

Understanding the Challenges When 3D Semantic Segmentation Faces Class Imbalanced and OOD Data

Yancheng Pan¹, Member, IEEE, Fan Xie, Member, IEEE, and Huijing Zhao¹, Senior Member, IEEE

Abstract—3D semantic segmentation (3DSS) is an essential process in the creation of a safe autonomous driving system. However, deep learning models for 3D semantic segmentation often suffer from the class imbalance problem and out-of-distribution (OOD) data. In this study, we explore how the class imbalance problem affects 3DSS performance and whether the model can detect the category prediction correctness, or whether data is ID or OOD. For these purposes, we conduct two experiments using four representative 3DSS models and five trust scoring methods, and conduct both a confusion and feature analysis of each class. Furthermore, a data augmentation method for the 3D LiDAR dataset is proposed to create a new dataset based on SemanticKITTI and SemanticPOSS, called AugKITTI. We propose the wPre metric and TSD for a more in-depth analysis of the results, and follow are proposals with an insightful discussion. Based on the experimental results, we find that: 1) classes are not only imbalanced in their data size but also in the basic properties of each semantic category; 2) intraclass diversity and interclass ambiguity make class learning difficult and greatly limit the models' performance, creating the challenges of semantic and data gaps; 3) trust scores are unreliable for classes whose features are confused with other classes. For 3DSS models, those misclassified ID classes and OODs may also be given high trust scores, making the 3DSS predictions unreliable, and leading to the challenges in judging 3DSS result trustworthiness. All of these outcomes point to several research directions for improving the performance and reliability of the 3DSS models used for real-world applications.

Index Terms—3D LiDAR, semantic segmentation, OOD detection, class imbalance.

I. INTRODUCTION

SEMANTIC segmentation [1], [2] is a fundamental perception task that finds semantically interpretable categories of each unit of scene data. The unit can be an image pixel [1] or 3D point [3]. A fine-grained semantic understanding is essential for an autonomous agent to navigate

Manuscript received 28 February 2022; revised 19 October 2022 and 7 January 2023; accepted 6 March 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 61973004 and Grant U22A2061 and in part by the High-Performance Computing Platform of Peking University. The Associate Editor for this article was J. Li. (*Corresponding author: Huijing Zhao.*)

Yancheng Pan and Huijing Zhao are with the Key Laboratory of Machine Perception (MOE), School of Artificial Intelligence (AI), Peking University, Beijing 100084, China (e-mail: panyancheng@pku.edu.cn; zhaohj@pku.edu.cn).

Fan Xie is with the School of Electronics Engineering and Computer Science (EECS), Peking University, Beijing 100084, China.

Digital Object Identifier 10.1109/TITS.2023.3256442

safely and smoothly in complex driving scenes. 3D LiDAR sensors are currently popular devices for mobile robots [4] and autonomous driving systems [5], [6] and can capture realistic images of the surroundings with rich 3D geometric shapes. Semantic segmentation using 3D LiDAR data as input (3D semantic segmentation, 3DSS), has been widely studied, and deep learning techniques have made promising progress in this task in recent years [3], [7].

Deep models require a large amount of training data. The performance limitation caused by insufficient training data is called “the data hungry effect” [8]. As described in [9], 3DSS studies that use deep learning techniques suffer severe data hunger problems, where 3D LiDAR datasets of real-world scenes are very limited and the class imbalance (also called the long-tail) problem is one of the key issues. Class imbalance is a common problem in machine learning and has been extensively studied [10], which means that the model could not be sufficiently learned for classes with a few samples. The class imbalance problem is even more severe for 3D LiDAR datasets due to the data acquisition method and the proportion of scene objects acquired in the real world. Although some studies have addressed alleviating the influence of a class imbalance during model training [11], there has been no rigorous study pertaining to the following question: *How does the class imbalance problem affect 3DSS models' performance?*

Out-of-distribution (OOD) data are another key issue when deploying an AI system in the real world [8], [12]. The problem is very severe for safety-critical applications such as autonomous driving, where some of the categories are unseen or rare in the datasets but need to be handled in the system [13]. OOD detection has also been studied as a general problem in machine learning [14]. However, the task is more difficult when 3DSS is faced with the dual challenges from class imbalances and OOD data, where the model could show a high confidence for wrong predictions [15] on either the object's semantic class or the judgment of its in-distribution (ID) or OOD. Compared with the bulk of efforts for improving 3DSS and OOD detection accuracies, far less attention is paid to understanding when the agent is uncertain. With an intent towards real-world deployment, it is important to ask: *Can the model be aware of its unsureness? Can the model detect whether the category prediction is correct or not? Can the model detect whether the input sample is ID or OOD?*

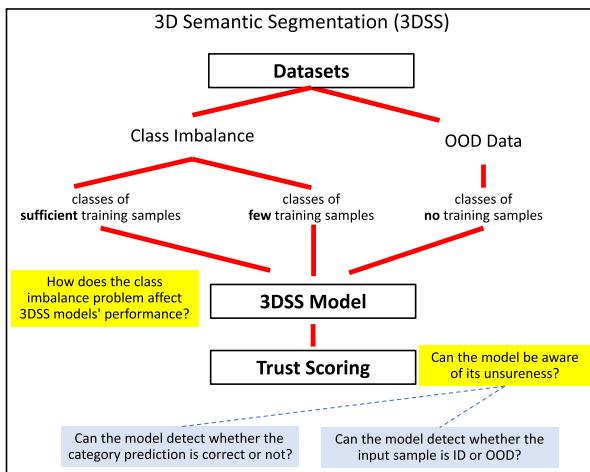


Fig. 1. A brief interpretation of the challenges of 3D semantic segmentation: The class imbalance problem and the existence of OOD data. We will explore how the class imbalance problem affects 3DSS performance and whether the model can detect the category prediction correctness, ID or OOD. (3DSS: 3D semantic segmentation, ID: in-distribution, OOD: out-of-distribution.)

Fig. 1 illustrates the key issues addressed in this research. With a focus on a deeper understanding of the challenges 3DSS models face with class imbalance and OOD data, two experiments are conducted to seek answers to the above questions. Experiment 1 studies 3DSS models' performance on a class-imbalanced dataset SemanticKITTI [16], where four 3DSS models, PointNet++ [17], Cylinder3D [18], RandLANet [19], and SqueezeSegV3 [20] are used and represent the popular and state-of-the-art models in the literature. Experiment 2 studies whether the model is aware of its unsureness when facing class imbalances and OOD data. Considering *people* and *rider* as OOD, the data from SemanticPOSS [21] are augmented to SemanticKITTI [16], and a new dataset AugKITTI is developed. Softmax confidence [12], data uncertainty and model uncertainty [22], [23], ODIN [24] and Mahalanobis distance [25] are used as the trust scores to predict whether the classification result is correct or wrong, or whether the data are ID or OOD. To the best of our knowledge, this is the first work to provide an in-depth analysis of how class imbalance and OOD data affect 3DSS model performances with insightful analyses and discussions.

Experimental results show that although the scale of the training data is a key factor, model performance could be greatly affected by intraclass diversity and interclass ambiguity. Some hard classes are found, that even with large training samples, have difficulty achieving a high classification accuracy or are easily confused with others. Facing the dual challenges of class imbalance and OOD, the model has difficulty predicting whether the classification result is correct or wrong or whether the data is ID or OOD. With the current trust scoring methods, a low trust score could be yielded by either OOD or ID for a wrong classification result, whereas a high trust score could also be given by an insufficiently trained model on wrong predictions.

The main research contributions of this work are as follows:

- 1) Experimental studies with both a quantitative analysis with cross-correlating metrics and a visualization

analysis of the feature space are conducted to gain a deeper understanding of the performance of the state-of-the-art 3DSS models processing class imbalanced and OOD data, which is crucial for real-world deployments.

- 2) A 3D LiDAR dataset augmentation method is developed. SemanticPOSS has rich data on dynamic objects, while SemanticKITTI describes mostly static scenes. A new dataset is generated by augmenting the data for the dynamic objects of SemanticPOSS to SemanticKITTI, which reduces dataset bias. Additionally, more realistic datasets for OOD studies are generated by exploiting the bias of the existing datasets.
- 3) New metrics are proposed to address the class imbalance issue in evaluating model performance, and we demonstrate that the traditional metrics are not sufficient and that the new metrics are required for appropriately evaluating 3DSS and OOD detection performance in cases where the class models are not sufficiently learnt.
- 4) We engage in an insightful discussion of the key issues to better understand the challenges of class imbalance and OOD data for the 3DSS task. We highlight potential topics for future works to improve the agent's awareness of the correctness or wrongness of the results and whether the data is ID or OOD.

The structure of this paper is as follows. Section II reviews the existing research for 3DSS models, class imbalance and the trust scores used for failure detection and OOD detection. Section III introduces the datasets and 3DSS models used for our experiments. Section IV provides an analysis of Experiment 1, which evaluates the performances of the 3D semantic segmentation models trained on a class-imbalanced dataset. Section V provides an analysis of Experiment 2 to explore the performances of the trust scores applied to class-imbalanced datasets, which is followed by a discussion on future topics and potential solutions in Section VI.

II. RELATED WORK

A. 3D Semantic Segmentation Models

3DSS has been extensively studied in the literature to gain a precise understanding of complex scenarios, such as autonomous driving and robotics applications. In recent years, great progress has been made in methods using deep learning techniques [7], [9], of which much of the pioneering work can be traced back to PointNet [26] and PointNet++ [17], which provides the base network architectures for 3DSS. 3D data can be represented in different formats. Based on these formats, 3DSS models scaled up to DNN models are generally divided into point-based, image-based and voxel-based methods [9]. Point-based methods take raw point clouds as input and output point wise labels. These types of methods, such as [27] and [28] [19] introduce special computation modules for local feature aggregation of 3D point clouds. Voxel-based methods [29] [18], [30] partition 3D space into a number of voxels to convert the point clouds into a structured data format and then apply an encoder-decoder architecture for feature extraction. In addition, image-based methods [31], [32] [20] project point clouds into 2D images and apply

2D semantic segmentation models to predict semantic labels. In the following experiments, we choose PointNet++ [17] to represent the earlier pioneering models and Cylinder3D [18], RandLA-Net [19], SqueezeSegV3 [20] to represent the state-of-the-art models.

B. Class Imbalance Problem

Class imbalance reflects that some specific classes have far fewer samples in the training data than others, limiting models' performance for these small classes. It is a common problem in training deep learning models for real-world applications and has been studied extensively in the literature [33], [34]. Many researchers have focused on analysing the imbalanced model performance caused by an imbalanced training data size [10], [35], [36], and methods such as data resampling [37], [38] or loss reweighting [39], [40] have been developed to alleviate the problem. These methods have also been used to address the class imbalance problem for 3DSS models [41].

As discussed in [9], the number and physical size of each object class vary in real-world scenarios. Therefore, in 3D LiDAR datasets, *road*, *building* and *plants* make up a large proportion of the data, whereas *people* and *rider* are rare and thus difficult to model. Furthermore, in 3D LiDAR sensing, the point density of the objects closer to the sensor is much higher than that of the objects farther away, which aggravates the class imbalance problem in 3D LiDAR datasets. Annotating the 3D LiDAR dataset is the result of a trade-off between labour cost and data size. Semantic categories cannot be defined in too much detail, thereby reducing the difficulty of manual annotation and ensuring that the amount of data in each category can meet the needs of model training. For this reason, some objects have the same semantic labels but different morphological shapes, while some objects are similar in data but have different semantics definitions. Therefore, in the current 3D LiDAR dataset, classes are imbalanced not only in terms of their data size but also in terms of their heterogeneous properties, a scenario that has not been studied in the literature.

