal{K}_n as old classes \mathcal{K}_0 . In Fig. 8 (c), A (construction vehicle) is classified as manmade, pedestrian, and truck; B (barrier) is classified as road and manmade; C (traffic-cone) is classified as road. Such misclassification may cause serious problems in autonomous driving. Thus we conduct the methods in Eq. 8 to finetune \mathcal{M}_c to \mathcal{M}_o so that this open-set model can identify these novel objects as *unknown*, as shown in pink area of Fig. 8 (d). Then, after incremental learning using the methods described in Eq. 11, the model can gradually classify new classes, *e.g.*, A (barrier), B (construction-vehicle), and C (traffic-cone) in Fig. 8 (e), (f), and (g). Note that after incremental learning the model can still identify unknown classes, as shown in the pink areas of Fig. 8 (e).

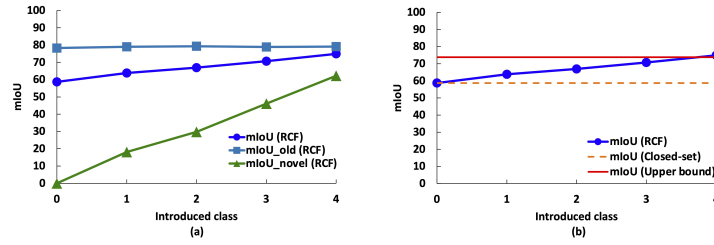


Fig. 7. Incremental learning results for nuScenes validation set. Introduced class: 1: barrier; 2: construction-vehicle; 3: traffic-cone; 4: trailer.

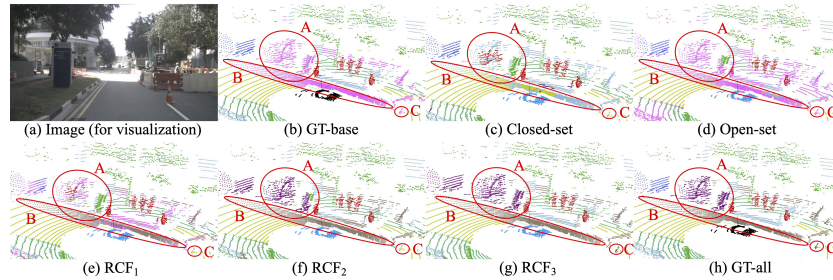


Fig. 8. Qualitative results of open-world semantic segmentation. GT: ground truth. In (b) GT-base we set the novel classes \mathcal{K}_n in pink (A: construction-vehicle; B: barrier; C: traffic-cone). (c) Closed-set prediction classifies novel objects as old classes. (d) Open-set prediction can identify these novel objects as *unknown*. We gradually introduce the labels of barrier, construction-vehicle, and traffic-cone in (e) REAL₁, (f) REAL₂, and (g) REAL₃, so they can classify these novel classes one by one. (h) GT-all contains ground truth of all classes.

6 Conclusion

Traditional closed-set semantic segmentation cannot handle objects of novel classes. In this paper, we propose the open-world semantic segmentation for LIDAR point clouds, where the model can identify novel objects (open-set semantic segmentation) and then gradually learn them when labels are available (incremental learning). We propose the redundancy classifier framework (REAL) and corresponding training and inference strategies to fulfill the open-world semantic segmentation system. We hope this work can draw the attention of researchers toward this meaningful and open problem.

References

1. Baur, C., Wiestler, B., Albarqouni, S., Navab, N.: Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. In: International MICCAI Brainlesion Workshop (2018)
2. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Gall, J., Stachniss, C.: Towards 3D LiDAR-based semantic scene understanding of 3D point cloud sequences: The SemanticKITTI Dataset. *The International Journal on Robotics Research* (2021)
3. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In: ICCV (2019)
4. Bendale, A., Boulton, T.: Towards open world recognition. In: CVPR (2015)
5. Bozhinowski, D., Di Ruscio, D., Malavolta, I., Pelliccione, P., Crnkovic, I.: Safety for mobile robotic systems: A systematic mapping study from a software engineering perspective. *Journal of Systems and Software* (2019)
6. 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)
7. Cen, J., Yun, P., Cai, J., Wang, M.Y., Liu, M.: Deep metric learning for open world semantic segmentation. In: ICCV (2021)
8. Cermelli, F., Mancini, M., Bulò, S.R., Ricci, E., Caputo, B.: Modeling the background for incremental learning in semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* pp. 9230–9239 (2020). <https://doi.org/10.1109/CVPR42600.2020.00925>
9. Cermelli, F., Mancini, M., Bulò, S.R., Ricci, E., Caputo, B.: Modeling the background for incremental learning in semantic segmentation. In: CVPR (2020)
10. Cheng, R., Razani, R., Taghavi, E., Li, E., Liu, B.: 2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In: CVPR (2021)
11. Delange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., Tuytelaars, T.: A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021)
12. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: ICML (2016)
13. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In: CVPR (2012)
14. Hendrycks, D., Basart, S., Mazeika, M., Mostajabi, M., Steinhardt, J., Song, D.: Scaling out-of-distribution detection for real-world settings. *arXiv preprint arXiv:1911.11132* (2019)
15. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. In: ICLR (2017)
16. Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A.: Learning semantic segmentation of large-scale point clouds with random sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021)
17. Joseph, K.J., Khan, S., Khan, F.S., Balasubramanian, V.N.: Towards open world object detection. In: CVPR (2021)
18. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: NeurIPS (2017)

19. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: *NeurIPS* (2017)
20. Li, D., Tasci, S., Ghosh, S., Zhu, J., Zhang, J.T., Heck, L.: Rilod: near real-time incremental learning for object detection at the edge. In: *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*. pp. 113–126 (2019)
21. Li, Z., Hoiem, D.: Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018)
22. Lis, K., Nakka, K., Fua, P., Salzmann, M.: Detecting the unexpected via image resynthesis. In: *ICCV* (2019)
23. Liu, L., Kuang, Z., Chen, Y., Xue, J., Yang, W., Zhang, W.: Incdet: In defense of elastic weight consolidation for incremental object detection. *IEEE Transactions on Neural Networks and Learning Systems* pp. 1–14 (2020)
24. MacKay, D.J.: Bayesian neural networks and density networks. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* (1995)
25. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: The sequential learning problem. In: *Psychology of learning and motivation* (1989)
26. Milioto, A., Vizzo, I., Behley, J., Stachniss, C.: Rangenet++: Fast and accurate lidar semantic segmentation. In: *IROS* (2019)
27. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: *CVPR* (2019)
28. Peng, C., Zhao, K., Lovell, B.: Faster ilod: Incremental learning for object detectors based on faster rcnn. *arXiv preprint arXiv:2003.03901* (2020)
29. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: *CVPR* (2017)
30. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413* (2017)
31. Rannen, A., Aljundi, R., Blaschko, M.B., Tuytelaars, T.: Encoder based lifelong learning. In: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 1320–1328 (2017)
32. Shmelkov, K., Schmid, C., Alahari, K.: Incremental learning of object detectors without catastrophic forgetting. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. pp. 3420–3429 (2017)
33. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: Flexible and deformable convolution for point clouds. In: *ICCV* (2019)
34. Wang, Y., Li, B., Che, T., Zhou, K., Liu, Z., Li, D.: Energy-based open-world uncertainty modeling for confidence calibration. In: *ICCV* (2021)
35. Wang, Y., Shi, T., Yun, P., Tai, L., Liu, M.: Pointseg: Real-time semantic segmentation based on 3d lidar point cloud. *arXiv preprint arXiv:1807.06288* (2018)
36. Welling, M., Teh, Y.W.: Bayesian learning via stochastic gradient langevin dynamics. In: *ICML*. pp. 681–688
37. Wu, B., Zhou, X., Zhao, S., Yue, X., Keutzer, K.: Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In: *ICRA* (2019)
38. Wu, W., Qi, Z., Fuxin, L.: Pointconv: Deep convolutional networks on 3d point clouds. In: *CVPR* (2019)
39. Xia, Y., Zhang, Y., Liu, F., Shen, W., Yuille, A.L.: Synthesize then compare: Detecting failures and anomalies for semantic segmentation. In: *ECCV* (2020)
40. Yun, P., Cen, J., Liu, M.: Conflicts between likelihood and knowledge distillation in task incremental learning for 3d object detection. In: *3DV* (2021)

41. Yun, P., Liu, Y., Liu, M.: In defense of knowledge distillation for task incremental learning and its application in 3d object detection. *IEEE Robotics and Automation Letters* **6**(2), 2012–2019 (2021). <https://doi.org/10.1109/LRA.2021.3060417>
42. Zhang, Y., Zhou, Z., David, P., Yue, X., Xi, Z., Gong, B., Foroosh, H.: Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In: *CVPR* (2020)
43. Zhou, D.W., Ye, H.J., Zhan, D.C.: Learning placeholders for open-set recognition. In: *CVPR* (2021)
44. Zhu, X., Zhou, H., Wang, T., Hong, F., Ma, Y., Li, W., Li, H., Lin, D.: Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In: *CVPR* (2021)