This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Open-set Semantic Segmentation for Point Clouds via Adversarial Prototype Framework

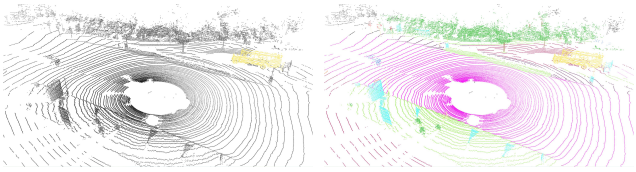
Jianan Li[1,2], Qiulei Dong[*,1,2,3]
[1]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[2]State Key Laboratory of Multimodal Artificial Intelligence Systems,
Institute of Automation, Chinese Academy of Sciences, Beijing, China
[3]Center for Excellence in Brain Science and Intelligence Technology,
Chinese Academy of Sciences, Beijing, China
lijianan211@mails.ucas.ac.cn, qldong@nlpr.ac.cn

Abstract

Recently, point cloud semantic segmentation has attracted much attention in computer vision. Most of the existing works in literature assume that the training and testing point clouds have the same object classes, but they are generally invalid in many real-world scenarios for identifying the 3D objects whose classes are not seen in the training set. To address this problem, we propose an **Adversarial Prototype Framework (APF)** *for handling the open-set 3D semantic segmentation task, which aims to identify 3D unseen-class points while maintaining the segmentation performance on seen-class points. The proposed APF consists of a feature extraction module for extracting point features, a prototypical constraint module, and a feature adversarial module. The prototypical constraint module is designed to learn prototypes for each seen class from point features. The feature adversarial module utilizes generative adversarial networks to estimate the distribution of unseen-class features implicitly, and the synthetic unseen-class features are utilized to prompt the model to learn more effective point features and prototypes for discriminating unseen-class samples from the seen-class ones. Experimental results on two public datasets demonstrate that the proposed APF outperforms the comparative methods by a large margin in most cases.*

1. Introduction

Point cloud semantic segmentation is an important and challenging topic in computer vision. Most of the existing works [9–11, 29] in literature are based on the assumption that both the training and testing point clouds have the same object classes, however, this assumption is no more valid in many real-world scenarios, due to the fact that the classes of some observed 3D points may not be presented in the training set. Hence, the following problem on open-set 3D semantic segmentation is naturally raised: How does a segmentation model simultaneously identify unseen-class 3D points and maintain the segmentation accuracy of seen-class 3D points in open-set scenarios?



(a) AD (b) O3D

Figure 1. Visualization of the goals of anmaly detection (AD) and open-set 3D semantic segmentation (O3D) on SemanticKITTI [2]. AD is to identify the unseen-class data, while O3D is to simultaneously identify the unseen-class data and segment seen-class data. The unseen-class points are colorized in yellow.

Compared with anomaly detection [3, 23, 26], open-set 3D semantic segmentation (O3D) is more challenging, for it also needs to assign labels to seen-class data simultaneously, as shown in Figure 1. In fact, some existing techniques [6, 15, 17, 18] for open-set 2D semantic segmentation (O2D) task could be extended to handle the O3D task, however, their open-set ability is generally limited in 3D scenarios. In addition, to our best knowledge, only one pioneering work [7] has investigated a special technique for O3D task. In [7], an O3D method called REAL is proposed to utilize normal classifiers to segment seen-class points and regard the randomly resized objects as unseen-class objects which are detected by the redundancy classifiers. REAL outperforms some extended O2D methods in the O3D task, how-

*Corresponding author

ever, the AUPR (Area Under the Precision-Recall curve) is lower than 21% on two public datasets as shown in Table 1 and Table 2 in Section 4, mainly because the resizing process in REAL alters the geometric structure of the initial point clouds to some extent. These results indicate that there still exists a huge space for improvement on O3D task.

Addressing the above issue, we propose an Adversarial Prototype Framework (APF) for open-set 3D segmentation, which segments point clouds from a discriminative perspective and estimates the distribution of unseen-class features from a generative perspective. The proposed APF consists of three modules: a feature extraction module, a prototypical constraint module, and a feature adversarial module. The feature extraction module is employed to extract latent features from the input point clouds, which could be an arbitrary closed-set point cloud segmentation network in principle. Given the point features, the prototypical constraint module is explored from the discriminative perspective to learn a prototype for each seen class. The feature adversarial module is explored from the generative perspective, which employs the generative adversarial networks (GAN) to synthesize point features to estimate the unseen-class feature distribution, based on the finding stated in [6] that the unseen-class features usually aggregate in the center of the feature space. And the synthesized unseen-class features in this module could further prompt the model to learn more discriminative point features and prototypes. After the whole APF is trained, a point-to-prototype hybrid distance-based criterion is introduced for open-set 3D segmentation.

In sum, the contributions of this paper are as follows:

- We propose the adversarial prototype framework (APF) for handling the open-set 3D semantic segmentation task. Under the proposed APF, various open-set 3D segmentation methods could be straightforwardly derived by utilizing existing closed-set 3D segmentation networks as the feature extraction module. The effectiveness of the proposed APF has been demonstrated by the experimental results in Section 4.
- Under the proposed framework, we explore the prototypical constraint module, which learns the corresponding prototype for each seen class. The learned prototypes are not only conducive to segmenting seen-class points, but also to detecting unseen-class points.
- Under the proposed framework, we explore the feature adversarial module to synthesize unseen-class features. The synthetic features are helpful for improving the discriminability of both the seen-class features and prototypes via the adversarial mechanism.

