SceneEncoder: Scene-Aware Semantic Segmentation of Point Clouds with A Learnable Scene Descriptor

Jiachen Xu[1*], **Jingyu Gong**[1*], **Jie Zhou**[1], **Xin Tan**[1,3], **Yuan Xie**[2†] and **Lizhuang Ma**[1,2†]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
[2]School of Computer Science and Technology, East China Normal University, Shanghai, China
[3]Department of Computer Science, City University of Hong Kong, HKSAR, China
{xujiachen, gongjingyu, lord_liang, tanxin2017}@sjtu.edu.cn,
xieyuan8589@foxmail.com, ma-lz@cs.sjtu.edu.cn

Abstract

Besides local features, global information plays an essential role in semantic segmentation, while recent works usually fail to explicitly extract the meaningful global information and make full use of it. In this paper, we propose a SceneEncoder module to impose a scene-aware guidance to enhance the effect of global information. The module predicts a scene descriptor, which learns to represent the categories of objects existing in the scene and directly guides the point-level semantic segmentation through filtering out categories not belonging to this scene. Additionally, to alleviate segmentation noise in local region, we design a region similarity loss to propagate distinguishing features to their own neighboring points with the same label, leading to the enhancement of the distinguishing ability of point-wise features. We integrate our methods into several prevailing networks and conduct extensive experiments on benchmark datasets ScanNet and ShapeNet. Results show that our methods greatly improve the performance of baselines and achieve state-of-the-art performance.

1 Introduction

As a basic task in 3D vision, semantic segmentation of point clouds has drawn more and more attention. Due to the irregularity and disorder of point clouds, many previous works convert point clouds into the grid representation through voxelization or projection to leverage the effectiveness of grid convolution [Graham *et al.*, 2018; Su *et al.*, 2015]. These methods inevitably destroy the original geometric information of point clouds. Therefore, PointNet [Qi *et al.*, 2017a] directly processes the raw point clouds and extracts features with shared Multi-Layer Perceptrons (MLPs). However, because of the unusual properties of point clouds and the complexity of scene segmentation, semantic segmentation of point clouds remains a challenging issue.

Global information, which is commonly extracted by the encoder network, usually contains the essential knowledge that directly summarizes the information of the whole scene, thus should play a more significant role in the semantic segmentation. For human beings, the prior knowledge towards a scene could directly influence the semantic comprehension. For instance, as in Figure 1, when being in a bathroom, it is easy to distinguish the "shower curtain" category from the "curtain" category even they look similar. *Therefore, the global information of point clouds can play the role of prior knowledge for semantic segmentation.*

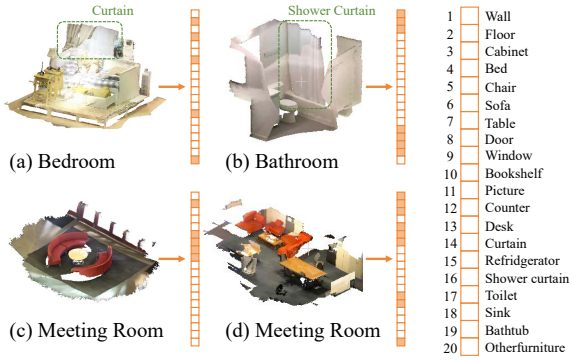

Figure 1: Illustration of scene descriptors for different scenes. The target scene descriptor is a binary vector for each scene. The i-th element of scene descriptor is 1 if the i-th category exists in the scene and 0 if not. Make an analogy with grids. The colored grids represent 1 and the empty grids represent 0.

Many previous works utilize the U-Net architecture to extract global features with growing receptive field, and then mix these global features with local features by concatenation hierarchically [Wu *et al.*, 2019; Graham *et al.*, 2018]. However, such a kind of mixture not only attenuates the guidance of global information, but also degrades the representation capability of local features. SpiderCNN [Xu *et al.*, 2018] adopts a more direct way to take the global label of a point cloud as a part of elements in feature vectors to enhance the effect of the global information. However, it is based on an impractical assumption that the global label is always available, especially in testing phase.