C. OOD and the Open-World Learning Problem

Learning models for autonomous driving and robotics applications need to address the open-world problem [42], which requires the models to deal with both the seen (ID) and unseen (OOD) objects in the training datasets. OOD detectors [14] have been developed as an auxiliary module combined with the main task model for this purpose. Several mainstream methods for OOD detection have been developed. The methods are broadly divided into two groups based on whether they need an additional dataset that contains OOD examples. Outlier Exposure methods [43], [44] use an auxiliary dataset to represent OOD data when training models to teach the model better representations for OOD detection. However, for many applications, it is not possible to predefine OOD data, making these approaches unsuitable. Reconstruction methods [45], [46] learn to map the training data to the hypersphere of the feature space and maps the OOD data outside the hypersphere. However, the unseen OODs could be

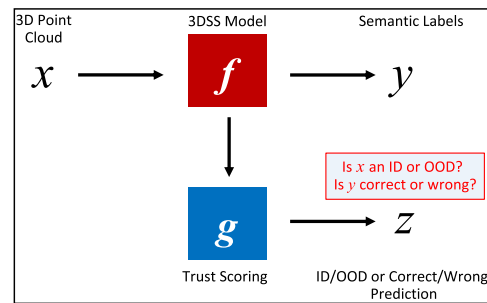


Fig. 2. Flow of 3DSS with OOD data. (3DSS: 3D Semantic Segmentation).

very diverse, and guaranteeing them outside the hypersphere remains an open issue. Trust scoring methods are the most popular methods and are studied in this research. They utilize a trust score metric to assess the reliability of the main task model's results and classifies the ID/OOD by thresholding the scores. Many trust score metrics have been developed in the literature [12], [24] [23], [25] [47], which will be detailed in the next subsection. However, in this type of approach, OOD detection is not an independent task, and the performance is strongly related to the capabilities of the main task model. Can OOD be detected by trust scoring the results of a 3DSS model that is trained on class imbalanced data? Research is needed.

D. Trust Scores

Knowing when a deep model's result is trustful and when the model is uncertain is of great importance in many safety-critical applications. To this end, many trust score metrics have been developed, such as Softmax confidence [12], ODIN [24], Mahalanobis distance (MD) [25], uncertainty [23] based on Monte Carlo dropout [48] and deep ensemble [47]. By thresholding these trust scores, OOD data [12], [24] [25] or model failures [49], [50] are detected.

In the literature, there are no universally recognized names for trust scores or for the thresholding methods on the trust scores. We borrow the word *trust score* from [49], and we name *trust scoring* for the group of threshold methods on trust scores. In Experiment 2 of the paper, we sequentially concatenate trust scoring with 3DSS to analyse the performance of the above trust scores on three tasks, namely, detecting ID/OOD, correct/wrong without OOD and correct/wrong with OOD. The focus is to understand the challenges when facing class imbalanced and OOD data.

III. METHODOLOGY

In this section, we introduce the flow of 3D semantic segmentation, datasets and the methods used for our experiments.

A typical 3DSS model takes 3D point clouds x as input, passes them to an end-to-end 3DSS model f , and outputs pointwise semantic labels y , as shown in Fig. 2. The 3DSS model requires a trust score g to judge whether the category prediction is reliable. The trust score function g takes the output probability vector or the feature vector given by model f as input and outputs the judgment z . In our experiments, we focus on exploring the different performances using different 3DSS models f and trust scores g .

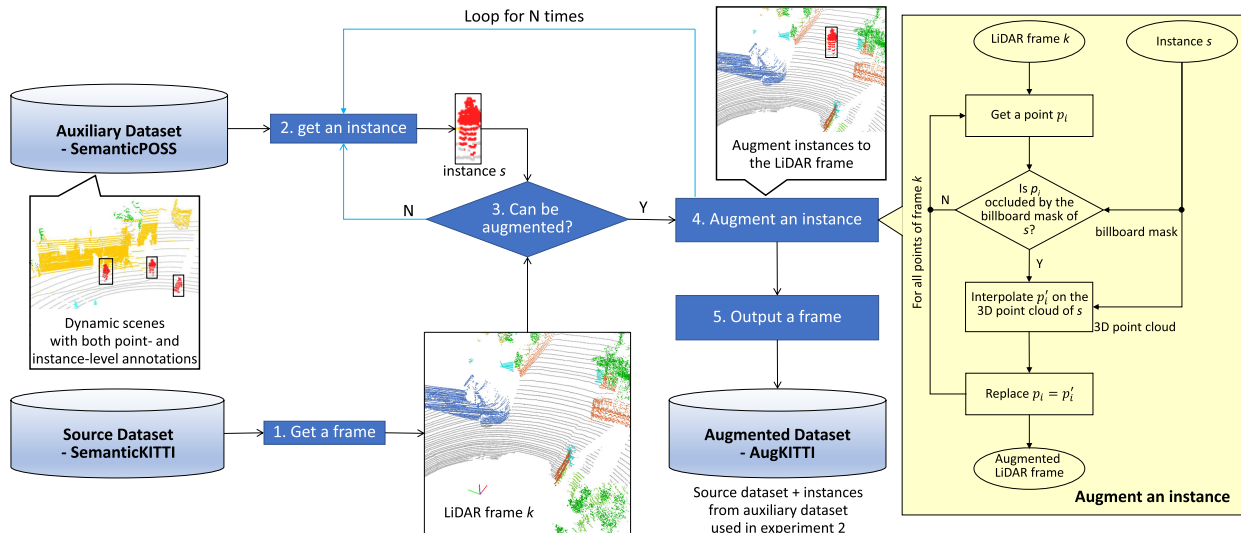


Fig. 3. The process to generate test frames by combining scenes from SemanticKITTI and *people*, *rider* instances from SemanticPOSS.

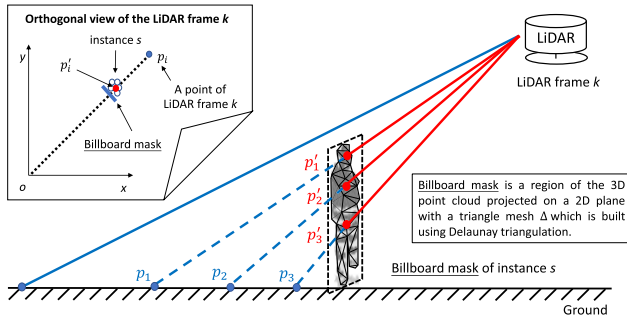


Fig. 4. Schematic diagram of the process to interpolate points on target LiDAR frames.

A. 3D Semantic Segmentation Models

We use four baseline models in our experiment, PointNet++ [17], RandLA-Net [19], Cylinder3D [18], and SqueezeSegV3 [20].

PointNet++ is a pioneering model in the field of deep neural networks used for 3D point cloud processing. We take PointNet++ as a representative traditional 3DSS model in our experiment.

RandLA-Net, Cylinder3D, and SqueezeSegV3 are recently proposed state-of-the-art methods for 3DSS tasks using different formats of input data. RandLA-Net is a point-based 3DSS model and uses random point sampling instead of a more complex point selection approach to reduce the computation and memory cost. Cylinder3D is a voxel-based 3DSS model that uses a cylindrical partition instead of a common rectangular partition when performing voxelization and extracts voxel features by a simplified PointNet. SqueezeSegV3 is an image-based model which takes range-image as input, and use spatially adaptive convolution (SAC) to dynamically adapt the filters to process different positions of the image. We take the three deep networks as representative state-of-the-art 3DSS models in our experiment.

B. Trust Scores

The trust scoring approach uses a specific score $g(x)$ to judge whether input x is an OOD sample or the category

prediction is wrong. If given a threshold value δ , the output of the trust scoring process z is given by:

$$z = \begin{cases} 0, & \text{if } g(x) \leq \delta \\ 1, & \text{if } g(x) > \delta \end{cases} \quad (1)$$

We use several widely used scores for failure detection and OOD detection in our experiment, Softmax confidence [12], uncertainty [23], ODIN [24] and MD [25].

The Softmax confidence $conf(x)$ is the output of the Softmax layer and is the most common score used to detect failure and OOD, and is given by:

$$conf(x) = \max_c p_c(x) = \max_c \frac{\exp(f_c(x))}{\sum_{i=1}^C \exp(f_i(x))} \quad (2)$$

where $p_c(x)$ is the prediction probability of class c , $f_c(x)$ is the last layer output of the network of class c and C is the number of classes.

Uncertainty is a score to evaluate how certain the model predictions are, which can also be used for failure and OOD detection. In addition, uncertainty can be divided into data uncertainty $du(x)$ and model uncertainty $mu(x)$ to distinguish between the uncertainty caused by data ambiguity and model disagreement. Data uncertainty can be quantized by prediction entropy, and model uncertainty can be quantized by mutual information [23], which are given by:

$$\begin{aligned} du(x) &= \mathbb{E}_M[H(\mathbf{p}(x))] = -\mathbb{E}_M\left[\sum_{c=1}^C p_i(x) \log p_c(x)\right] \\ mu(x) &= H(\mathbb{E}_M[\mathbf{p}(x)]) - \mathbb{E}_M[H(\mathbf{p}(x))] \end{aligned} \quad (3)$$

where H is the prediction entropy and \mathbb{E}_M is the expectation for all the models distributed in the model space, which can be estimated by Monte Carlo dropout [48] or deep ensemble [47].

ODIN [24] applies temperature scaling to the Softmax confidence using a temperature scaling parameter T :

$$temp(x) = \max_c \frac{\exp(f_c(x)/T)}{\sum_{i=1}^C \exp(f_i(x)/T)} \quad (4)$$

Mahalanobis distance [25] is a distance measure in feature space:

$$md(x) = \min_c (f(x) - \mu_c)^T \Sigma^{-1} (f(x) - \mu_c) \quad (5)$$

where μ_c is the mean vector of class c and Σ is the covariance matrix of the training samples in feature space.

In our experiment, we try to explore the failure and OOD detection performance of these scores for the 3DSS models trained on the class-imbalanced datasets.

C. Dataset Augmentation

Due to the difficulty in manual point-level labelling of 3D LiDAR point clouds, there is a lack of large-scale 3D LiDAR datasets used for 3DSS tasks [9]. Driving scene point clouds collected by vehicle-mounted LiDAR such as nuScenes [51], SemanticKITTI [16], and SemanticPOSS [21] are commonly used for understanding driving scenes. However, the above widely used public 3D datasets have a large difference in the data size of the different classes. Fig. 7 shows the data size of the popular and largest 3D LiDAR dataset, SemanticKITTI, which reflects the class imbalance problem.

We use SemanticKITTI as a representative 3D dataset in Experiment 1 to study how the class imbalance dataset affects the 3DSS performance. However, there is no publicly available dataset for a study of OOD in a 3DSS task. To find a dataset for Experiment 2 on whether the model is aware of its unsureness when facing class imbalanced and OOD data, a dataset augmentation method is developed.

SemanticKITTI contains few samples of *people* and *rider*, whereas SemanticPOSS describes scenes populated with these dynamic objects, reflecting two completely different scenarios and making it most feasible for us to use the two classes as OOD. Meanwhile, for autonomous driving system, *people* and *rider* are very critical to safety. Considering *people* and *rider* as OOD, the frames of SemanticKITTI that have no *people* and *rider* are extracted to compose a dataset **SubKITTI** for training, while the remaining frames of SemanticKITTI are augmented by the instances of *people* and *rider* of SemanticPOSS to generate a new dataset **AugKITTI** for testing.¹

Fig. 3 illustrates the flow of the dataset augmentation. SemanticPOSS is used as an auxiliary dataset that provides instances, while SemanticKITTI is a source dataset that provides LiDAR frames. An instance dataset is first generated by assembling all the instances of SemanticPOSS. Here, an instance is a 3D point cloud at the LiDAR sensor's coordinate system. Given a LiDAR frame k of SemanticKITTI, an instance s is sampled, and the 3D point cloud is projected to the LiDAR frame on their own coordinates. Augmentation can only be conducted if the projected space is on the road and not occupied by other objects.