2. Related Work

2.1. 3D semantic segmentation

In recent years, numerous works have been proposed for segmenting 3D point clouds, which could be roughly divided into three categories: projection-based methods, voxel-based methods, and point-based methods.

Projection-based methods [32, 36, 37] project 3D point clouds into multi-view or spherical images, and then leverage the 2D CNNs to extract features, which are aggregated to output point clouds features. However, the 3D topology and geometric relations are inevitably altered or ignored.

Voxel-based methods [16, 31] voxelize the point clouds into a series of dense girds and utilize 3D convolution to extract point clouds features directly. Zhu *et al*. [41] designed a cylindrical partition strategy and asymmetrical 3D convolution networks to explore the 3D geometric pattern while maintaining the inherent properties of the outdoor point clouds. Volumetric representation has shown its capability of processing the point clouds, especially for the sparse large-scale outdoor point clouds.

Point-based methods directly take the raw point clouds as input. The pioneering work PointNet [27] is proposed to extract per-point features via the shared MLP, alignment network and symmetric aggregation strategy. Inspired by PointNet, numerous sophisticated point-based methods [13,14,19,28,33,35] have been proposed to capture the local geometric patterns and contextual information among the points. In recent years, with transformer and self-attention models [5, 12, 34] revolutionizing the deep learning community, several works have tentatively applied transformer to conduct semantic segmentation for point clouds. Zhao *et al*. [40] constructed high-performance Point Transformer network for point clouds processing, which could serve as a general backbone for point clouds understanding tasks. Lai *et al*. [22] proposed Stratified Transformer to capture long-range contexts for point clouds segmentation.

It is noted that the aforementioned methods are only available in closed-set scenarios, and compared with these closed-set works, the investigation of open-set 3D semantic segmentation (O3D) is still in its infancy. As discussed in Section 1, only Cen *et al*. [7] have paid special attention on open-set 3D segmentation, however, its performance on open-set 3D segmentation is still limited.

2.2. Open-set 2D semantic segmentation

Open-set 2D semantic segmentation (O2D) has drawn increasing attention in computer vision recently. Existing works for O2D could be roughly divided into two categories: discriminative methods and generative methods.

Discriminative methods [15, 17, 18] usually estimate the uncertainty or calibrate the probability distribution to separate unseen-class pixels from seen-class pixels. Inspired by the exemplar theory, Hwang *et al*. [20] proposed an novel exemplar-based network to detect novel classes by clustering. Analogously, based on contrastive clustering, Cen *et al*. [6] proposed to identify anomalous pixels by calculating the embedded feature similarity in the metric space. How-

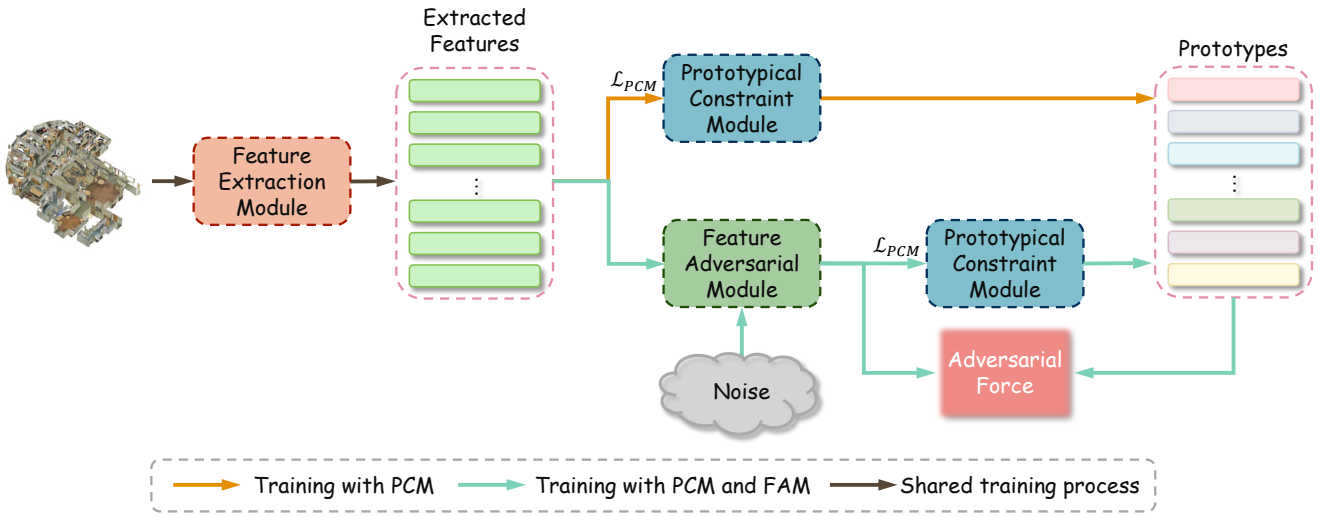


Figure 2. Architecture of the adversarial prototype framework. Firstly, the feature extraction module and prototypical constraint module are trained jointly in a prototype-based clustering manner under the guidance of $\mathcal{L}_{PCM}$. Then the feature extraction module is fixed, the feature adversarial module and prototypical constraint module are trained jointly. The parametric prototypes are updated by $\mathcal{L}_{PCM}$ and the adversarial force which is formulated in Equation (10).

ever, these methods usually suffer from the vulnerability to confusing samples and thus resulting in too many false-positive detections, as stated in [4].

