

Open-world Semantic Segmentation for LIDAR Point Clouds

Jun Cen Peng Yun Junhao Cai Di Luan Michael Yu Wang Ming Liu
The Hong Kong University of Science and Technology
`{jcenaa, pyun, jcaciaq, dluan}@connect.ust.hk, {mywang, eelium}@ust.hk`

Abstract

*Classical LIDAR semantic segmentation is not robust enough for real-world applications, such as 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, when labels of novel classes are given. We propose the Redundancy Classifier Framework (RCF) to provide a general architecture for both the open-set semantic segmentation and incremental learning problems. Our experiments show that our approach achieves state-of-the-art performance in the open-set semantic segmentation task on the SemanticKITTI and nuScenes datasets, and alleviates the catastrophic forgetting problems with a large margin during 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 [1–4], benefiting from well-annotated datasets including SemanticKITTI [5–7] and nuScenes [8]. 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 assign the labels of old classes to novel classes by mistake, which may have disastrous consequences in safety-sensitive applications, such as autonomous driving [9]. 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.

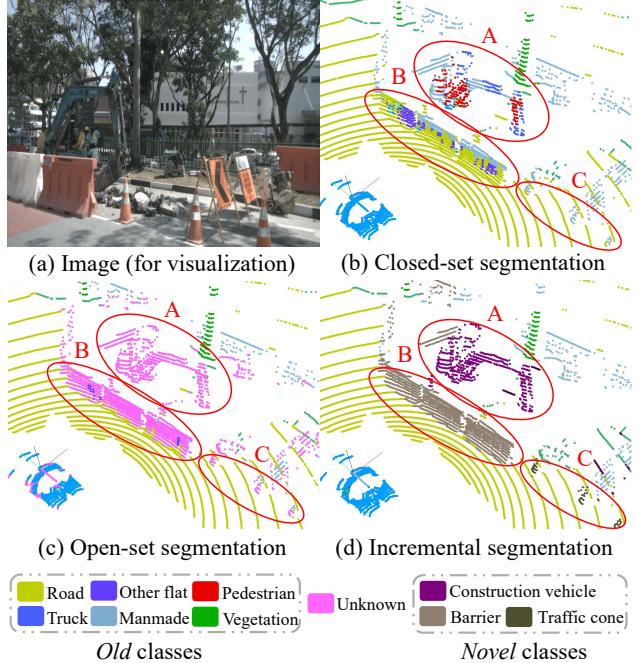


Figure 1. Closed-set segmentation wrongly assigns the labels of old classes to novel objects (**A**: construction vehicle is classified as the manmade, truck, and even pedestrian; **B**: barrier is classified as the road, manmade and other flat; **C**: traffic cone is classified as the manmade). Open-set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

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 (OSSS)** 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 the open-set semantic segmentation (OSSS) 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 [10–12] and uncertainty-based methods [13–15], though none of them can be directly utilized. Generative network-based methods adopt a conditional generative adversarial network (cGAN) [16] 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 always predicts the results with high confidence scores, no matter for old classes or novel classes, as shown in Fig. 3 (a) and (c).

In addition to the challenges of the OSSS task, the catastrophic forgetting of old classes in incremental learning (IL) [17] is another problem to solve. Directly finetuning 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* property 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 the **Redundancy Classifier Framework (RCF)** to provide a dynamic classifier architecture to adapt the model to both the OSSS and IL tasks. For the OSSS 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 new 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 OSSS and IL tasks under RCF, based on the unknown object synthesis, predictive distribution calibration, and pseudo label generation. We show the effectiveness of the RCF 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 open-set semantic segmentation (OSSS) and incremental learning (IL) tasks.
- We propose the Redundancy Classifier Framework (RCF) to provide a general architecture for both the OSSS 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

OSSS 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 RCF.

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 [18–20] use PointNet [21] and PointNet++ [22] 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 [23] and RangeNet++ [4] convert the point cloud to a range image while PolarNet [3] converts the point cloud to the bird’s-eye-view under the polar coordinates. However, 2D representations inevitably lose some of the 3D topology and geometric information. Cylinder3D [1] 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 [5–7] and nuScenes [8], so we adopt it as the base architecture in our work.