Fig. 4 illustrates the procedure of augmenting an instance s with LiDAR frame k . For each instance, a billboard mask is generated describing a region of the projected 3D point cloud on a 2D plane, with a triangle mesh Δ which is built using Delaunay triangulation. Given LiDAR frame k and instance s ,

¹SubKITTI and AugKITTI are published at <http://www.poss.pku.edu.cn/augkitti.html>

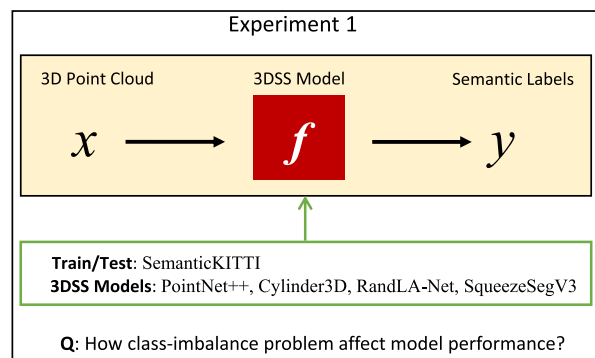


Fig. 5. Setup of Experiment 1: Evaluate the performances of 3DSS models trained on a class-imbalanced dataset.

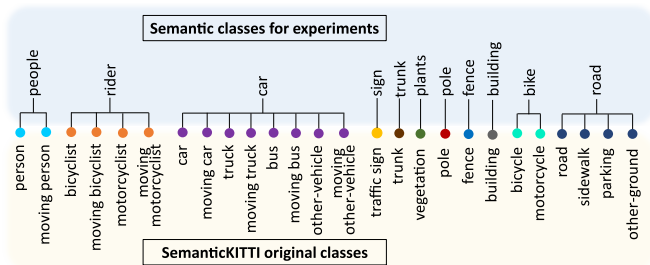


Fig. 6. Class definition for our experiments.

for any point p_i of the LiDAR frame, if its beam intersects the billboard mask and the intersection point lies inside a triangle $\delta_i \in \Delta$, then a new p'_i is estimated to replace p_i . The original object points of three vertexes of δ_i are retrieved, their range distances are linear interpolated to estimate a new range distance r_i , thus $p'_i = r_i \cdot \frac{p_i}{\|p_i\|}$ is estimated along the same beam direction of p_i .

It should be noted that p'_1 , p'_2 , and p'_3 in Fig. 4 are not necessarily in the same plane, nor in the plane of the billboard mask, because they are generated by interpolating the original coordinates corresponding to the triangular mesh, rather than the projected coordinates.

Although the data augmentation method is motivated by the OOD study in this research, it is a general method for generating augmented 3D point clouds, which can also be used to reduce the dataset bias between different 3D datasets. In addition, the proposed method can be applied to datasets with different sensor characteristics, e.g., transferring instances collected by a 32-line LiDAR to a 64-line LiDAR dataset. In this way, we create an augmented dataset with a certain number of *people*, *rider* samples and other classes have the same distribution as the training set. The data size of each class of AugKITTI is shown in Table II.

IV. PERFORMANCE OF 3DSS FACING CLASS IMBALANCED DATA

A. Experimental Setup

In this section, we evaluate the performances of 3D semantic segmentation models trained on a class-imbalanced dataset, as shown in Fig. 5. We train and test models on the SemanticKITTI dataset, and some morphologically similar classes are merged in the experiment, as shown in Fig. 6.

TABLE I

DATA SIZE OF EACH CLASS ON SEMANTICKITTI FOR TRAINING AND TESTING IN OUR EXPERIMENTS, AND TRAINING WEIGHTS OF EACH CLASS

scale	Large					Middle		Small			
class	road(ro)	plants(pl)	building(bu)	fence(fe)	car(ca)	trunk(tr)	pole(po)	sign(si)	bike(bi)	people(pe)	rider(ri)
SemanticKITTI (train)	730.27M	522.39M	268.33M	143.17M	98.19M	12.43M	5.78M	1.19M	1.12M	0.569M	0.386M
SemanticKITTI (test)	154.65M	145.94M	56.88M	12.64M	33.59M	5.51M	1.67M	0.381M	0.594M	0.477M	0.329M
Training weights	1.00	1.00	1.00	1.00	1.00	1.36	2.19	8.48	8.98	17.19	25.09

There are a total of 11 classes for model training, and we simply use the first two letters of class names to denote these classes. The data size of each class is shown in Table I. For a qualitative analysis, we divided the 11 classes into *large*, *middle*, and *small* classes according to the order of magnitude of each class data size on the training set.

Four 3D semantic segmentation models PointNet++, SqueezeSegV3, RandLA-Net and Cylinder3D are used in the experiment, and all the models are trained for 32 epochs using a weighted cross-entropy loss given by:

$$L = -\frac{1}{N} \sum_{c=0}^K \sum_{y_i=c} w_c h_i^T \log p_i \quad (6)$$

where N is the total number of samples used for the loss calculation, K is the number of classes, w_c is the weight of class c , y_i is the ground truth of sample i , h_i is the ground truth one-hot vector of sample i , and p_i is the output probability of sample i . We use class weights calculated by [40], which are given by:

$$w_c = \frac{1 - \beta}{1 - \beta^{N_c}} \quad (7)$$

where N_c is the data size of class c and $\beta = 0.9$ when training models. The weights of classes are shown in the last line of Table I. The Adam optimizer with a learning rate of 0.001 is used for network optimization.

B. Experimental Results

Here, the traditional evaluation metrics of **intersection over union (IoU)**, **precision (Pre)** and **recall (Rec)** are used to analyse and evaluate the model performance of each semantic category. Hereinafter, we denote $|y_{GT} = c|$ as the number of points whose ground truth (GT) labels are equal to class c and $|y_{PD} = c|$ as the number of points predicted (PD) to class c . The IoU, Pre and Rec of class c are estimated as follows:

$$\text{IoU}(c) = \frac{|y_{GT} = c \wedge y_{PD} = c|}{|y_{GT} = c| + |y_{PD} = c| - |y_{GT} = c \wedge y_{PD} = c|} \quad (8)$$

$$\text{Pre}(c) = \frac{|y_{GT} = c \wedge y_{PD} = c|}{|y_{PD} = c|} \quad (9)$$

$$\text{Rec}(c) = \frac{|y_{GT} = c \wedge y_{PD} = c|}{|y_{GT} = c|} \quad (10)$$

The model performance of each class and the number of data points in training are shown in Fig. 7(a). The state-of-the-art methods significantly improve the overall performance of the model compared to the earlier method PointNet++. From Fig. 7(b), it can be found that the classes can be divided

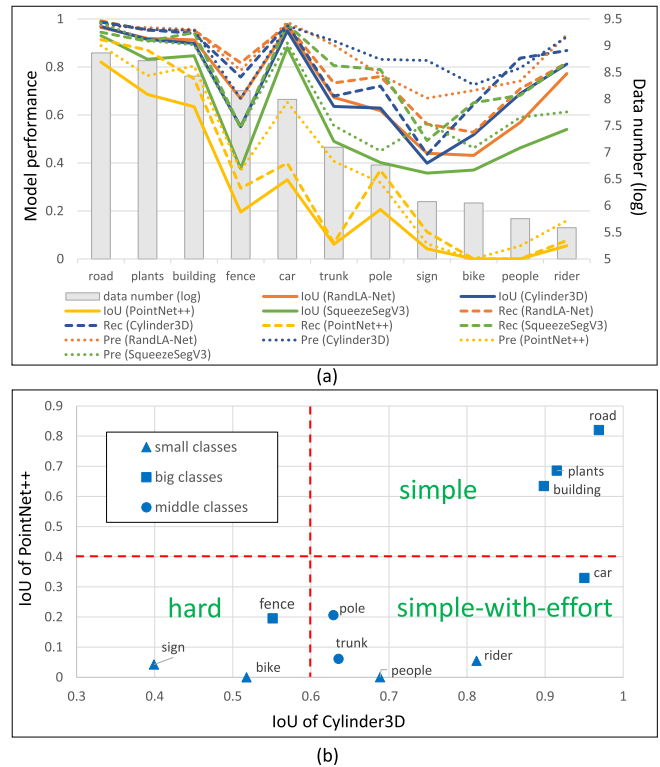


Fig. 7. (a) Training data size and model performances on SemanticKITTI dataset. (b) Scatter plot of IoU of PointNet++ and Cylinder3D, which divides the classes into three groups.

into three groups. The classes in the first group demonstrate excellent performance in all the models, which are marked as *simple* classes, and interestingly, they are all the large-scale classes.

The classes in the second group achieve great performance improvements in the state-of-the-art models, which are marked as *simple-with-effort* classes. In this group, *people* and *rider* are small-scale classes, *pole* and *trunk* are middle-scale classes, and *car* is a large-scale class. Interestingly, these classes are objects that have regular sizes and shapes. This may be the reason why they outperform other classes with more data samples.

The classes in the third group are hard ones, which demonstrate unsatisfactory performances in all models, and are marked as *hard* classes. This group contains the small-scale classes *bike* and *sign*, and the large-scale class *fence*. The models of these classes are hard to be learnt no matter on a small or large set of training data, and fewer performance improvements are found in the models.

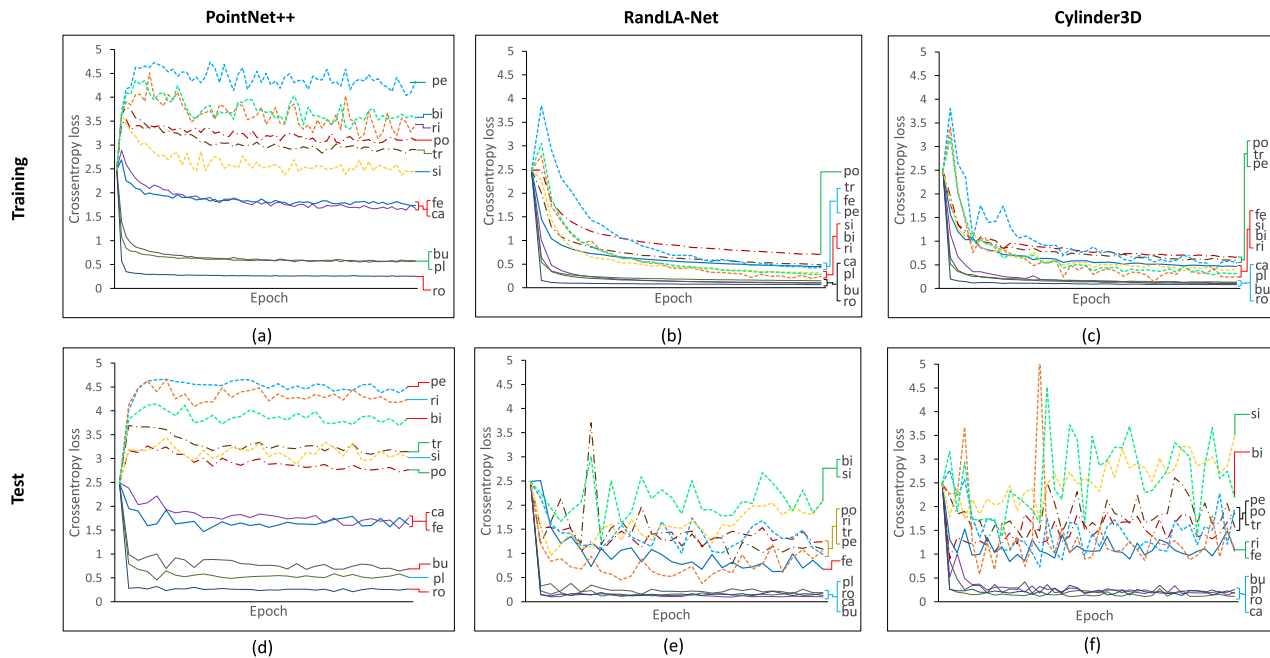


Fig. 8. Training loss curve of each class using (a) PointNet++, (b) RandLA-Net, (c) Cylinder3D. And test loss curve of each class using (d) PointNet++, (e) RandLA-Net, (f) Cylinder3D. The abbreviations of class names are defined in Table I.