Generative methods [24, 38] usually utilize autoencoder or GAN to reconstruct images from the segmentation maps. The unseen-class pixels are detected in the light of the differences between the reconstructed images and the original input. The success of these generative models is attributed to the reliable high-resolution generation results. Deviating from previous generative approaches, Kong *et al*. [21] proposed to synthesize features to augment the training data and utilize the discriminator as the unseen-class detector after a delicate selection strategy. However, as stated in [39], the generative models usually yield lower closed-set segmentation accuracy than the discriminative ones, which limits their applications.

As indicated in Section 1, when some O2D methods are utilized jointly with the existing closed-set 3D segmentation networks, they could indeed handle the O3D task. However, their performances are lower in comparison to the methods (e.g., REAL [7] and the proposed method in this work) that are specially designed for handling the O3S task as demonstrated by the results in [7] and Section 4 of this paper.

3. Methodology

In this section, we propose the Adversarial Prototype Framework (APF) for open-set 3D semantic segmentation (O3D). Firstly, we describe the architecture of APF. Then, we elaborate the prototypical constraint module and feature adversarial module respectively. Finally, the training and inference procedure is presented.

3.1. Architecture

The architecture of the proposed APF is illustrated in Figure 2, and it contains three components: a feature extraction module, a prototypical constraint module, and a feature adversarial module.

The feature extraction module is to extract features from point clouds, which could be an arbitrary closed-set 3D segmentation network. The prototypical constraint module is explored from a discriminative perspective to learn a prototype for each seen class in a clustering fashion. The feature adversarial module is explored from a generative perspective to synthesize point features to estimate the underlying unseen-class feature distribution.

3.2. Prototypical constraint module

Given the extracted point features, the prototypical constraint module (PCM) is designed to learn the prototypes for seen classes. And Figure 3 illustrates the architecture of PCM.

We expect to acquire discriminative prototypes for each seen class, and segment the points according to the distances between point features and prototypes in the feature space, mimicking the traditional clustering algorithm.

Specifically, in addition to the Euclidean distance, we incorporate the cosine of the angle into distance setting, as done in [8]. The hybrid distance between a point feature $f_i \in \mathbb{R}^d$ and a prototype $P_j \in \mathbb{R}^d$, (j = 1, $\cdots$, C, where C denotes the number of seen classes) is formulated as fol-

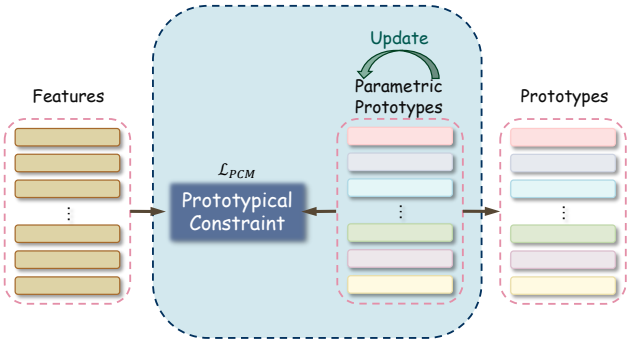


Figure 3. Architecture of the prototypical constraint module. The prototypes are learnable parameters which are updated under the guidance of the prototypical constraint $\mathcal{L}_{PCM}$.

lows:

$$d(f_i, P_c) = \|f_i - P_c\|_2^2 - f_i \cdot P_c, \tag{1}$$

where $\|\cdot\|_2$ denotes the L2-norm.

We randomly initialize the parametric prototype set $\mathbf{P} = \{P_j\}_{j=1}^{C}$. And the probability of a point feature f_i belonging to its corresponding category c is defined as:

$$p(y_i = c|f_i, \mathbf{P}) = \frac{\exp(-d(f_i, P_c))}{\sum_{j=1}^{C} \exp(-d(f_i, P_j))}, \tag{2}$$

where y_i is the predicting label of f_i.

To ensure the seen-class features to be closer to their corresponding prototypes and far away from other prototypes, we utilize a distance-based cross entropy loss:

$$\mathcal{L}_{dce} = -\frac{1}{N_c} \sum_{i=1}^{N_c} \log p(y_i = c|f_i, \mathbf{P}), \tag{3}$$

where N_c represents the number of the seen-class points.

To further tighten the features of the same category, we formulate an attractive loss term, putting more emphasis on the attractive force between the feature f_i and its corresponding prototype P_c:

$$\mathcal{L}_{attr} = \frac{1}{N_c} \sum_{i=1}^{N_c} \|f_i - P_c\|_2^2. \tag{4}$$

As noted in Eq. (3) and Eq. (4), there exists a degenerate all-zero solution to the loss terms $\mathcal{L}_{dce}$ and $\mathcal{L}_{attr}$ in theory. Considering that the numerical interval in each feature dimension has been normalized into [0, 1], we design the following regularization term by enforcing each dimension to be close to 1, in order to avoid the degenerate solution and compulsively force the model to learn more informative features. The regularization term is formulated as:

$$\mathcal{L}_{info} = \frac{1}{Cd} \sum_{i=1}^{C} \sum_{j=1}^{d} |\mathbf{P}_{ij} - 1|. \tag{5}$$

In summary, the total loss function of prototypical constraint is a weighted sum of the above three terms:

$$\mathcal{L}_{PCM} = \mathcal{L}_{dce} + \lambda_{attr}\mathcal{L}_{attr} + \lambda_{info}\mathcal{L}_{info}, \tag{6}$$

where λ_{attr} and λ_{info} are hyper-parameters.

Theoretically, the optimization of the model will decrease $d(f_i, P_c)$, which encourages the features to be closer to their corresponding prototypes and makes the distribution of features of the same category more compact.