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) [13] is the baseline of uncertainty-based methods, while Dan *et al.* [24] found that Maximum Logit (MaxLogit) is a better choice than the probability. MC-Dropout [14] and Ensembles [15] are used to approximate Bayesian inference [25, 26], which regards the network from a probabilistic view. Meanwhile, generative-based methods, including SynthCP [12] and DUIR [11], adopt conditional GAN (cGAN) [16] 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. [27, 28] 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) [29] to approximate the unknown class distribution. We draw inspiration from them, and take a step further by using RCs for both OSSS 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.* [30], 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 classifi-

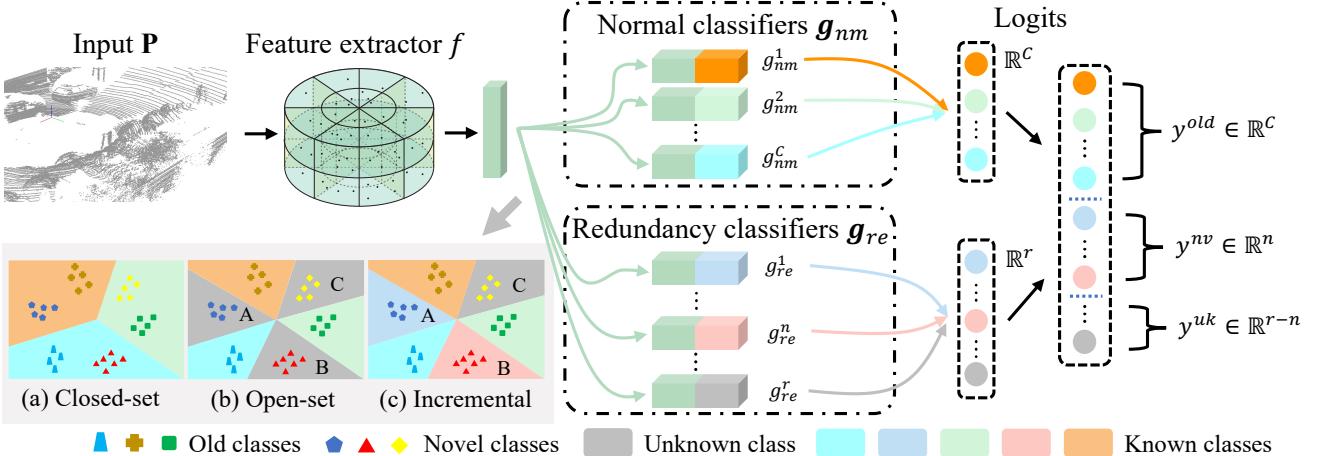


Figure 2. Redundancy Classifier Framework (RCF). On top of the original output for old classes (y^{old}), we add redundancy classifiers (RCs) to output the logits for the unknown class (y^{uk}) and introduced novel classes (y^{nv}). Novel classes are classified as old classes in the closed-set prediction in (a). In OSSS, RCs (gray regions in (b)) are all used to assign the *unknown* label for novel classes (A, B, and C in (b)). After IL, several RCs are trained to classify introduced novel classes, so that they become *known* for the network (A and B in (c)), while the remaining RCs are still kept for the unknown class (C in (c)).

cation 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.* [31] extended the open-world problem to the 2D object detection domain, and proposed a methodology which is based on contrastive clustering, an unknown-aware proposal network and energy-based unknown identification to address the challenges of open-world detection. Jun *et al.* [32] later adopted deep metric learning for open-world semantic segmentation for 2D images. Here, we extend the open-world problem to the 3D LIDAR cloud point domain, and both sub-tasks including OSSS 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}_0 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 (OSSS) and incremental learning (IL). For the OSSS task, the model \mathcal{M}_0 will be finetuned to \mathcal{M}_0^{open} 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}_0^{open} will be further finetuned to \mathcal{M}_n^{open} 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. Note that during IL, 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}_n^{open} still keeps the open-set property, *i.e.*, assigns the *unknown* label to the remaining novel classes $\{C+n+1, \dots\}$.