We also find the influence of class imbalance reflected by the loss curve. We visualize the cross-entropy loss curve of class c given by:

$$L(c) = -\frac{1}{N_c} \sum_{y_i=c} \mathbf{h}_i^T \log \mathbf{p}_i \quad (11)$$

where N_c is the data size of class c , y_i is the ground truth of sample i , \mathbf{h}_i is the ground truth one-hot vector of sample i and \mathbf{p}_i is the output probability of sample i . As shown in Fig. 8, we discriminate between semantic classes by the colour and different data scales of the classes by line type. We find that the training loss curve of the simple classes converges faster and lower than the hard classes. In the test loss curves, we observe that simple classes retain a low and smooth loss value, whereas the curves of simple-with-effort and hard classes are relatively high and zigzag. Compared with the test loss of PointNet++, the loss of simple-with-effort classes *people*, *rider*, *car* are evidently reduced when using RandLA-Net or Cylinder3D, whereas the loss of hard classes are not significantly reduced, indicating the learning difficulty of these classes.

Based on the above results and observations, we can find that, in addition to the size of the training data, there are other factors that may have a significant impact on the performance of the model, which may be in terms of the basic properties of each semantic category. These basic properties are identified as the confusion property in Section IV-C, and further identified as the intra-class diversity and interclass ambiguity properties in Section IV-D.

C. Confusion Analysis

Fig. 9 shows the confusion matrixes of the three models. Each value $p(r, c)$ on row r and column c of the confusion

matrix is estimated by:

$$p(r, c) = \frac{|y_{GT} = r \wedge y_{PD} = c|}{|y_{GT} = r|} \quad (12)$$

which indicates the ratio of points whose ground truth labels are r but classified to c . Therefore, the diagonal values $p(c, c)$ equal the recall for class c , i.e., $\text{Rec}(c) = p(c, c)$. In addition to the diagonal values, the values in each row r compose a vector of **wrong prediction ratios (WPR)** that describes how data points of the ground truth class r are misclassified to other classes $c \neq r$. Similarly, the values in each column c compose a vector of **be confused ratio (BCR)** that describes how data points of other ground truth class $r \neq c$ are misclassified to the predictive class c .

With the confusion matrix of Cylinder3D in Fig. 9(c), the WPR and BCR vectors are generated and shown in Fig. 10(a-b). The class *plants* demonstrates excellent performance on IoU, Pre and Rec in all models in Fig. 7, and its data points have a very small part misclassified to other classes, as shown in Fig. 10(a). However, interestingly, the data points of the other classes have a large ratio predicted to *plants* by mistake, meaning that *plants* is a class that is easily confused. This tendency for imprecision has been omitted in the literature. To this end, this research proposes a new metric called **weighted precision (wPre)** by reshaping the precision metric as follows.

Let $\sim \text{Pre} = 1 - \text{Pre}$ be the negation of precision; we have:

$$\sim \text{Pre}(c) = \frac{\sum_{r \neq c} |y_{GT} = r \wedge y_{PD} = c|}{\sum_r |y_{GT} = r \wedge y_{PD} = c|} \quad (13)$$

For a small-scale class r , even the wrong predictions occupy large proportions in its ground truth set, and the absolute point number is small compared to larger classes; hence, it has little impact in the evaluation. To balance the impacts from the

PointNet++												RandLA-Net												Cylinder3D																
PD	pe	ri	ca	si	tr	pl	po	fe	bu	bi	ro	Rec	PD	pe	ri	ca	si	tr	pl	po	fe	bu	bi	ro	Rec	PD	pe	ri	ca	si	tr	pl	po	fe	bu	bi	ro	Rec		
pe	0.00	0.03	0.11	0.01	0.03	0.40	0.17	0.01	0.06	0.00	0.17	0.00	pe	0.71	0.01	0.02	0.00	0.01	0.15	0.00	0.01	0.06	0.00	0.01	0.71	pe	0.84	0.02	0.01	0.00	0.00	0.07	0.00	0.01	0.03	0.01	0.01	0.84		
ri	0.00	0.08	0.31	0.01	0.02	0.22	0.11	0.00	0.00	0.00	0.25	0.08	ri	0.11	0.82	0.02	0.00	0.00	0.03	0.00	0.00	0.00	0.01	0.01	0.82	ri	0.06	0.87	0.01	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.02	0.87		
ca	0.00	0.00	0.40	0.00	0.00	0.31	0.00	0.06	0.03	0.00	0.20	0.40	ca	0.00	0.00	0.97	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.01	0.97	ca	0.00	0.00	0.98	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.91		
si	0.00	0.01	0.03	0.11	0.05	0.42	0.17	0.01	0.06	0.00	0.13	0.11	si	0.01	0.00	0.00	0.56	0.02	0.10	0.26	0.02	0.03	0.00	0.00	0.56	si	0.01	0.00	0.00	0.44	0.01	0.19	0.22	0.11	0.02	0.00	0.00	0.44		
tr	0.00	0.00	0.06	0.02	0.07	0.62	0.07	0.01	0.05	0.00	0.11	0.07	tr	0.00	0.00	0.00	0.00	0.73	0.21	0.01	0.00	0.03	0.00	0.01	0.73	tr	0.00	0.00	0.00	0.00	0.68	0.30	0.00	0.00	0.00	0.00	0.01	0.68		
pl	0.00	0.00	0.02	0.00	0.00	0.87	0.00	0.01	0.03	0.00	0.06	0.87	pl	0.00	0.00	0.00	0.00	0.00	0.95	0.00	0.01	0.01	0.00	0.02	0.95	pl	0.00	0.00	0.00	0.00	0.00	0.96	0.00	0.02	0.00	0.00	0.02	0.96		
po	0.00	0.01	0.03	0.01	0.05	0.35	0.37	0.01	0.03	0.00	0.13	0.37	po	0.00	0.00	0.01	0.03	0.05	0.09	0.76	0.01	0.02	0.00	0.02	0.76	po	0.00	0.00	0.01	0.01	0.07	0.12	0.72	0.02	0.03	0.00	0.02	0.72		
fe	0.00	0.00	0.05	0.00	0.00	0.30	0.00	0.29	0.26	0.00	0.08	0.29	fe	0.00	0.00	0.00	0.00	0.00	0.08	0.00	0.82	0.08	0.00	0.02	0.82	fe	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.09	0.00	0.00	0.76	0.14	0.01	0.76
bu	0.00	0.00	0.00	0.00	0.00	0.21	0.00	0.02	0.75	0.00	0.01	0.75	bu	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.01	0.95	0.00	0.01	0.95	bu	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.02	0.94	0.00	0.00	0.94		
bi	0.00	0.01	0.18	0.01	0.01	0.60	0.03	0.01	0.03	0.00	0.14	0.00	bi	0.00	0.00	0.07	0.00	0.00	0.35	0.00	0.01	0.02	0.53	0.02	0.53	bi	0.00	0.00	0.06	0.00	0.00	0.24	0.00	0.03	0.01	0.64	0.02	0.64		
ro	0.00	0.00	0.02	0.00	0.00	0.06	0.00	0.01	0.01	0.00	0.91	0.91	ro	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.99	0.99	ro	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.99	0.99		
wPre	0.14	0.57	0.33	0.64	0.30	0.20	0.40	0.65	0.57	0.00	0.42		wPre	0.85	0.98	0.89	0.94	0.89	0.48	0.73	0.90	0.79	0.96	0.89		wPre	0.91	0.98	0.91	0.97	0.89	0.47	0.76	0.77	0.80	0.95	0.90			

Fig. 9. Confusion matrix of (a) PointNet++, (b) RandLA-Net, (c) Cylinder3D. The labels of each row represent the ground truth, and the labels of each column represent the prediction results. Recall (Rec) is the diagonal and weighted precision (wPre) is the diagonal dividing the sum of each column. (GT: ground truth, PD: predictions, and the abbreviations of class names are defined in Table 1.)

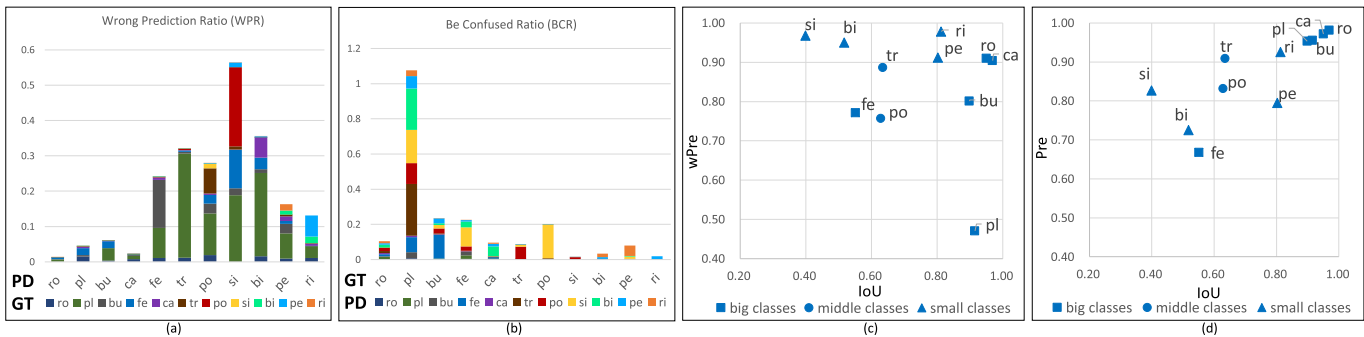


Fig. 10. (a) Wrong prediction to others classes using Cylinder3D. (b) Wrong prediction from other classes using Cylinder3D. (c) wPre vs. IoU scatter plot using Cylinder3D training and testing on SemanticKITTI. (d) Pre vs. IoU scatter plot using Cylinder3D training and testing on SemanticKITTI. (GT: ground truth, PD: predictions, and the abbreviations of class names are defined in Table 1.)

classes of different scales, a new metric is developed by a weighting on the data size of the ground truth class.

$$\begin{aligned}
 \sim \text{wPre}(c) &= \frac{\sum_{r \neq c} |y_{GT} = r \wedge y_{PD} = c|}{\sum_r |y_{GT} = r \wedge y_{PD} = c|} \\
 &= \frac{\sum_{r \neq c} p(r, c)}{\sum_r p(r, c)} = \sum_{r \neq c} \frac{1}{\eta_c} p(r, c) \quad (14)
 \end{aligned}$$

where $\eta_c = \sum_r p(r, c)$ is a factor to normalize each column vector of the confusion matrix to 1, and $\sim \text{wPre}(c)$ is the sum of the normalized non-diagonal values in column c , which describes how easily class c can be confused. Similarly, we have:

$$\text{wPre}(c) = 1 - [\sim \text{wPre}(c)] = \frac{1}{\eta_c} p(c, c) \quad (15)$$

By cross correlating IoU with the weighted precision metrics wPre of each class in Fig. 10(c), three groups are shown in these classes, high-accuracy and hard to be confused, high-accuracy but easy to be confused, and low-accuracy but hard to be confused. For comparison, Fig. 10(d) cross correlates IoU with the traditional precision metrics Pre, which failed to reflect such a property.