3.3. Feature adversarial module

To take the unseen-class factors into consideration, the feature adversarial module (FAM) is designed to estimate the underling characteristics of unseen-class features. The architecture of FAM is illustrated in Figure 4, and it contains three parts: a generator G, a discriminator D, and an adversarial mapper M.

Similar to the traditional GAN, the generator takes the gaussian noises $n = \{n_i\}_{i=1}^{N_c}$ as input to synthesize high-fidelity point features, and the discriminator maps the input features to $[0, 1]$, which represents the probability of the input belonging to real features. The discriminator is trained to correctly distinguish the real extracted features and synthetic features:

$$\max_{D} \frac{1}{N_c} \sum_{i=1}^{N_c} \big[\log D(f_i) + \log(1 - D(G(n_i)))\big]. \tag{7}$$

And the synthetic features are expected to deceive the discriminator:

$$\max_{G} \frac{1}{N_c} \sum_{i=1}^{N_c} \log(D(G(n_i))). \tag{8}$$

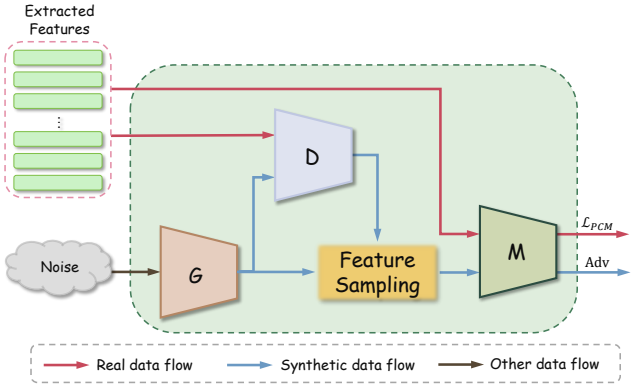


Figure 4. Architecture of the feature adversarial module. G, D, and M denote the generator, discriminator, and adversarial mapper respectively. The real features output by M are employed to calculate $\mathcal{L}_{PCM}$, and the synthetic features output by M are involved in the optimization of Adv, which is formulated in Equation (10).

Deviating from the existing works [8,21] that incorporate all the synthetic samples into training, we employ a feature sampling (FS) strategy to prevent some synthetic features that are excessively similar to the seen-class features from misleading the closed-set segmentation. Specifically, we select the features f^s whose corresponding confidence output by D is under a predetermined threshold λ_s:

$$f^s = \{G(n_i) | D(G(n_i)) < \lambda_s\}. \tag{9}$$

Notably, we freeze the feature extractor for the stability of GAN when FAM is added into training, which leaves the extracted features fixed. Thus we utilize the adversarial mapper to map the features to a novel feature space for calibrating the feature distribution.

The region where the unseen-class features aggregate is defined as the **open space** in [30]. And it is stated in [6] that **open space** is found to be located in the center of the feature space. Namely, the distances between the unseen-class features and all prototypes should be equally small. Thus, we utilize a delicate objective function to put an additional adversarial force on the synthetic features:

$$\mathrm{Adv}(f^s, \mathbf{P}) = \frac{1}{N_s} \sum_{i=1}^{N_s} \big[- \frac{1}{C} \sum_{j=1}^{C} [h(f_i^s, P_j) \cdot \log(h(f_i^s, P_j) + \epsilon)] \big], \tag{10}$$

where $h(f_i^s, P_j) = \mathrm{Softmax}(\|M(f_i^s) - P_j\|_2^2)$, ϵ is a predetermined small number to avert NaN issue, and N_s is the number of the selected features.

The design of Equation (10) is referenced from the information entropy in information theory [25]. Theoretically, the maximization of Equation (10) forces the synthetic features to be closer to the **open space**.

Combining Equation (8) and Equation (10), the generator G is optimized by the following formula:

$$\max_G \big[\frac{1}{N_c} \sum_{i=1}^{N_c} \log(D(G(n_i))) \big] + \lambda_{adv} \cdot \mathrm{Adv}(f^s, \mathbf{P}), \tag{11}$$

where λ_{adv} is a hyper-parameter.

To maintain the closed-set segmentation ability of the model while boosting its open-set segmentation ability, the adversarial mapper M is optimized by:

$$\min_M \big[\frac{1}{N_c} \sum_{i=1}^{N_c} \mathcal{L}_{PCM} \big] - \lambda_{adv} \cdot \mathrm{Adv}(f^s, \mathbf{P}). \tag{12}$$

Notably, it has to be pointed out that $\mathrm{Adv}(f^s, \mathbf{P})$ also has an impact on $\mathbf{P}$, for the prototypes are learnable parameters in this work.

As shown in Figure 5, the seen-class features are only influenced by the prototype force, while the synthetic features tend to be attracted to a certain prototype because they are encouraged to be analogous to some seen-class features

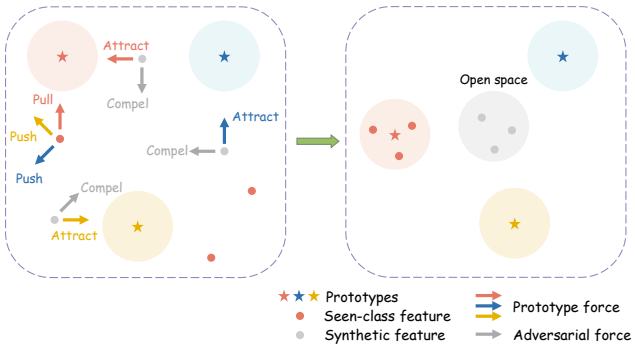


Figure 5. Illustration of the feature calibration procedure. The seen-class features are encouraged to be pulled closer to their corresponding prototypes and pushed away from the other prototypes. The synthetic features tend to be attracted by a certain prototype, while the adversarial force compels them to strike a balance between all prototypes.

to deceive the discriminator. In the meantime, the synthetic features are forced to strike a balance between all prototypes by Equation (10). A more distinguishable feature distribution is eventually formed through the adversarial mechanism between the prototypes and synthetic features.