4. RCF: Redundancy Classifier Framework

The closed-set network can only output the predictions of old classes, even for the points of the novel class \mathcal{U} , as the whole feature space is carved up by the normal classifiers, as illustrated in Fig. 2 (a). Therefore, we propose the Redundancy Classifier Framework (RCF) to provide a general architecture for both the OSSS and IL tasks. In the OSSS task, redundancy classifiers (RCs) are used to classify the novel classes as *unknown*, so part of feature space is occupied by the RCs for the unknown class, as shown in gray regions A, B, and C of Fig. 2 (b). Then several RCs are trained to classify novel classes introduced during IL when new labels are available, so that introduced novel classes be-

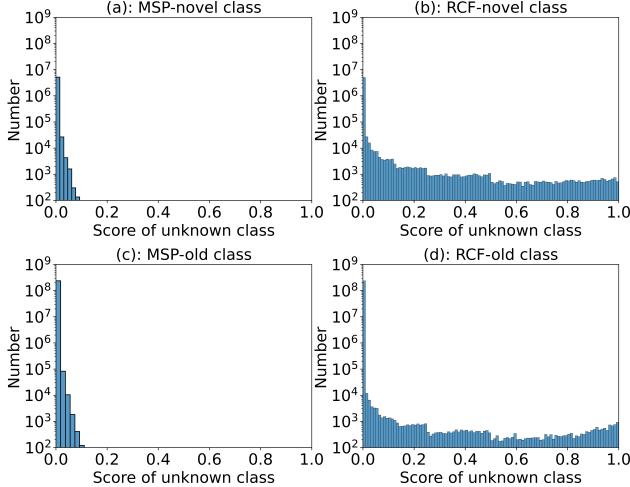


Figure 3. Distribution of scores of the unknown class for Maximum Softmax Probability (MSP) and our RCF method. The scores of the unknown class for novel classes are low in MSP (a), meaning the closed-set prediction classifies novel classes as old classes with a high softmax probability, while in our method much more novel objects are classified as the unknown class with a high score (b). More details are discussed in the supplementary material.

come *known* for the network, *e.g.*, region A and B in Fig. 2 (c), while the remaining RCs are still kept for the unknown class, *e.g.*, region C in Fig. 2 (c).

We adopt the unknown object synthesis and predictive prediction calibration described in Sec. 4.2 and Sec. 4.3 to train the RCs in the OSSS task. On top of the OSSS task, we develop the pseudo label generation strategy described in Sec. 4.4 for the IL task. Note that during the IL task, we still keep the training strategies in the OSSS task so that the model after IL maintains the open-set property.

4.1. Network Architecture under RCF

The original trained model \mathcal{M}_0 , which can already 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} , the output of the model \mathcal{M}_0 is

$$\mathcal{M}_0(\mathbf{P}) = [g_{nm}(f(\mathbf{P}))] \in \mathbb{R}^{M \times C}. \quad (1)$$

OSSS task: We add r redundancy classifiers (RCs) $g_{re} = \{g_{re}^1, g_{re}^2, \dots, g_{re}^r\}$ (*e.g.*, A, B, and C in Fig. 2 (b)) on top of the original feature extractor f , and we let the maximum response of all RCs 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}_0^{open} is

$$\mathcal{M}_0^{open}(\mathbf{P}) = [\max g_{re}(f(\mathbf{P})), g_{nm}(f(\mathbf{P}))] \in \mathbb{R}^{M \times (1+C)}. \quad (2)$$

IL task: 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 introduced

novel classes \mathcal{K}_n (*e.g.*, A and B in Fig. 2 (c)), 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 (*e.g.*, C in Fig. 2 (c)). In this way, the output of \mathcal{M}_n^{open} can be represented as

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

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

4.2. 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 novel 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

$$\mathcal{L}_{syn} = \ell(\mathcal{M}_0^{open}(\mathbf{P}_{syn}), \mathbf{0}), \quad (4)$$

where ℓ is the cross-entropy loss, and \mathcal{M}_0^{open} is \mathcal{M}_0^{open} for the OSSS task and \mathcal{M}_n^{open} for the IL task. The ground truth labels of synthesized objects are set to be the unknown class 0, so the first terms in Eq. (2) and Eq. (3) are trained to give high scores to objects not seen before.