Therefore, we propose a SceneEncoder module to predict a multi-hot scene descriptor for every point cloud, a novel representation of global label rather than the manually annotated

[*]Equal Contribution.
[†]Corresponding Author.

one-hot scene label, to perform a scene-level guidance. An ideal scene descriptor is designed to be a binary vector with each element representing the existence of corresponding category of object. As shown in Figure 1(a), the scene descriptor of a bedroom, consisting of floor, bed, table, curtain and some other furniture, can be represented by a 5-hot vector.

Different from concatenating global features to local features, we utilize the scene descriptor as a mask/attention to aid the point-level semantic segmentation by filtering out irrelevant object categories that are impossible to exist in the current scene. For example, there is a curtain and a shower curtain in Figure 1(a) and (b) respectively, which are difficult to distinguish due to the similar appearance. However, our descriptors can directly help to exclude the classification option of the shower curtain in bedroom scenes and exclude the curtain in the bathroom. Besides, compared with a one-hot scene label, our well-designed multi-hot scene descriptor is also able to subdivide each category of scenes. As shown in Figure 1(c) and (d), meeting rooms which are designed for different purposes can be further represented by different scene descriptors. To train the SceneEncoder module, we dynamically generate the ground truth of scene descriptors by checking which categories of objects exist in the input training scene point clouds based on labels of all points.

Since the segmentation task is a point-level task, global information is not enough for point-level classification. Therefore, an increasing number of works [Zhao *et al.*, 2019; Wu *et al.*, 2019] focus on exploiting the local context which is important for recognizing fine-grained patterns and generalizing to complex scenes. A problem in the prevailing methods is that the extracted feature usually involves the features of different categories in the local region. For example, the point clouds belonging to a chair might be nearby a table, such that the extracted local feature would incorporate features of both the chair and table. This would make point features less distinguishing because these features encode more than one class information, and it also results in poor object contours in the segmentation task. To tackle this problem, we design a novel loss, namely region similarity loss, to propagate distinguishing point features to ambiguous features in local regions. That is, we only propagate each distinguishing feature to its adjacent points (*i.e.,* points in its neighborhood) with same labels since the points in the same local region but with different categories would differ on their features. Overall, the major contributions can be summarized as follows:

- We design a SceneEncoder module with a scene descriptor to provide a scene-level guidance, leading to an effective collaboration of the global information and local features for semantic segmentation.
- We propose a novel region similarity loss to propagate distinguishing point features to ambiguous features in local regions so as to alleviate the segmentation noise effectively.
- We integrate our SceneEncoder module into two prevailing networks by involving the region similarity loss. The experimental results on two benchmark datasets demonstrate that our method outperforms many state-of-the-art competitors.

2 Related Work

Segmentation on point clouds. Impressively, PointNet [Qi *et al.*, 2017a] processed the raw point clouds directly and extracted point features with MLPs. Built upon it, PointNet++ [Qi *et al.*, 2017b] designed a hierarchical framework to exploit local context with growing receptive fields. However, the key operation to aggregate features in both methods is max-pooling, which leads to great loss of context information. To utilize the local context efficiently, PointWeb [Zhao *et al.*, 2019] densely connected each pair of points in the local region and adjusted each point feature adaptively. An independent edge branch was introduced in [Jiang *et al.*, 2019] to exploit the relation between neighboring points at different scale and interwove with the point branch to provide more context information. PointConv [Wu *et al.*, 2019] focused on the distribution of points in the local region and utilized density information. Furthermore, an octree guided selection strategy was proposed in [Lei *et al.*, 2019] to guide the convolution process of point clouds in a more reasonable way. Compared with these methods which utilize the scene information by fusing local and global features together, we directly predict a scene descriptor from global features to guide the point-level segmentation so as to enhance the effect of the global information.

Segmentation refinement. Additionally, noises and poor contours are usually caused by ambiguous features because they are mixed with the features of different categories in the local region. SEGCloud [Tchapmi *et al.*, 2017] proposed to combine networks with fine-grained representation of point clouds using 3D Conditional Random Fields (CRF). GAC [Wang *et al.*, 2019] determined how each point feature contribute to the extracted feature by the similarity between these two features to improve the distinguishability of features. By contrast, we propose a more general loss to refine the results of segmentation by propagating distinguishing features in the local region. This strategy could be adopted in the training process of most networks in semantic segmentation task.