3.4. Training and inference

At the training stage, we first train the feature extraction module (FEM) and prototypical constraint module (PCM) jointly to acquire coarse point features, which enjoy a promising closed-set discriminability. Note that when FEM and PCM are trained jointly, the features employed to calculate $\mathcal{L}_{PCM}$ are the extracted features $\mathbf{F} = \{f_i\}_{i=1}^{N_c}$ output by the feature extractor.

Then the feature extractor in FEM is fixed, and the feature adversarial module (FAM) is added into training to estimate the potential distribution of the unseen-class features. The FAM and PCM are trained jointly to further refine the coarse point features and acquire more discriminative prototypes. Note that when FAM and PCM are trained jointly, the features employed to calculate $\mathcal{L}_{PCM}$ are the refined features $\hat{\mathbf{F}} = M(\mathbf{F}) = \{\hat{f}_i\}_{i=1}^{N_c}$ output by the adversarial mapper.

At the inference stage, only the feature extractor and adversarial mapper are used, as shown in Figure 6. Concretely, the FEM is implemented to extract corresponding features $\mathbf{F}$ for the testing points, then the adversarial mapper outputs the refined features $\hat{\mathbf{F}}$ to conduct open-set 3D semantic segmentation.

According to the revealed finding in [6] that unseen-class features usually aggregate in the center of the feature space, if a 3D point $\mathbf{x}_i$ does not belong to any seen class, the sum of the distances between $\mathbf{x}_i$ and all the prototypes are expected to be relatively small. Hence, the probability $p_u(\cdot)$ that $\mathbf{x}_i$

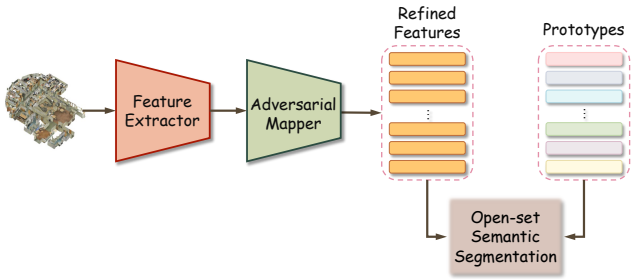


Figure 6. Architecture of APF in the inference phase. The point clouds are simply fed into the feature extractor and adversarial mapper. The refined features and prototypes are employed to conduct open-set semantic segmentation.

is identified as $unseen$ is defined as:

$$p_u(\mathbf{x}_i) = 1 - \frac{\sum_{j=1}^{C} \|\hat{f}_i - P_j\|_2^2}{\max_k \sum_{j=1}^{C} \|\hat{f}_k - P_j\|_2^2}. \quad (13)$$

Then, each testing point will be classified as one of the seen classes or identified as $unseen$ via a point-to-prototype hybrid distance-based criterion:

$$y_i = \begin{cases} unseen, & p_u(\mathbf{x}_i) \geq \lambda_u \\ \arg\max_j p(y_i = j | f_i, \mathbf{P}), & p_u(\mathbf{x}_i) < \lambda_u \end{cases}, \quad (14)$$

where λ_u is the threshold and $p(\cdot)$ is defined in Equation (2).

4. Experiments

4.1. Datasets

We conduct experiments on the following two public datasets to verify the effectiveness of the proposed framework.

The SemanticKITTI dataset [2] is a 3D large-scale outdoor urban scene dataset that includes 22 sequences and 19 categories. Following the setting of previous work [7], we select $\{other\text{-}vehicle\}$ as the unseen class.

The S3DIS dataset [1] is a 3D indoor scene dataset consisting of 271 rooms, which contains 13 categories. Since we are the first attempt to conduct open-set 3D semantic segmentation (O3D) on S3DIS dataset, we choose unseen classes elaborately. Here, we select $\{window, sofa\}$ as the unseen classes. Please refer to the supplementary material for the detailed principles of choosing the unseen classes.

4.2. Evaluation metrics

As done in previous work [7], we adopt three metrics for evaluation, including AUROC (Area Under the ROC curve), AUPR(Area Under the Precision-Recall curve), and mIoU_c (closed-set mean Intersection over Union). The AUROC and AUPR are used for measuring the open-set segmentation ability, which are more attuned to class imbalances and give a holistic measure of performance when the cutoff for detecting unseen-class data is not a priori obvious. And the mIoU_c is used for measuring the closed-set segmentation ability.

4.3. Implementation details

We use the SGD optimizer with initial learning rate, momentum, and weight decay setting to 0.5, 0.9, and 0.0001 respectively. The learning rate is dropped by 10% after each epoch. The hyper-parameters λ_{attr}, λ_{info}, λ_s, and λ_{adv} are set to 0.1, 0.1, 0.6, and 0.1.