4.3. 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 calibrating the probability: force every point of old classes to have the largest probability on its original class, and have the second largest probability on the unknown class. 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

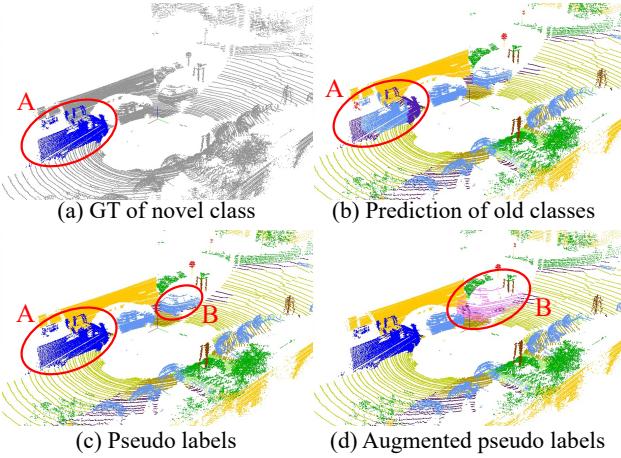


Figure 4. Pseudo labels generating process for incremental learning. Ground truth (a) only contains the label of the novel class (A: other-vehicle). So we combine the prediction results of \mathcal{M}_0^{open} (b) to generate the pseudo labels (c). Then we resize objects of old classes as the synthesized objects in (d) (B: resized car).

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

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

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

where $\mathcal{M}^{open}(\mathbf{P}_{nm}) \setminus \mathbf{Y}_{nm}$ 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 of OSSS task: In this task, the ground truth labels \mathbf{Y}_{nm} of unchanged normal objects \mathbf{P}_{nm} are available. So the overall loss function to train the model \mathcal{M}_0^{open} is

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

where \mathcal{L}_{cal}^{oss} is determined by Eq. (5), Eq. (6), and Eq. (7), while \mathcal{L}_{syn}^{oss} is determined by Eq. (4). All \mathcal{M}^{open} in the related loss terms are \mathcal{M}_0^{open} .

4.4. Pseudo Label Generation

After obtaining the open-set model \mathcal{M}_0^{open} by Eq. (8), we conduct IL to finetune \mathcal{M}_0^{open} to \mathcal{M}_n^{open} . 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 two parts, \mathbf{P}_{nm}^{old} , which belongs to old classes \mathcal{K}_0 , and \mathbf{P}_{nm}^{nv} , which belongs to novel 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). So the key

is how to deal with \mathbf{P}_{nm}^{old} as their labels are not given, e.g., gray points in Fig. 4 (a).

We propose to use model \mathcal{M}_0^{open} to predict the pseudo labels \mathbf{pY}_{nm}^{old} for \mathbf{P}_{nm}^{old} , as shown in Fig. 4 (b). This is reasonable as this method can preserve the learned knowledge of old classes 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 of IL task: The overall loss function to train the model \mathcal{M}_n^{open} from \mathcal{M}_0^{open} is

$$\mathcal{L}^{il} = \mathcal{L}_{cal}^{il} + \lambda_{syn} \mathcal{L}_{syn}^{il}, \quad (9)$$

where \mathcal{L}_{cal}^{il} and \mathcal{L}_{syn}^{il} are determined by Eq. (5), Eq. (6), Eq. (7), and Eq. (4). Note that \mathbf{Y}_{nm} in Eq. (6) and Eq. (7) are generated as

$$\mathbf{Y}_{nm} = \mathbf{pY}_{nm}^{old} \cup \mathbf{Y}_{nm}^{nv}, \quad (10)$$

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

$$\mathbf{pY}_{nm}^{old} = \mathcal{M}_0^{open}(\mathbf{P}_{nm}^{old}). \quad (11)$$