D. Feature Analysis

Model performances are relevant to the feature description of models. Some experimental results using Cylinder3D are

notable; large class *plants* is easier to be confused compared with other large classes, while small classes *people* and *rider* have a relatively satisfactory performance. Therefore, we visualize the last layer features using t-distributed stochastic neighbor embedding (T-SNE) to analyse feature distribution, as shown in Fig. 11.

We find that the proposed wPre metric can appropriately evaluate the feature confusion in the presence of the class imbalance. As mentioned in the previous sections, the IoU, Pre and Rec of class *plants* are all high. However, the features of plants in Fig. 11 are severely confused with other class features, which is not reflected by traditional metrics. In contrast, the wPre of *plants* is very low because of the reweighting of the precision metric by class data size, which shows a better estimation of feature confusion.

In Fig. 11, confusion of the different classes can be easily observed; we find that the points of large classes *road*, *car* distribute far away from other classes, while *fence*, *plants* are confused. This result can explain the performance differences among classes in our experiment and the influence of class imbalance. The feature confusion of the different classes also reflects the classification confusion in Fig. 10(a-b).

Two noticeable phenomena, intraclass diversity and interclass ambiguity, are reflected in Fig. 11. Some classes have various features, causing high intraclass diversity and learning difficulty. A prominent example is the feature distribution of *plants* in Fig. 11, where we find that feature points of *plants*

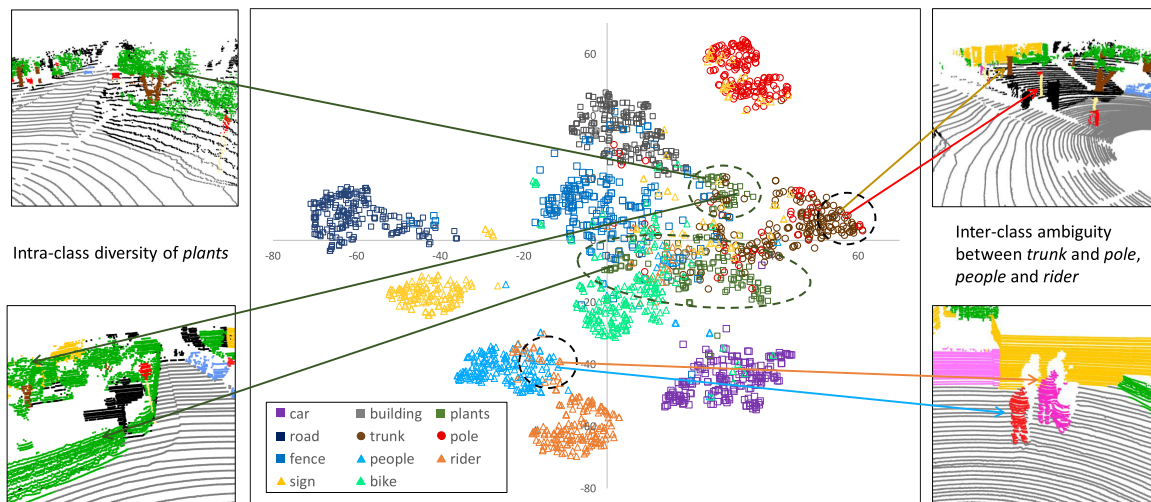


Fig. 11. T-SNE plot of 200 random sampled points from SemanticKITTI test frames using Cylinder3D. Some classes show intraclass diversity such as *plants*. And some classes are confused due to the interclass ambiguity, such as *trunk* and *pole*, *people* and *rider*.

can be divided into two parts. The right-up part corresponds to the points of the crown, and the other part corresponds to the roadside shrub. The morphological features of *plants* can be extremely varied, which leads to a high intraclass diversity, and other classes that are not learned sufficiently are more likely to be misclassified to *plants*. By contrast, classes such as *road*, *people*, and *rider* have relatively consistent features, making them easier to learn.

On the other hand, some of the different classes have similar features, causing interclass ambiguity and feature confusion. As shown in the right part of Fig. 11, two feature points of *trunk* and *pole* are close in feature space due to their similar columnar features. Similarly, *people* and *rider* show very similar geometrical features and are close in the feature space. The reason for confusion is the interclass ambiguity, namely, the feature similarity of these classes.

From the feature space analysis, we can give a systematic summary. Some classes have various features, while some different classes have similar features, causing these samples to be misclassified or hard to distinguish from other classes. The intraclass diversity and interclass ambiguity determine the learning difficulty of the classes and greatly affects model performance.

V. PERFORMANCE OF TRUST SCORES FACING CLASS IMBALANCED AND OOD DATA

A. Experimental Setup

In this section, we evaluate the failure and OOD detection performances of trust scores using 3DSS models trained on class-imbalanced datasets, as shown in Fig. 12. We train models on the SubKITTI dataset and test models on the AugKITTI dataset. The data size of each class is shown in Table II. Compared with Experiment 1, *people* and *rider* are considered to be OOD classes.

Four 3D semantic segmentation models PointNet++, SqueezeSegV3, RandLA-Net, and Cylinder3D are used in the experiment, and all the models are trained for 32 epochs using the weighted cross-entropy loss given by Equation (6) and the

Adam optimizer with a learning rate of 0.001. In addition, we use the class weights given by Equation (7) with $\beta = 0.9$ when training the models, which are shown in the last line of Table II.

Five trust scoring approaches, Softmax confidence, data uncertainty, model uncertainty, ODIN and MD, are used in the experiment. For data uncertainty and model uncertainty, we use the Monte Carlo dropout [48] with the number of forward passes $M = 5$ and the dropout probability $p = 0.25$ as a commonly used setup [52]. For ODIN, we use the temperature parameter $T = 1000$.

B. Experimental Result

Given a 3DSS model f and a trust score g , a $g(x) \in [0, 1]$ can be estimated for each data point x based on the output of f . Ideally, a high $g(x)$ indicates that the model f is confident in its results, while a lower value indicates that the model is uncertain about its results. Discriminating whether the predicted semantic class is correct or wrong, or whether the data are ID or OOD is a binary decision, which has usually been made by thresholding $g(x)$ using formula (1).

Before examining the experimental results, let us first define three class sets: ID/correct ($\mathcal{A}_{ID.co}$), ID/wrong ($\mathcal{A}_{ID.wr}$) and OOD (\mathcal{A}_{OOD}). $\mathcal{A}_{ID.co}$ and $\mathcal{A}_{ID.wr}$ contain both of the ID classes that appeared in the training data. For each predicted class c_{PD} , the former has a single class $\mathcal{A}_{ID.co} = \{c_{GT} = c_{PD}\}$, while the latter are the rest $\mathcal{A}_{ID.wr} = \{c_{GT} \neq c_{PD}\}$. \mathcal{A}_{OOD} are OOD classes that are not known in training and thus are not included in the predicted label set. In this experiment, the OOD classes are *rider* and *people*.

This study addresses three tasks that differ only in the definition of their true and false class sets. Task 1 - I/O, discriminating whether the data are ID or OOD. For this task, we define the true class set as $\mathcal{A}^{I/O} = \mathcal{A}_{ID.co} \cup \mathcal{A}_{ID.wr}$ and the false class set as $\bar{\mathcal{A}}^{I/O} = \mathcal{A}_{OOD}$. Task2 - C/W and Task3 - C/W with OOD, discriminating whether the predicted semantic class is correct or wrong, where the two tasks vary in whether OOD are addressed. Both tasks share the same true

TABLE II
DATA SIZE OF EACH CLASS ON SUBKITTIT AND AUGKITTIT, AND TRAINING WEIGHTS OF EACH CLASS

scale	Large					Middle		Small		OOD	
class	road(ro)	plants(pl)	building(bu)	fence(fe)	car(ca)	trunk(tr)	pole(po)	sign(si)	bike(bi)	people(pe)	rider(ri)
SubKITTIT (train)	491.66M	385.99M	159.58M	107.45M	55.32M	7.23M	3.59M	0.801M	0.644M	0	0
AugKITTIT (test)	42.48M	36.07M	22.62M	15.98M	2.71M	0.479M	0.447M	0.082M	0.027M	2.49M	1.21M
Training weights	1.00	1.00	1.00	1.00	1.00	1.87	3.17	12.36	15.33	/	/

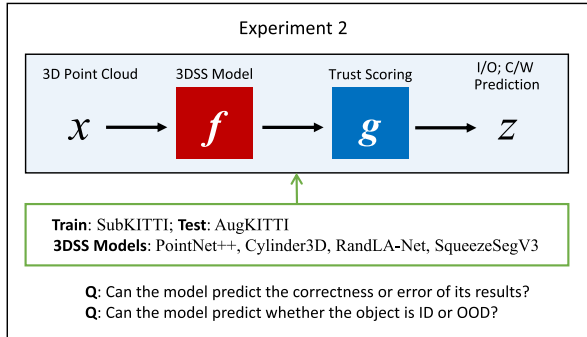


Fig. 12. Setup of Experiment 2: Evaluate the performances of trust scores using 3DSS models trained on class-imbalanced datasets. (I/O: ID/OOD. C/W: Correct/Wrong. SubKITTIT: SemanticKITTIT without OOD data. AugKITTIT: SemanticKITTIT with augmented OOD data. OOD data: people, rider.)

class set that is $\mathcal{A}^{C/W} = \mathcal{A}_{ID.co}$. They have different false class sets, which are $\bar{\mathcal{A}}^{C/W} = \mathcal{A}_{ID.wr}$ and $\bar{\mathcal{A}}^{C/W \text{ with } OOD} = \mathcal{A}_{ID.wr} \cup \mathcal{A}_{OOD}$ for Task 2 and Task 3, respectively.

Here, the traditional evaluation metric of the **area under the receiver operating characteristic curve (AUROC)** is used to analyse and evaluate the trust scoring performance. A receiver operating characteristic (ROC) curve is first plotted with the **true positive ratio (TPR)** and **false positive ratio (FPR)** for the vertical and horizontal axes, and the area under the ROC curve is estimated to evaluate the performance. For a predicted class c and a given threshold δ , TPR and FPR are estimated as follows:

$$\text{TPR}(c, \delta) = \frac{\text{TP}(c, \delta)}{\text{TP}(c, \delta) + \text{FN}(c, \delta)} \quad (16)$$

$$\text{FPR}(c, \delta) = \frac{\text{FP}(c, \delta)}{\text{FP}(c, \delta) + \text{TN}(c, \delta)} \quad (17)$$

Given the true and false class sets, \mathcal{A}^* and $\bar{\mathcal{A}}^*$, where $*$ represents the task, we have:

$$\text{TP}(c, \delta) = \sum_{r \in \mathcal{A}^*} |y_{GT} = r \wedge y_{PD} = c \wedge g(x) > \delta| \quad (18)$$

$$\text{TN}(c, \delta) = \sum_{r \in \bar{\mathcal{A}}^*} |y_{GT} = r \wedge y_{PD} = c \wedge g(x) \leq \delta| \quad (19)$$

$$\text{FP}(c, \delta) = \sum_{r \in \bar{\mathcal{A}}^*} |y_{GT} = r \wedge y_{PD} = c \wedge g(x) > \delta| \quad (20)$$

$$\text{FN}(c, \delta) = \sum_{r \in \mathcal{A}^*} |y_{GT} = r \wedge y_{PD} = c \wedge g(x) \leq \delta| \quad (21)$$

The AUROC of all 3DSS models f , and the trust scores g , on three different tasks are plotted in Fig. 13. We discriminate between the models by colour, trust scores by line type and tasks by subfigures. It can be found that for each task, lines

of the same colour are cluttered, revealing that the model performance is the dominant factor as compared to the trust score methods.