4.4. Comparative evaluation

We firstly evaluate the proposed APF on the outdoor dataset SemanticKITTI in comparison to the SOTA method REAL [7] that is specially designed for O3D. In addition, we extend four typical O2D methods to handle the 3D case for further comparison, including MSP [18], Maxlogit [17], MC dropout [15], and DMLNet [6]. It has to be pointed out that considering the comparative method REAL uses Cylinder3D [41] as its backbone, all the other comparative methods (including the proposed method) are evaluated here by utilizing the same backbone for a fair comparison. The corresponding results are reported in Table 1. In addition, the closed-set result of Cylinder3D is reported as the upper bound for comparative closed-set evaluation.

As seen from Table 1, MSP, Maxlogit, and MC dropout obtain the same mIoU_c for measuring the closed-set performance as the backbone Cylinder3D, mainly because they do not only have the same or approximately same network architecture as Cylinder3D, but also have the same inference strategy for segmenting the closed-set points. DMLNet utilizes the fixed one-hot prototypes for segmentation, and may lose some essential information for large-scale point clouds segmentation, which results in the unsatisfactory mIoU_c, compared with other extended O2D methods. The methods which are tailor-made for O3D (including the proposed APF and REAL) obtain slightly lower mIoU_cs than the upper bound result, probably because the false positive open-set detection inevitably deteriorates the closed-set segmentation accuracy. Moreover, the proposed APF significantly outperforms all the comparative methods under the two OS metrics AUROC and AUPR. All the above results demonstrate that the proposed APF could achieve a better balance between open-set and closed-set segmentation performances.

In addition, we also evaluate all the referred methods on the public indoor dataset S3DIS. It is noted that REAL only conducts experiments on the outdoor datasets. Hence, we simply use the popular Point Transformer, whose effec-

	AUROC	AUPR	mIoU_c
Closed-set C3D	-	-	**58.0**
C3D + MSP [18]	74.0	6.7	**58.0**
C3D + MaxLogit [17]	70.5	7.6	**58.0**
C3D + MC dropout [15]	74.7	7.4	**58.0**
C3D + DMLNet [6]	80.6	20.1	52.9
C3D + REAL [7]	84.9	20.8	57.8
C3D + APF	**85.6**	**36.1**	57.3

Table 1. Evaluation results on SemanticKITTI dataset. C3D denotes Cylinder3D [41]. The best results are in bold in each metric.

	AUROC	AUPR	mIoU_c
Closed-set PT	-	-	**69.8**
PT + MSP	70.3	15.2	**69.8**
PT + MaxLogit	74.3	17.5	**69.8**
PT + MC dropout	75.9	18.2	**69.8**
PT + DMLNet	80.7	20.5	67.2
PT + REAL	87.6	25.4	69.7
PT + APF	**90.0**	**31.6**	69.3

Table 2. Evaluation results on S3DIS datset. PT denotes Point Transformer [40].

tiveness for handling indoor point clouds has been demonstrated in [40], as the backbone for evaluation on S3DIS dataset. The corresponding results are reported in Table 2. As seen from this table, the proposed method obtains a slightly lower mIoU_c, but significantly larger AUROC and AUPR than the other comparative methods. These experimental results are consistent with those in the above outdoor experiments, further demonstrating the effectiveness of the proposed method.

Moreover, we visualize the segmentation results on the two datasets by all the comparative methods, and Figure 7 shows several samples. The visualization results demonstrate that the proposed APF has a better performance in distinguishing the unseen-class points while yielding a promising performance in classifying the seen-class points.

4.5. Ablation study

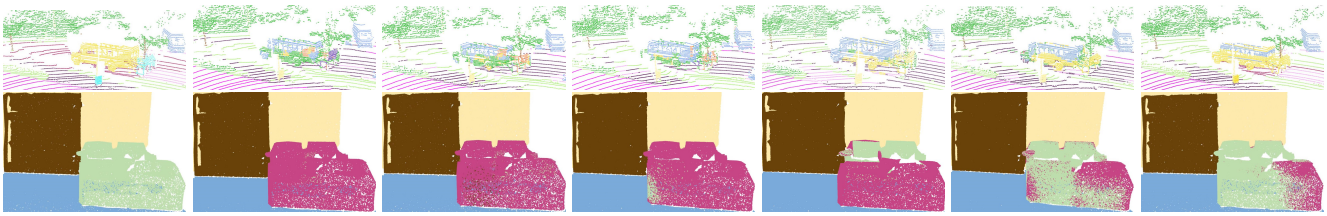
The effectiveness of each key element in APF is verified by conducting ablation studies on the S3DIS dataset [1]. The main results are reported and analyzed as follows, and please refer to the supplementary material for more details.

Effectiveness of the involved components. The proposed APF consists of a feature extraction module (FEM), a prototypical constraint module (PCM), and a feature adversarial module (FAM). And the FAM contains three key components: a GAN, a feature sampling (FS) strategy, and an adversarial mapper (M). We firstly train the vanilla version of APF, consisting of the FEM and PCM. Then, we sequentially add the three key components of FAM into the model. The results are reported in Table 3.