4.5. Inference

OSSS task: Both the closed-set and open-set performance of the finetuned model \mathcal{M}_0^{open} will be evaluated. For the closed-set prediction, the inference result $\hat{\mathbf{Y}}_{close}$ is defined as

$$\hat{\mathbf{Y}}_{close} = \arg \max_{i=1,2,\dots,C} g_{nm}(f(\mathbf{P})). \quad (12)$$

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:

$$\hat{\mathbf{Y}}_{open} = \begin{cases} \arg \max_{i=1,2,\dots,C} g_{nm}(f(\mathbf{P})) & conf < th \\ 0 & otherwise, \end{cases} \quad (13)$$

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

IL task: To evaluate the performance of incremental learning, 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. (13), although after incremental learning the model \mathcal{M}_n^{open} can still classify the unknown class. The closed-set inference result $\hat{\mathbf{Y}}'_{close}$ is defined as

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

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
RCF	20.8	84.9	57.8	21.2	84.5	56.8

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

5. Experiments

We conduct experiments for both tasks of the open-world semantic segmentation, including OSSS 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:

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

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

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}_0 and open-set model \mathcal{M}_0^{open} , 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}_0^{open} , 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) [24].

To evaluate the performance of the model \mathcal{M}_n^{open} 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.

5.2. Open-set Semantic Segmentation

Implementation: We adopt Cylinder3D as the base network and train the traditional closed-set model \mathcal{M}_0 following the training settings in [1] 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}_0 to \mathcal{M}_0^{open} based on the training strategies described in Eq. (8). The old classes used to synthesize novel objects \mathcal{K}_{syn} are *car*

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

Table 2. Ablation study results of \mathcal{L}_{cal} and \mathcal{L}_{syn} for open-set semantic segmentation task.

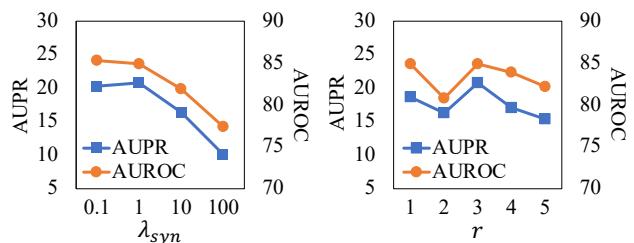


Figure 5. Ablation experiments of coefficient λ_{syn} and number of redundancy classifiers r for open-set semantic segmentation task.

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 open-set semantic segmentation 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 RCF 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.

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. Furthermore, the result of Row 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. 5. λ_{cal} is 0.1, and it does not influence the result with a large margin, based on our experiments results in the supplementary material.