For each model, all trust scoring methods demonstrate a similar overall performance. The performance of PointNet++ for each class is related to data size to some extent, where there are no data in *sign* and *bike* from PointNet++ because none of the test points is predicted to be *sign* and *bike* by PointNet++. The general tendency is that the larger the data size, the higher the AUROC value, which means that OOD detection also suffers from the class imbalance.

On the other hand, the state-of-the-art models, SqueezeSegV3, Cylinder3D and RandLA-Net, have greatly improved the performance of all classes. However, the performance is not equivalent. For example, *sign* and *bike* are both small classes and have very poor performance on PointNet++. Although *sign* presents superior performance on Cylinder3D, the improvement of *bike* is limited, and the AUROC values on all three tasks are unsatisfactory. This phenomenon suggests that in these tasks, different properties of the classes may have a greater impact on performance than the data size imbalances. Some classes are hard because they are easily confused with OOD data or other classes.

C. Confusion Analysis

In this section, we choose the results of Cylinder3d with the Softmax confidence for further in-depth analysis as shown in Fig. 15. We begin by examining the distribution of the $g(x)$ values of each predicted category.

1) *Trust Score Distribution (TSD)*: Given a sequence of monotonically increasing values $\{\delta_0, \dots, \delta_i, \dots, \delta_n\}$ with $\delta_0 = 0$ and $\delta_n = 1$, a measure is defined below:

$$q(r, c, \delta_i) = \frac{|y_{GT} = r \wedge y_{PD} = c \wedge \delta_i < g(x) \leq \delta_{i+1}|}{|y_{GT} = r|} \quad (22)$$

The numerator of the right-hand term counts the number of points whose ground truth label is r while classified to c with a trust score $g(x) \in (\delta_i, \delta_{i+1}]$. In fact, $\sum_i q(r, c, \delta_i) = p(r, c)$.

Corresponding to the confusion matrix in Fig. 9, each column vector of a predicted class c can be extended to a **trust score distribution (TSD)** matrix, where each row r is for a ground truth class, each column is for δ_i , each matrix value is $q(r, c, \delta_i)$, and the sum of row r is $p(r, c)$. Taking PD = *plants* as an example, a matrix is shown in Fig. 14(a) and plotted in the mode of the stacked area chart in Fig. 14(b), where horizontal bands of a TSD are ordered. From top to bottom, they are sequentially the classes of $\mathcal{A}_{ID.co}$, $\mathcal{A}_{ID.wr}$ and \mathcal{A}_{OOD} . The TSD of each predicted class c is plotted

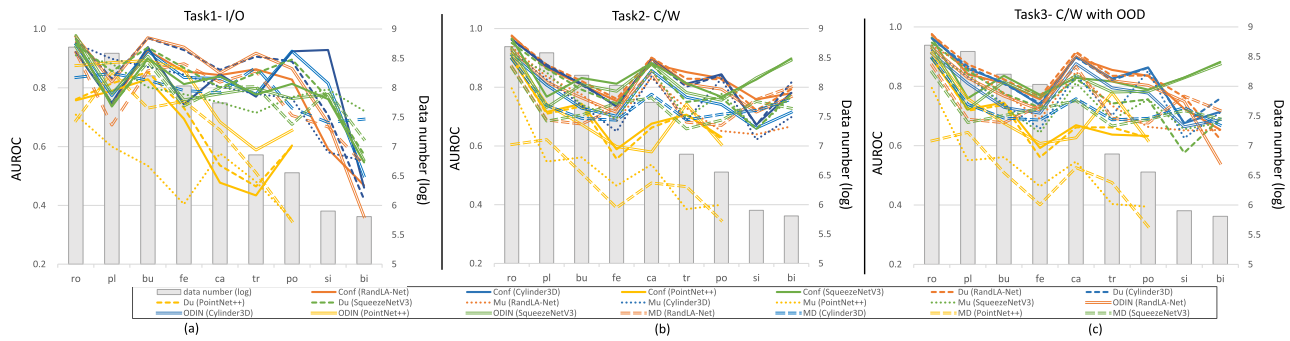


Fig. 13. AUROC of different models and trust scores for (a) Task1 - I/O, (b) Task2 - C/W, (c) Task3 - C/W with OOD.

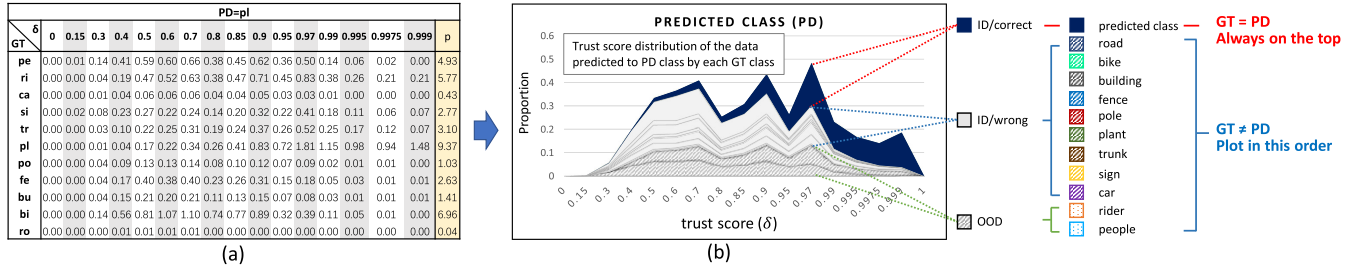


Fig. 14. (a) An example of a trust score distribution (TSD) matrix ($\times 10^{-1}$) (b) An example of the TSD according to the matrix. From top to bottom, the horizontal bands are sequentially the classes of $\mathcal{A}_{ID,co}$, $\mathcal{A}_{ID,wr}$ and \mathcal{A}_{OOD} .

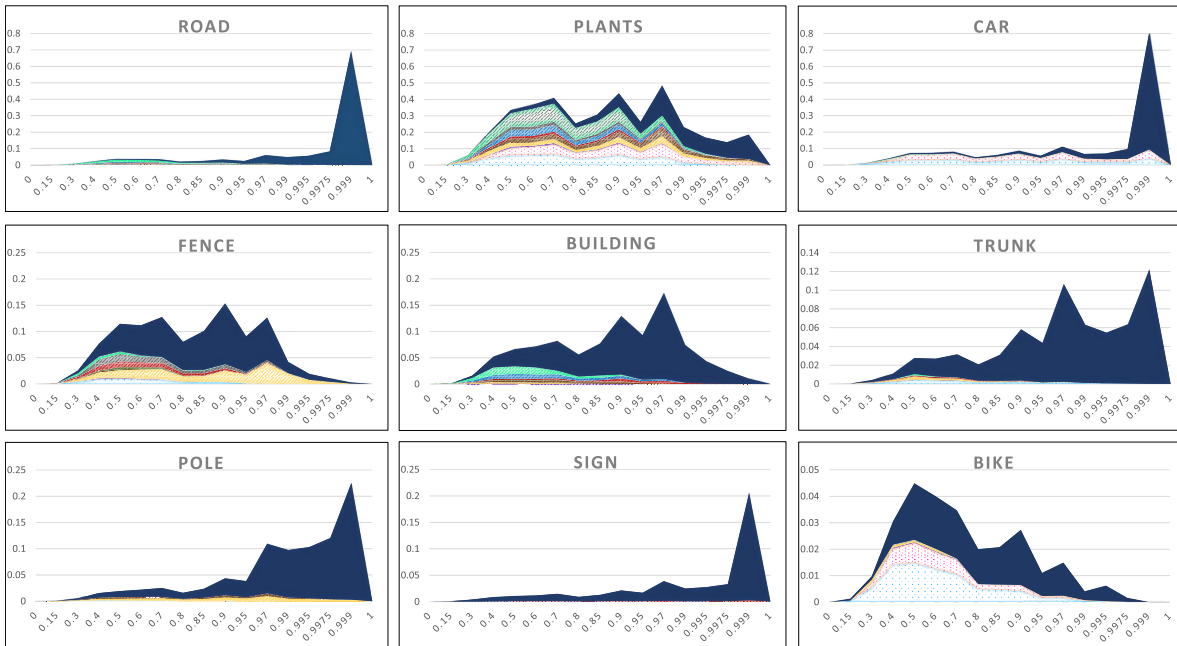


Fig. 15. TSD of 9 ID classes using Cylinder3D and Softmax confidence. *Road* and *sign* demonstrate perfect examples, whereas *bike*, *plants*, *fence* and *building* demonstrate examples of the worst performance.

in Fig. 15. Each horizontal band of a TSD depicts an $g(x)$ distribution for a certain ground truth class r and predicted class c with the area equal to $p(r, c)$.

In Fig. 15, *road* and *sign* demonstrate perfect examples, where the data predicted for these classes almost fall in the top horizontal band, $\mathcal{A}_{ID,co}$ data. The TSD of *car* looks similar. However, there are many OOD data predicted as *car*, which fall in the bottom horizontal bands, and some have very high trust scores. The performance of *trunk* and *pole*

are not bad too, and they are almost $\mathcal{A}_{ID,co}$ data, while their trust scores span wider ranges. *Bike*, *plants*, *fence* and *building* demonstrate examples of the worst performance, where many $\mathcal{A}_{ID,co}$ data have lower trust scores than those of $\mathcal{A}_{ID,wr}$ and \mathcal{A}_{OOD} . With such distributions, it is difficult to discriminate between ID/OOD or correct/wrong by thresholding on the trust scores.

2) *Per-Class AUROC for Three Tasks*: We extract the results of Cylinder3d with a Softmax confidence in Fig. 16

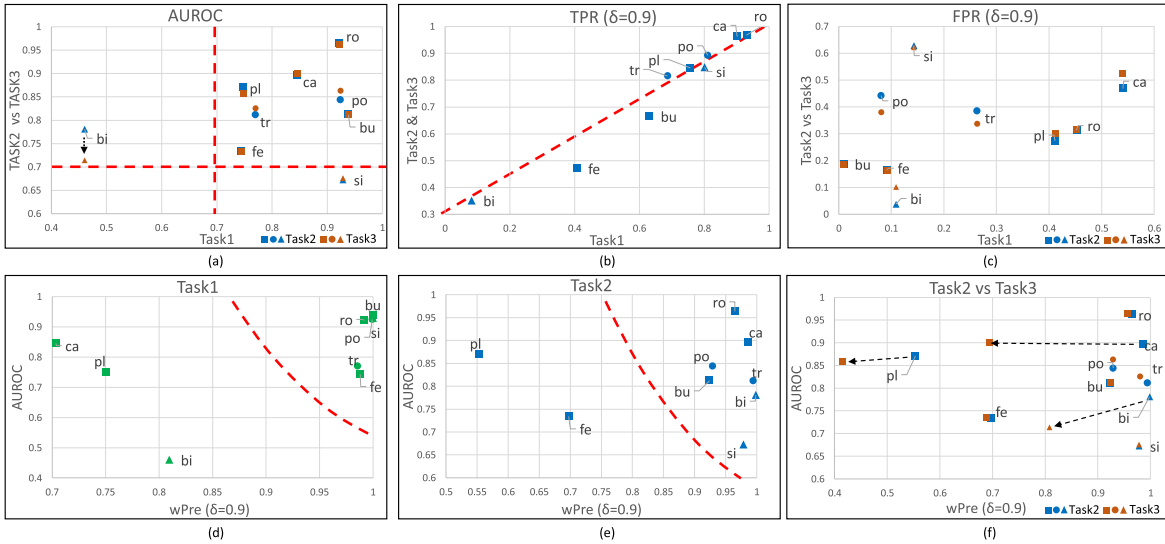


Fig. 16. (a) per-class AUROC of the three tasks. (b) per-class TPR with $\delta = 0.9$ of the three tasks. (c) per-class FPR with $\delta = 0.9$ of the three tasks. Per-class AUROC and wPre for (d) Task1, (e) Task2, (f) Task2 vs Task3. The abbreviations of class names are defined in Table II.

and cross-correlate the per-class AUROC of the three tasks in Fig. 16(a). The horizontal axis is for Task 1 - I/O, while the vertical axis compares Task 2 - C/W and Task 3 - C/W with OOD. Two dotted lines are manually drawn at AUROC=0.7 for analysis. It can be found that for the data predicted as *bike*, it is difficult to discriminate whether it is an ID or OOD. Comparing its performance on Task 2 and Task 3, it is found that *bike* is easily affected by OOD. In fact, the data of ID class *bike* and OOD class *rider* have similar properties, which makes them confusing. This phenomenon suggests that the performance of OOD detection and the influence of OOD data need to be addressed with the confusion properties of the classes.