As seen from the first row of Table 3, the vanilla APF yields a promising performance, demonstrating the powerful potential of the prototype-based discriminative method in O3D. The results in the second row of Table 3 indicate that when the unseen-class features are taken into consideration, the open-set ability of the model is improved, but its closed-set ability evidently drops, mainly because the prototypes are influenced by the synthetic features while the seen-class features remain unchanged. When the feature sampling strategy is adopted, the sacrifice of the closed-set segmentation accuracy is reduced, owing to the filtering process which prevents the training procedure from being misled by the synthetic confusing features. When the

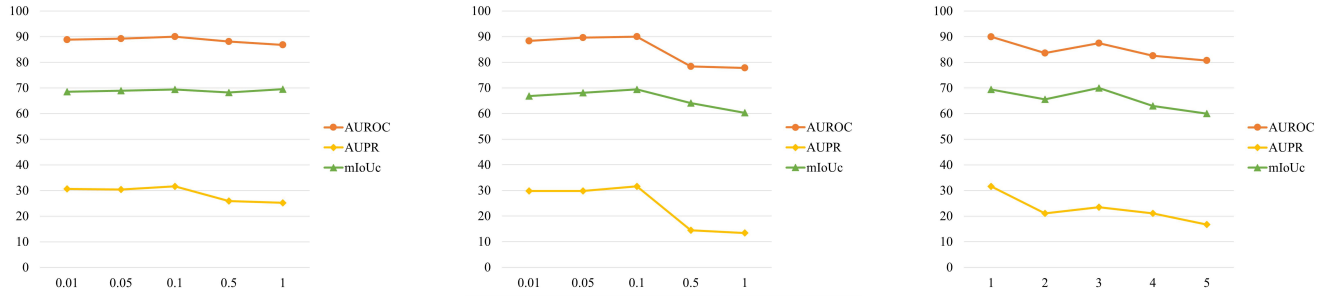
FEM	PCM	FAM			AUROC	AUPR	mIoU_c
		GAN	FS	M			
✓	✓				86.7	23.4	66.5
✓	✓	✓			88.6	26.7	63.1
✓	✓	✓	✓		88.7	26.9	64.3
✓	✓	✓	✓	✓	**90.0**	**31.6**	**69.3**

Table 3. Ablation studies of the involved components in APF.



(a) Ground truth (b) MSP [18] (c) MaxLogit [17] (d) MC dropout [15] (e) DMLNet [6] (f) REAL [7] (g) APF

Figure 7. Visualization of open-set semantic segmentation results on SemanticKITTI [2] (top) and S3DIS [1] (bottom) datasets by the proposed APF and the comparative methods. The unseen-class points are colorized in yellow (*other-vehicle*) and light green ($sofa$) respectively, while the seen-class points are colorized in other colors.



(a) λ_{attr} (b) λ_{info} (c) Prototype number in each seen class.

Figure 8. Ablation studies of λ_{attr}, λ_{info} and prototype number in each seen class.

adversarial mapper is added into training, the open-set and closed-set ability of the model are both boosted, for the adversarial mapper is designed to refine the feature distribution so that the seen-class features and unseen-class features become more distinguishable.

Effectiveness of the loss terms in PCM. The proposed PCM utilizes three loss terms: $\mathcal{L}_{dce}$, $\mathcal{L}_{attr}$, and $\mathcal{L}_{info}$, as noted in Equation (6). We evaluate their effects on the final segmentation performance, and the results are reported in Table 4. As seen from this table, when $\mathcal{L}_{attr}$ or $\mathcal{L}_{info}$ is added into training, the performance of APF is evidently improved, demonstrating the effectiveness of our designed loss terms. We further investigate the effect of different weights λ_{attr} and λ_{info} for L_{attr} and L_{info} respectively. The results in Figure 8a indicate that the proposed method is relatively insensitive to the weights of our designed loss terms when the weights range in [0.01, 0.1].

Effect of prototype number. In previous experiments, we only maintain one prototype for each seen class. Hence, we evaluate the effect of prototype number, and the results in Figure 8c attest that increasing prototype number does not promote the performance significantly. In contrast, multiple prototypes brings more complexity to the model and makes the cluster distribution in the feature space not tight enough, and thus deteriorates the performance.

Effect of unseen classes. We choose $\{window, sofa\}$ as the unseen classes in previous experiments. To verify the robustness of the proposed APF, we select different classes as the unseen classes. Considering that REAL already outperforms the other comparative methods, we only compare the performance of the closed-set backbone, REAL, and APF here. The results in Table 5 indicate that APF still outperforms REAL in AUROC and AUPR, with a slight sacrifice in mIoU_c, which is consistent with the results in Table 1 and Table 2.

$\mathcal{L}_{dce}$	$\mathcal{L}_{attr}$	$\mathcal{L}_{info}$	AUROC	AUPR	mIoU_c
✓			79.3	12.2	62.0
✓	✓		80.1	14.8	64.2
✓		✓	85.2	20.6	67.7
✓	✓	✓	**90.0**	**31.6**	**69.3**

Table 4. Ablation studies of the loss terms in PCM.

Unseen classes	Method	AUROC	AUPR	mIoU_c
$\{window, door\}$	Closed-set PT	-	-	**70.7**
	PT + REAL	86.5	26.3	69.1
	PT + APF	**87.1**	**33.1**	68.4
$\{sofa, board\}$	Closed-set PT	-	-	**69.6**
	PT + REAL	86.8	27.1	68.8
	PT + APF	**87.3**	**29.1**	68.0

Table 5. Ablation studies of unseen classes.

5. Conclusion

In this work, we introduce the Adversarial Prototype Framework (APF) for handling the open-set 3D semantic segmentation task, which contains a feature extraction module to extract features from original point clouds, a prototypical constraint module, and a feature adversarial module. The prototypical constraint module updates parametric prototypes for each seen class in a prototype-based clustering fashion. The feature adversarial module incorporates the synthetic unseen-class features into training, which further boosts the performance. Experimental results demonstrate that the proposed APF achieves a better balance in open-set and closed-set segmentation performances than the comparative methods.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. U1805264 and 61991423), the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No.XDB32050100), the Beijing Municipal Science and Technology Project (Grant No. Z211100011021004).