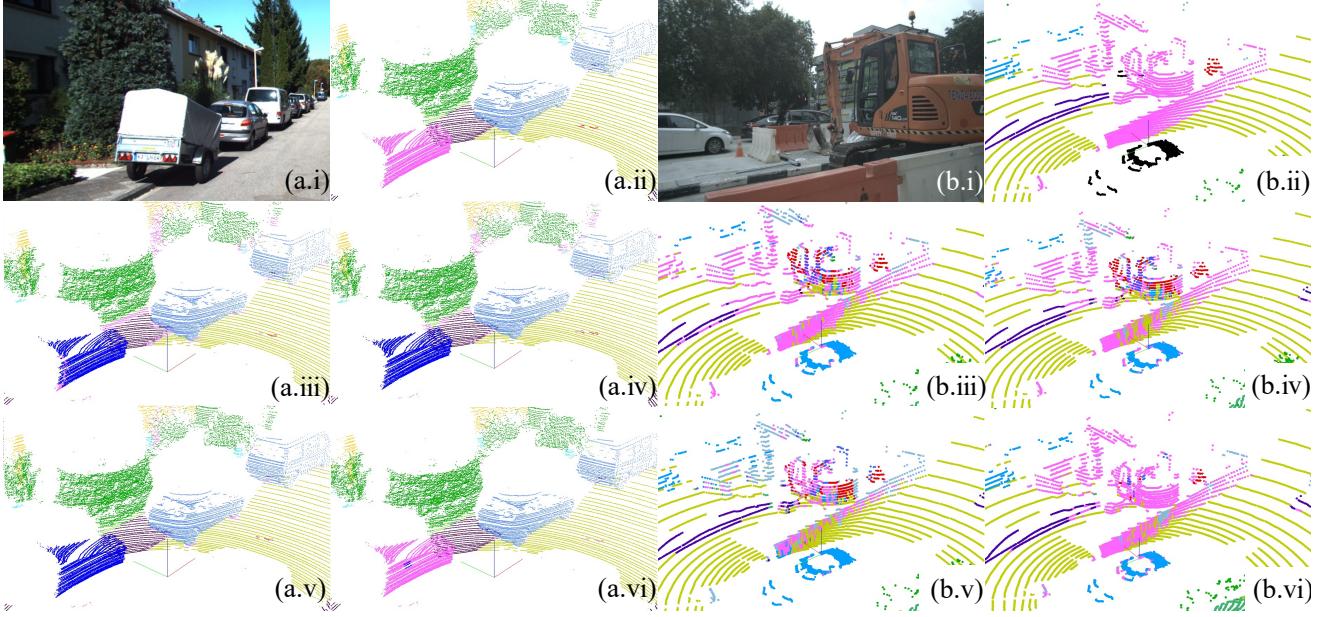


Figure 6. (a) and (b) come from SemanticKITTI and nuScenes respectively. (i) Image; (ii) Ground truth; (iii) MSP method; (iv) MaxLogit method; (v) MC-Dropout method; (vi) RCF method (ours). Novel objects are in pink (other-vehicle in (a), construction-vehicle and barrier in (b)). The results show that our method has a better performance in distinguishing the novel class from old classes than all the baselines.

Method	car	bicycle	motorcycle	truck	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	truck	terrain	pole	traffic	other-vehicle	mIoU	mIoU _{novel}	mIoU _{old}
Closed-set	94.2	56.6	48.8	50.4	63.3	56.9	45.4	90.7	65.4	74.2	26.4	90.7	63.6	84.2	70.8	65.9	61.7	65.2	0.0	61.8	0.0	65.2
Upper bound	95.8	50.1	50.6	49.2	63.3	58.3	11.9	91.4	64.0	74.9	25.4	91.1	65.2	85.7	70	68.5	60.8	66.2	40.1	62.2	40.1	63.4
Finetune	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
RCF	95.8	49.4	45.7	41.3	61.3	58.3	32.7	91.3	65.1	74.8	24.5	89.7	57.9	84.7	69.8	66.8	60.9	65.8	25.3	61.1	25.3	63.1

Table 3. Incremental learning results on SemanticKITTI for 18 + 1 (other-vehicle) setting. The novel classes are in blue. Finetune is the baseline, as described in Sec. 5.3. It classifies all points as the introduced novel class, so it has the catastrophic forgetting problem of old classes. Our method, RCF, can alleviate the catastrophic forgetting problem with a large margin. All results are reported in IoU.

Qualitative results: Fig. 6 contains the qualitative results from SemanticKITTI and nuScenes respectively. Fig. 6 (a) shows that our method can identify the other-vehicle as the novel class, while all baselines consider it as the truck. In Fig. 6 (b), the baselines classify the construction-vehicle as the truck, pedestrian, and manmade, while our method distinguishes it as the novel object.

5.3. Incremental Learning

Implementation: We keep the architecture of \mathcal{M}_0^{open} unchanged, and adopt the training strategies described in Eq. (9) to finetune the model \mathcal{M}_0^{open} to \mathcal{M}_n^{open} . The old classes used for synthesis are the same as the set during training from \mathcal{M}_0 to \mathcal{M}_0^{open} .