On the other hand, for those predicted as *sign*, it is difficult to discriminate whether the predicted semantic class is correct or wrong, no matter whether OOD exists. However, this result contradicts what we discussed earlier, where *sign* demonstrates perfect TSD examples. We further analyse the results below.

3) *Threshold on Trust Scores*: An ideal trust score should provide a reliable prediction when it is high. Therefore, we are also interested in how model performance is influenced by an incorrect prediction and an OOD with high thresholding trust scores δ . We analyse TPR and FPR with $\delta = 0.9$ as the basis of a trust score evaluation. Fig. 16(b) shows the TPR results.

Since Task 2 and Task 3 share the same true class set, their per-class TPRs are the same, exhibiting a nearly linear trend. Similar to the results of TSD, *bike*, *fence* and *building* are the worst three performers, while *road*, *car*, *pole* and *sign* are among the top groups. However, the FPR results in Fig. 16(c) also contradict what we discussed earlier, where *bike*, *fence* and *building* have very low FPRs for all three tasks, whereas *road* and *sign* have high values. By examining the formula (20), we found that some classes have very small FP and TN, as exhibited in Fig. 15, and even a small FP could yield a high FPR. This is the main reason for the contradictory results in Fig. 16(a) and (c). If classes are highly imbalanced, the metrics TPR and FPR may not properly and sufficiently

evaluate the model performance of each class, which shows the need to combine them with a metric on precision.

4) *Per-Class AUROC and wPre*: By incorporating the trust score, the wPre metrics in the previous section are extended below to evaluate the reliability of a model prediction.

$$\text{wPre}(c, \delta) = \frac{\text{wTP}(c, \delta)}{\text{wTP}(c, \delta) + \text{wFP}(c, \delta)}. \quad (23)$$

where

$$\text{wTP}(c, \delta) = \sum_{r \in \mathcal{A}^*} \frac{|y_{GT} = r \wedge y_{PD} = c \wedge g(x) > \delta|}{|y_{GT} = r|} \quad (24)$$

$$\text{wFP}(c, \delta) = \sum_{r \in \bar{\mathcal{A}}^*} \frac{|y_{GT} = r \wedge y_{PD} = c \wedge g(x) > \delta|}{|y_{GT} = r|} \quad (25)$$

Cross correlating AUROC with wPre and $\delta = 0.9$, we evaluate the performance for Task 1, Task 2 and Task 2 vs Task 3 in Fig. 16(d-f), respectively. We obtain the following findings. For Task 1, *car*, *plants* and *bike* have poor precisions, and *bike* has the worst performance in discriminating between ID and OOD data due to the interclass ambiguity. For Task 2, *plants* and *fence* have poor precisions, whereas *sign* has difficulty in discriminating between correct or wrong predictions of the semantic classes. In the case that OOD data exist in Task 3, *car*, *plants* and *bike* are the most affected.

D. Feature Space Analysis

The validity of trust scores, namely, the failure detection and OOD detection abilities, are also relevant to the feature description of the models. We have found that *bike*, *plants*, and *fence* are hard to distinguish from OOD classes in the experimental results. Therefore, we visualize the last layer features using T-SNE to compare the feature distribution of OOD classes and ID classes, as shown in Fig. 17.

The OOD detection performance is affected by the feature confusion between the predictive class and the OOD data. Similar to Experiment 1, the feature distribution of classes

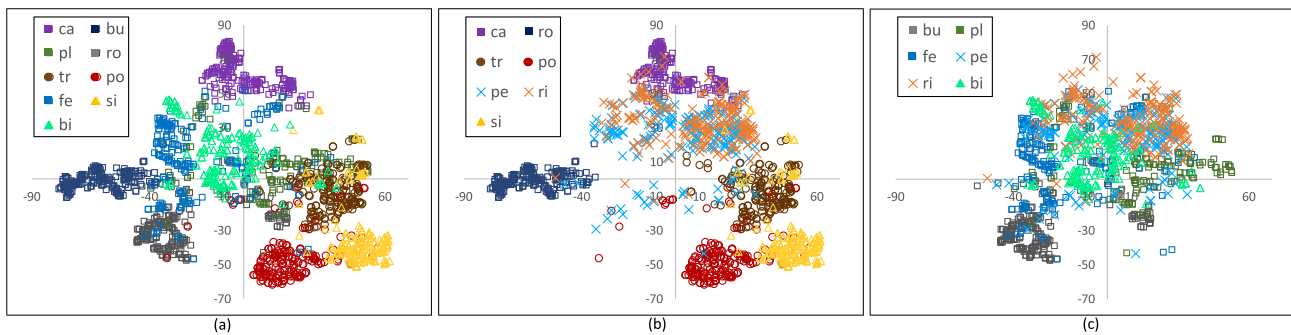


Fig. 17. T-SNE plot of 200 random sampled points from AugKITTI using Cylinder3D. (a) Feature points of ID classes. (b) Feature points of *car*, *sign*, *trunk*, *pole*, *road* and OOD classes. (c) Feature points of *plants*, *fence*, *building*, *bike* and OOD classes. The abbreviations of class names are defined in Table II.

also determines the OOD detection performance. As we define *people* and *rider* as OOD classes, ID classes such as *plants* and *bike* have feature confusion with OOD classes to a degree, making it difficult to distinguish the OOD data. This result can explain the ID/OOD AUROC differences in our experiment.

We also find that the proposed TSD analysis can appropriately evaluate the feature confusion, failure and OOD detection performances. As illustrated in Fig. 15, *bike*, *plants*, *fence* and *building* demonstrate examples of the worst performance, which corresponds to Fig. 17(c), where features of these classes are easily confused with OOD classes. In contrast, features of *road*, *sign*, *pole* are far away from the OOD classes, whereas *car* is slightly confused with the OOD classes, which also corresponds to the TSD analysis.

In addition, trust scores are not reliable for classes whose features are confused with other classes. Using wPre with $\delta=0.9$, as shown in Fig. 16(d-f), we find that *plants* have poor precisions in all three tasks, which means that the reliability of *plants* prediction is limited even though it gives a high confidence value. The features of *plants* are severely confused with other classes in Fig. 17(a)(c), which explains the results. In addition, if features are confused with OOD classes, they will be influenced by OOD data more easily. As our analysis shows in Fig. 16(f), *car*, *plants* and *bike* are most affected by OOD, while their features are also confused with OOD classes to different degrees.

From the feature space analysis, we can give a systematic summary. The validity of trust scores is also greatly affected by feature confusion among classes, which makes failure detection and OOD detection more challenging.

VI. DISCUSSION

A. The Class Imbalance Problem

Class imbalance is a general problem in the deep learning domain. Real-world scenes are occupied with various objects in different proportions. Although *people*, *rider*, *bike*, *sign*, etc. occupy small proportions in the scenes, and subsequently with 3D LiDAR data, precise perception of these objects is essential for an autonomous agent to traverse safely and smoothly in populated environments. However, due to the small number of data samples of these categories, their models could be under-trained, and their classification failure could be underestimated. Although many methods [10], [33] have

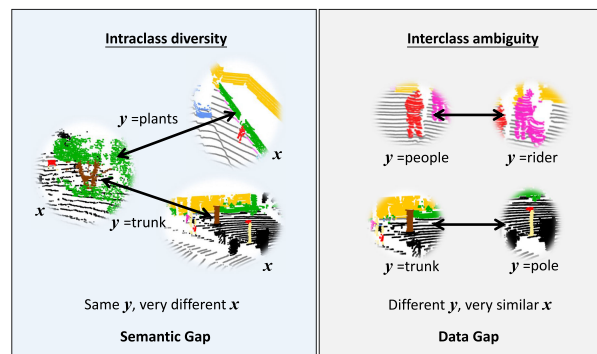


Fig. 18. Examples of intraclass diversity and interclass ambiguity.

been developed, the class imbalance problem is far from being solved in 3DSS tasks. Improving model performance on small classes needs more attention, and semi-supervised learning [53], few-shot learning [54], and zero-shot learning [55] for 3DSS could be the future research directions.

There is more than one aspect of class imbalance. Not only data size but also basic properties of each semantic class will affect the model performance, which determines the difficulty of learning. Notice that the state-of-the-art model is less affected by data size in our experiments. Specifically, this difference is mainly composed of the following reasons. First, the enhanced feature extraction ability brought by the model structure. Second, the targeted processing of small classes by the loss function, such as focal loss [39], lovasz loss [56], etc.. However, model performance of some classes is hard to improve such as *plants* and *fence* due to their complicated characteristics. These hard classes bring a huge challenge for the 3DSS models, and we will discuss it further in the next subsection.

B. Intraclass Diversity and Interclass Ambiguity

Experimental results show that even for some classes having a large number of training samples, their model performance is unpleasant, such as *fence* and *plants*. Through feature space analysis, it is found that intraclass diversity and interclass ambiguity are among the main reasons, which reveals that there are large semantic and data gaps in the data.

As illustrated in Fig. 18, although some objects belong to the same semantic class, they may have various shapes, e.g.,

trunk with straight and forked branches, *plants* with nature and trimmed crowns. In these cases, the semantic category y is the same, whereas the sensor data x are very different, which is called the semantic gap. Conversely, some objects are of different semantic categories, whereas due to partial observation, occlusion and/or low-resolution point sampling, their data could present similar features, such as *people* and *rider*, *pole* and straight *trunk* with its crown occluded. In these cases, the semantic category y are different, whereas sensor data x are very similar, which is called the data gap.

Semantic gap and data gap make some classes difficult to learn. However, the learning ability of model and the category definition are a very coupled relationship. The ability of the model is the key to whether it can distinguish between classes, and whether the categories themselves are easy to distinguish will also affect the model performance. Therefore, on one hand, improving model learning ability on imperfect dataset such as robust deep learning [57], meta learning [58] can alleviate the problem. On the other hand, more reasonable category definition could help to alleviate the intra-class diversity and inter-class ambiguity problem. For example, the lack of detailed class definitions may be an important cause of intra-class diversity and lead to inter-class ambiguity. Recently, some works [59], [60] have paid attention to refine the labeling of ImageNet datasets. These works show that reasonable category definitions can help model learning. However, this problem has not yet been explored in 3DSS tasks, and unsupervised category discovery [61] may be a potential solution that needs to be further investigated in the future.

C. Can 3DSS Results Be Trusted

For safety-critical applications such as autonomous driving, when compared to improving the overall statistical accuracy of 3DSS and/or OOD detection, it is more important to understand whether the current prediction results are trustworthy. Many metrics have been developed for such a purpose, where Softmax confidence, uncertainty, ODIN and Mahalanobis distance are among the most popular metrics, as discussed in earlier sections. These metrics use a 3DSS model's output at the intermediate or final layers and estimates a trust score on whether the prediction of semantic class is correct or whether it is an ID/OOD data. Many methods have been developed to find or calibrate metrics, such that a high score reflects that the model is confident of the result, whereas a low score shows that the model is unsure and thus could be either a wrong prediction on the semantic class or an unseen object in the training data (OOD).