References

[1] Iro Armeni, Ozan Sener, Amir Roshan Zamir, Helen Jiang, Ioannis K. Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *CVPR*, pages 1534–1543, 2016. 6, 7

[2] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, C. Stachniss, and Juergen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *ICCV*, pages 9296–9306, 2019. 1, 6, 7

[3] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *CVPR*, pages 4182–4191, 2020. 1

[4] Petra Bevandic, Ivan Kreso, Marin Orsic, and Sinisa Segvic. Dense outlier detection and open-set recognition based on training with noisy negative images. *ArXiv*, abs/2101.09193, 2021. 3

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2

[6] Jun Cen, Peng Yun, Junhao Cai, Michael Yu Wang, and Ming Liu. Deep metric learning for open world semantic segmentation. In *ICCV*, pages 15313–15322, 2021. 1, 2, 5, 6, 7

[7] Jun Cen, Peng Yun, Shiwei Zhang, Junhao Cai, Di Luan, Michael Yu Wang, Meilin Liu, and Mingqian Tang. Open-world semantic segmentation for lidar point clouds. In *ECCV*, 2022. 1, 2, 3, 6, 7

[8] Guangyao Chen, Peixi Peng, Xiangqian Wang, and Yonghong Tian. Adversarial reciprocal points learning for open set recognition. *IEEE TPAMI*, 44:8065–8081, 2022. 3, 5

[9] Ran Cheng, Ryan Razani, Ehsan Moeen Taghavi, Enxu Li, and Bingbing Liu. (af)2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In *CVPR*, pages 12542–12551, 2021. 1

[10] Shuang Deng and Qiulei Dong. Ga-net: Global attention network for point cloud semantic segmentation. *IEEE SPL*, 28:1300–1304, 2021. 1

[11] Shuang Deng, Qiulei Dong, Bo Liu, and Zhanyi Hu. Superpoint-guided semi-supervised semantic segmentation of 3d point clouds. In *ICRA*, pages 9214–9220, 2022. 1

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2

[13] Francis Engelmann, Theodora Kontogianni, and B. Leibe. Dilated point convolutions: On the receptive field size of point convolutions on 3d point clouds. In *ICRA*, pages 9463–9469, 2020. 2

[14] Siqi Fan, Qiulei Dong, Fenghua Zhu, Yisheng Lv, Peijun Ye, and Feiyue Wang. Scf-net: Learning spatial contextual features for large-scale point cloud segmentation. In *CVPR*, pages 14499–14508, 2021. 2

[15] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016. 1, 2, 6, 7

[16] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, pages 9224–9232, 2018. 2

[17] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Xiaodong Song. Scaling out-of-distribution detection for real-world settings. In *ICML*, 2022. 1, 2, 6, 7

[18] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017. 1, 2, 6, 7

[19] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Agathoniki Trigoni, and A. Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *CVPR*, pages 11105–11114, 2020. 2

[20] Jaedong Hwang, Seoung Wug Oh, Joon-Young Lee, and Bohyung Han. Exemplar-based open-set panoptic segmentation network. In *CVPR*, pages 1175–1184, 2021. 2

[21] Shu Kong and Deva Ramanan. Opengan: Open-set recognition via open data generation. In *ICCV*, pages 793–802, 2021. 3, 5

[22] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *CVPR*, pages 8490–8499, 2022. 2

[23] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *CVPR*, pages 9659–9669, 2021. 1

[24] Krzysztof Lis, Krishna Kanth Nakka, Pascal V. Fua, and Mathieu Salzmann. Detecting the unexpected via image resynthesis. In *ICCV*, pages 2152–2161, 2019. 3

[25] Kevin P. Murphy. Machine learning - a probabilistic perspective. In *Adaptive computation and machine learning series*, 2012. 5

[26] Hyunjong Park, Jongyoun Noh, and Bumsub Ham. Learning memory-guided normality for anomaly detection. In *CVPR*, pages 14360–14369, 2020. 1

[27] C. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 77–85, 2017. 2

[28] C. Qi, L. Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 2

[29] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. In *NeurIPS*, 2022. 1

[30] Walter J. Scheirer, Anderson Rocha, Archana Sapkota, and Terrance E. Boult. Toward open set recognition. *IEEE TPAMI*, 35:1757–1772, 2013. 5

[31] Hang Su, V. Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. Splatnet: Sparse lattice networks for point cloud processing. In *CVPR*, pages 2530–2539, 2018. 2

[32] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. In *CVPR*, pages 3887–3896, 2018. 2

[33] Hugues Thomas, C. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *ICCV*, pages 6410–6419, 2019. 2

[34] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[35] Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep parametric continuous convolutional neural networks. In *CVPR*, pages 2589–2597, 2018. 2

[36] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *ICRA*, pages 1887–1893, 2018. 2

[37] 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 *ICRA*, pages 4376–4382, 2019. 2

[38] Yingda Xia, Yi Zhang, Fengze Liu, Wei Shen, and Alan Yuille. Synthesize then compare: Detecting failures and anomalies for semantic segmentation. In *ECCV*, 2020. 3

[39] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, Qing Yang, and Cheng-Lin Liu. Convolutional prototype network for open set recognition. *IEEE TPAMI*, 44:2358–2370, 2022. 3

[40] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H. S. Torr, and Vladlen Koltun. Point transformer. In *ICCV*, pages 16239–16248, 2021. 2, 7

[41] 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 *CVPR*, pages 9934–9943, 2021. 2, 6, 7