Baselines and upper bound: We adopt direct finetuning of \mathcal{M}_0^{open} to \mathcal{M}_n^{open} using only the labels of novel classes

\mathcal{K}_n^{sk} and \mathcal{K}_n^{ns} as the baseline. 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 incremental learning performance of SemanticKITTI and nuScenes dataset respectively. All results are reported under the test set. Tab. 3 and Tab. 4 show that directly finetuning the model \mathcal{M}_0^{open} only using labels of the novel class incurs the catastrophic forgetting problem, *i.e.*, the network classifies all points as the new class, so the IoUs of all classes become 0. In contrast, our method can keep the performance of the old classes (mIoU_{old} is 63.1 for SemanticKITTI and 75.0 for nuScenes), as well as learn the new classes one by one. The overall mIoU of our method is close to the upper bound (61.1 and 62.2 for our method and upper bound for SemanticKITTI; 74.2 and 73.8 for our method and up-

Method	bicycle	bus	car	motorcycle	pedestrian	truck	driveable -surface	other-flat	sidewalk	terrain	manmade	vegetation	barrier	construction -vehicle	traffic-cone	trailer	mIoU	mIoU _{novel}	mIoU _{old}
Closed-set	28.0	83.0	86.3	75.8	74.1	58.6	96.9	67.2	77.1	72.6	86.6	86.0	0.0	0.0	0.0	0.0	55.8	0	74.4
Upper bound	26.7	83.7	84.7	72.6	73.7	68.8	96.9	68.8	75.7	71.0	88.3	86.4	80.9	55.1	67.4	79.4	73.8	70.7	74.8
Finetune ₄	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
RCF ₁	28.1	83.6	86.7	78.1	75.0	58.4	97.1	67.3	77.6	74.0	88.3	87.2	80.9	0.0	0.0	0.0	61.4	20.2	75.1
RCF ₂	27.3	82.5	86.3	77.7	75.4	58.6	97.1	67.0	77.4	73.3	88.1	86.8	80.9	56.7	0.0	0.0	64.7	34.4	74.8
RCF ₃	28.1	82.1	86.1	77.7	75.4	57.6	97.0	66.3	77.4	73.7	88.4	87.0	81.1	57.6	66.5	0.0	68.9	51.3	74.8
RCF ₄	23.3	82.7	85.8	76.2	75.0	68.8	96.9	65.0	77.4	73.2	88.7	86.8	81.2	58.6	66.8	81.1	74.2	71.9	75.0

Table 4. Incremental learning results on nuScenes for 12 + 4 (barrier, construction-vehicle, traffic-cone, and trailer) setting. The novel classes are in blue. RCF₁ to RCF₄ means we introduce the label of one novel class per time and conduct incremental learning one by one.

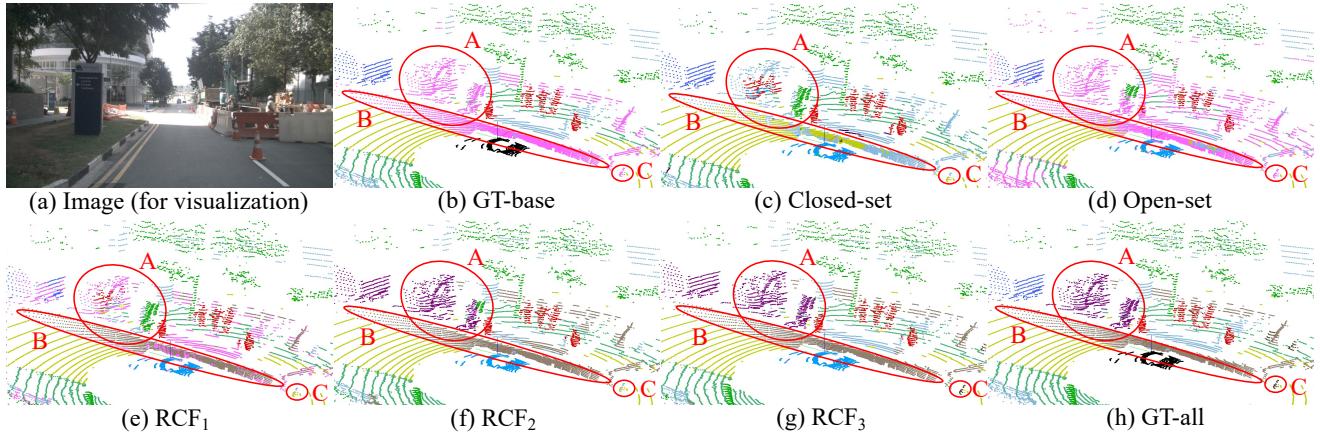


Figure 7. Qualitative results of open-world semantic segmentation. GT: ground truth. In (b) GT-base we set the novel classes \mathcal{K}_n in pink (**A**: construction-vehicle; **B**: barrier; **C**: traffic-cone). (c) Closed-set prediction classifies novel objects as old classes. (d) Open-set prediction can identify these novel objects. We gradually introduce the labels of barrier, construction-vehicle, and traffic-cone in (e) RCF₁, (f) RCF₂, and (g) RCF₃, so they can classify these novel classes one by one. (h) GT-all contains ground truth of all classes.