However, the performance of such trust scoring methods depends on both the metrics and the 3DSS model. Faced with the dual challenges of class imbalance and OOD data, the performance of a 3DSS model on each class could be much different, including both well-learned and poorly-learned classes. For the poorly-learned classes, high trust scores could also be given on those wrong classified ID categories and OOD, making it hard to capture a correct understanding of the trustworthiness of the 3DSS results. A meticulous design of the

metrics that addresses the 3DSS model's various performances on each semantic class is required, which may lead to a more pinpointed understanding of whether 3DSS results can be trusted. More studies are needed in the future.

VII. CONCLUSION AND FUTURE WORK

This work aims to explore the relationship among class imbalance, 3DSS model performance, failure and the OOD detection ability for 3D semantic segmentation tasks and to analyse the underlying reasons for these performances. For these purposes, we conduct two experiments and conduct both a confusion and feature analysis of each class. For experiments and analysis, we introduce a data augmentation method for the 3D LiDAR dataset, create the AugKITTI dataset based on SemanticKITTI and SemanticPOSS, and propose the wPre metric and a TSD for confusion analysis. The major findings from the above experimental studies are as follows:

- 1) The classes are imbalanced not only in terms of their data size but also in terms of the basic properties of each semantic category, in which confusion among different semantic categories could be another key factor that greatly affects the model performance.
- 2) The intraclass diversity and interclass ambiguity make class learning difficult and greatly limits model performance, causing the semantic and data gap challenge.
- 3) Trust scores are unreliable for classes whose features are confused with other classes. For these classes, high trust scores could also be given on those wrongly classified ID classes and OOD, making the 3DSS predictions unreliable and creating a challenge of judging the 3DSS results to be trustful.

For real-world 3D semantic segmentation with complex scenes, deep models are required to address the class imbalance problem, have an awareness when the model is uncertain and detect unseen objects. This work examines the performance of representative 3DSS models, and the results show that there are still challenges for real-world applications using deep learning models. From the experimental results, three directions can be explored for improving OOD detection, improving 3DSS model performance, designing thresholding scores, and alleviating the intraclass diversity and interclass ambiguity in training data. However, this study only considers a small portion of the 3DSS methods and trust scoring methods with manually chosen OOD classes. In the future, we would like to consider more types of failure and OOD detection methods and try to develop a practical failure and OOD detector for real-world 3D semantic segmentation task.

REFERENCES

- [1] H. Yu et al., "Methods and datasets on semantic segmentation: A review," *Neurocomputing*, vol. 304, pp. 82–103, Aug. 2018.
- [2] F. Lateef and Y. Ruichek, "Survey on semantic segmentation using deep learning techniques," *Neurocomputing*, vol. 338, pp. 321–348, Apr. 2019.
- [3] Y. Xie, J. Tian, and X. X. Zhu, "Linking points with labels in 3D: A review of point cloud semantic segmentation," 2019, *arXiv:1908.08854*.
- [4] B. Patz, Y. Papelis, R. Pillat, G. Stein, and D. Harper, "A practical approach to robotic design for the DARPA urban challenge," *J. Field Robot.*, vol. 25, no. 8, pp. 528–566, 2008.

- [5] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3D LiDAR using fully convolutional network," 2016, *arXiv:1608.07916*.
- [6] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1907–1915.
- [7] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A survey," 2019, *arXiv:1912.12033*.
- [8] G. Marcus, "Deep learning: A critical appraisal," 2018, *arXiv:1801.00631*.
- [9] B. Gao, Y. Pan, C. Li, S. Geng, and H. Zhao, "Are we hungry for 3D LiDAR data for semantic segmentation? A survey of datasets and methods," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6063–6081, Jul. 2022.
- [10] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Netw.*, vol. 106, pp. 249–259, Oct. 2017.
- [11] G. Haixiang et al., "Learning from class-imbalanced data: Review of methods and applications," *Expert Syst. Appl.*, vol. 73, pp. 220–239, May 2017.
- [12] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," 2016, *arXiv:1610.02136*.
- [13] P. Koopman and M. Wagner, "Autonomous vehicle safety: An interdisciplinary challenge," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 1, pp. 90–96, Jan. 2017.
- [14] M. Salehi, H. Mirzaei, D. Hendrycks, Y. Li, M. Hossein Rohban, and M. Sabokrou, "A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: Solutions and future challenges," 2021, *arXiv:2110.14051*.
- [15] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 427–436.
- [16] J. Behley et al., "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9297–9307.
- [17] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.
- [18] H. Zhou et al., "Cylinder3D: An effective 3D framework for driving-scene LiDAR semantic segmentation," 2020, *arXiv:2008.01550*.
- [19] Q. Hu et al., "RandLA-Net: Efficient semantic segmentation of large-scale point clouds," 2019, *arXiv:1911.11236*.
- [20] C. Xu et al., "SqueezeSegV3: Spatially-adaptive convolution for efficient point-cloud segmentation," 2020, *arXiv:2004.01803*.
- [21] Y. Pan, B. Gao, J. Mei, S. Geng, C. Li, and H. Zhao, "SemanticPOSS: A point cloud dataset with large quantity of dynamic instances," 2020, *arXiv:2002.09147*.
- [22] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" 2017, *arXiv:1703.04977*.
- [23] A. Malinin, "Uncertainty estimation in deep learning with application to spoken language assessment," Ph.D. dissertation, Dept. Eng., Univ. Cambridge, Cambridge, U.K., 2019.
- [24] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," 2017, *arXiv:1706.02690*.
- [25] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 7167–7177.
- [26] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.
- [27] H. Thomas, C. R. Qi, J.-E. Deschaut, B. Marcotegui, F. Goulette, and L. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6411–6420.
- [28] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep convolutional networks on 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9621–9630.
- [29] J. Huang and S. You, "Point cloud labeling using 3D convolutional neural network," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 2670–2675.
- [30] D. Rethage, J. Wald, J. Sturm, N. Navab, and F. Tombari, "Fully-convolutional point networks for large-scale point clouds," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 596–611.
- [31] Y. Wang, T. Shi, P. Yun, L. Tai, and M. Liu, "PointSeg: Real-time semantic segmentation based on 3D LiDAR point cloud," 2018, *arXiv:1807.06288*.
- [32] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "RangeNet++: Fast and accurate LiDAR semantic segmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 4213–4220.
- [33] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J. Big Data*, vol. 6, no. 1, pp. 1–54, 2019.
- [34] Y. Zhang, B. Kang, B. Hooi, S. Yan, and J. Feng, "Deep long-tailed learning: A survey," 2021, *arXiv:2110.04596*.
- [35] P. Hensman and D. Masko, "The impact of imbalanced training data for convolutional neural networks," Dept. Comput. Sci., KTH Roy. Inst. Technol., Stockholm, Sweden, 2015.
- [36] K. Ghosh, C. Bellinger, R. Corizzo, B. Krawczyk, and N. Japkowicz, "On the combined effect of class imbalance and concept complexity in deep learning," 2021, *arXiv:2107.14194*.
- [37] G. Menardi and N. Torelli, "Training and assessing classification rules with imbalanced data," *Data Mining Knowl. Discovery*, vol. 28, no. 1, pp. 92–122, Jan. 2014.
- [38] S. Pouyanfar et al., "Dynamic sampling in convolutional neural networks for imbalanced data classification," in *Proc. IEEE Conf. Multimedia Inf. Process. Retr. (MIPR)*, Apr. 2018, pp. 112–117.
- [39] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
- [40] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9268–9277.
- [41] X. Chen, C. Zhang, G. Lin, and J. Han, "Compositional prototype network with multi-view comparison for few-shot point cloud semantic segmentation," 2020, *arXiv:2012.14255*.
- [42] V. Schwag et al., "Analyzing the robustness of open-world machine learning," in *Proc. 12th ACM Workshop Artif. Intell. Secur.*, Nov. 2019, pp. 105–116.
- [43] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," 2018, *arXiv:1812.04606*.
- [44] A.-A. Papadopoulos, M. R. Rajati, N. Shaikh, and J. Wang, "Outlier exposure with confidence control for out-of-distribution detection," *Neurocomputing*, vol. 441, pp. 138–150, Jun. 2021.
- [45] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun, "Learning discriminative reconstructions for unsupervised outlier removal," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1511–1519.
- [46] L. Ruff et al., "Deep one-class classification," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4393–4402.
- [47] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," 2016, *arXiv:1612.01474*.
- [48] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.
- [49] H. Jiang, B. Kim, M. Y. Guan, and M. Gupta, "To trust or not to trust a classifier," 2018, *arXiv:1805.11783*.
- [50] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1321–1330.
- [51] H. Caesar et al., "NuScenes: A multimodal dataset for autonomous driving," 2019, *arXiv:1903.11027*.
- [52] Y. Ovadia et al., "Can you trust your Model's uncertainty? Evaluating predictive uncertainty under dataset shift," 2019, *arXiv:1906.02530*.
- [53] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Mach. Learn.*, vol. 109, no. 2, pp. 373–440, Feb. 2020.
- [54] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–34, 2020.
- [55] W. Wang, V. W. Zheng, H. Yu, and C. Miao, "A survey of zero-shot learning: Settings, methods, and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–37, 2019.
- [56] M. Berman, A. R. Triki, and M. B. Blaschko, "The Lovasz-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4413–4421.

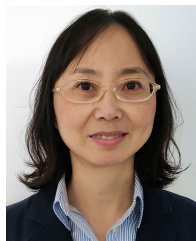
- [57] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 7, 2022, doi: [10.1109/TNNLS.2022.3152527](https://doi.org/10.1109/TNNLS.2022.3152527).
- [58] T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey, "Meta-learning in neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5149–5169, Sep. 2022.
- [59] L. Beyer, O. J. Hénaff, A. Kolesnikov, X. Zhai, and A. van den Oord, "Are we done with ImageNet?" 2020, *arXiv:2006.07159*.
- [60] S. Yun, S. J. Oh, B. Heo, D. Han, J. Choe, and S. Chun, "Re-labeling ImageNet: From single to multi-labels, from global to localized labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2340–2350.
- [61] K. Han, A. Vedaldi, and A. Zisserman, "Learning to discover novel visual categories via deep transfer clustering," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8401–8409.



Yancheng Pan (Member, IEEE) received the B.S. degree in artificial intelligence (artificial intelligence and technology) from Peking University, Beijing, China, in 2020, where he is currently pursuing the M.S. degree with the Key Laboratory of Machine Perception (MOE). His research interests include intelligent vehicles, computer vision, 3D LiDAR perception, and uncertainty estimation.



Fan Xie (Member, IEEE) is currently pursuing the B.S. degree in artificial intelligence (artificial intelligence and technology) from Peking University, Beijing, China. His research interests include computer vision, deep learning, 3D LiDAR perception, and uncertainty estimation.



Huijing Zhao (Senior Member, IEEE) received the B.S. degree in computer science from Peking University, Beijing, China, in 1991, and the M.E. and Ph.D. degrees in civil engineering from The University of Tokyo, Japan, in 1996 and 1999, respectively. From 1999 to 2007, she was a Post-Doctoral Researcher and a Visiting Associate Professor at the Center for Space Information Science, The University of Tokyo. In 2007, she joined Peking University as a tenure-track Professor with the School of Electronics Engineering and Computer Science and became an Associate Professor with tenure in 2013 and promoted to a Full Professor in 2020. Her research interests include intelligent vehicle and mobile robot, such as machine perception, behavior learning, and motion planning. She has special interests on the studies through real-world data collection.