per bound for nuScenes), meaning that using our method to incrementally learn the new classes one by one has the similar performance to using all labels and training from scratch. Kindly refer to supplementary material for the results on the validation set and more discussion.

5.4. Open-world Semantic Segmentation

We illustrate the whole open-world semantic segmentation system in Fig. 7. Traditional closed-set model \mathcal{M}_0 classifies objects of novel classes \mathcal{K}_n as old classes \mathcal{K}_0 . In Fig. 7 (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}_0 to $\mathcal{M}_0^{\text{open}}$ so that this open-set model can identify these novel objects, as shown in Fig. 7 (d). Then, after incremental learning using the methods described in Eq. (9), the model can gradually classify new classes, e.g.,

$\mathcal{M}_1^{\text{open}}$ can classify B (barrier), $\mathcal{M}_2^{\text{open}}$ can classify A (construction-vehicle), and $\mathcal{M}_3^{\text{open}}$ can classify C (traffic-cone), as shown in Fig. 7 (e), (f), and (g). Note that after incremental learning the model can still identify unknown classes, as shown in the pink areas of 7 (e).

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 (RCF) 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] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9939–9948, 2021. [1](#), [2](#), [6](#)
- [2] Ran Cheng, Ryan Razani, Ehsan Taghavi, Enxu Li, and Bingbing Liu. 2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12547–12556, 2021. [1](#)
- [3] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9601–9610, 2020. [1](#), [2](#)
- [4] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4213–4220. IEEE, 2019. [1](#), [2](#)
- [5] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019. [1](#), [2](#)
- [6] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, J. Gall, and C. Stachniss. Towards 3D LiDAR-based semantic scene understanding of 3D point cloud sequences: The SemanticKITTI Dataset. *The International Journal on Robotics Research*, 40(8-9):959–967, 2021. [1](#), [2](#)
- [7] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012. [1](#), [2](#)
- [8] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liang, Qiang Xu, Anush Krishnan, Yu Pan, Giacarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. [1](#), [2](#)
- [9] 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. [1](#)
- [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] 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](#)
- [14] 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](#)
- [15] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6402–6413, 2017. [2](#)
- [16] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019. [2](#)
- [17] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989. [2](#)
- [18] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Learning semantic segmentation of large-scale point clouds with random sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. [2](#)
- [19] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6411–6420, 2019. [2](#)
- [20] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. [2](#)
- [21] 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 conference on computer vision and pattern recognition*, pages 652–660, 2017. [2](#)
- [22] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. [2](#)
- [23] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and

- unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4376–4382. IEEE, 2019. 2
- [24] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadmreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. *arXiv preprint arXiv:1911.11132*, 2019. 2, 6
- [25] David JC MacKay. Bayesian neural networks and density networks. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 354(1):73–80, 1995. 2
- [26] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5574–5584, 2017. 2
- [27] Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Learning placeholders for open-set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2021. 2
- [28] Yezhen Wang, Bo Li, Tong Che, Kaiyang Zhou, Ziwei Liu, and Dongsheng Li. Energy-based open-world uncertainty modeling for confidence calibration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9302–9311, October 2021. 2
- [29] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011. 2
- [30] A. Bendale and T. Boult. Towards open world recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1893–1902, 2015. 2
- [31] K J Joseph, Salman Khan, Fahad Shahbaz Khan, and Vi neeth N Balasubramanian. Towards open world object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3
- [32] Jun Cen, Peng Yun, Junhao Cai, Michael Yu Wang, and Ming Liu. Deep metric learning for open world semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15333–15342, October 2021. 3