Loss function. In [De Brabandere *et al.*, 2017], a widely used loss was proposed to minimize the difference between point features of the same instance object. Similarly, Pairwise Similarity Loss [Engelmann *et al.*, 2018] maximized the similarity among point features of the same semantic object. It is noteworthy that points which are far apart should have different features even if they belong to the same category. Therefore, our loss just focus on forcing adjacent point features of the same category to be similar.

3 Method

First we introduce the overall architecture in Sec. 3.1. Then, the SceneEncoder module and the scene descriptor will be described in details in Sec. 3.2. In Sec. 3.3, the novel region similarity loss would be depicted. Finally, we summarize different losses used in the training process of the whole network in Sec. 3.4.

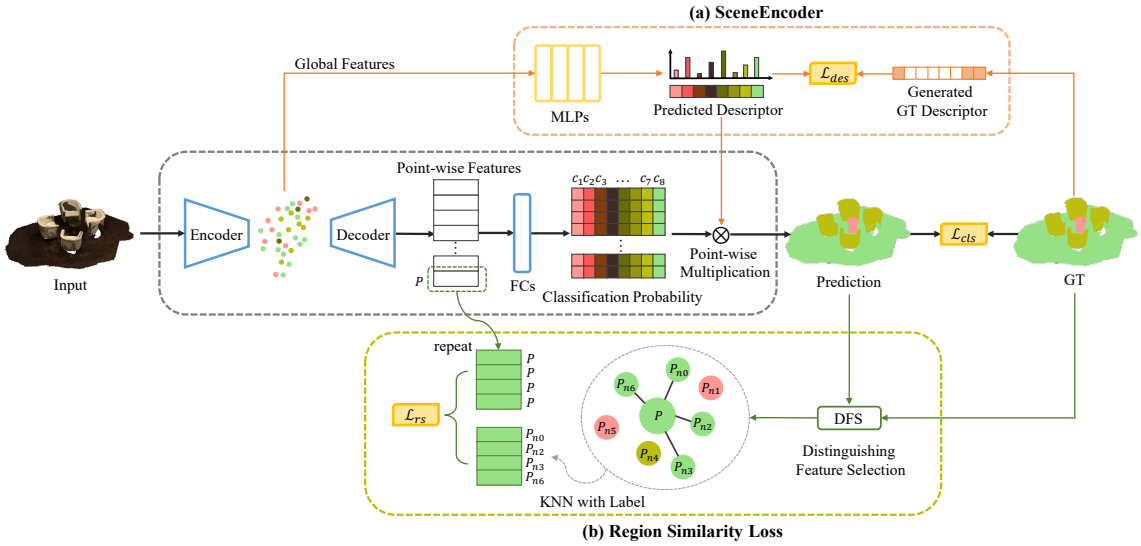

Figure 2: Overall architecture of our network. (a) represents the SceneEncoder module in which the predicted descriptor is directly regularized by a well-designed scene descriptor and aids the final segmentation. (b) stands for the region similarity loss in which distinguishing features help to refine other ambiguous features.

3.1 Overview

Figure 2 illustrates our overall architecture. We encode the raw point cloud into few global features, which will be taken as the input into the proposed SceneEncoder module (Figure 2(a)). The output of this module, *i.e.,* the predicted scene descriptor, would be used to point-wise multiply with classification probability vector of each point, so as to exclude the impossible category options and get final prediction. Furthermore, during the training process, we select M distinguishing points that are correctly classified and their corresponding features from the feature map. Then, we define a region similarity loss (Figure 2(b)) to propagate distinguishing features by improving the feature similarity between each distinguishing point and its neighboring points with the same label.

3.2 SceneEncoder Module

The global information can give a top-down guidance to point-level classifier. To make full use of the global information as the scene prior knowledge, we propose a SceneEncoder module with a well-designed multi-hot scene descriptor to enhance the effect of global information in the semantic comprehension. This descriptor is able to substitute the global scene label, and acts as a mask/attention to eliminate impossible results when classifying each point.

Multi-hot Scene Descriptor

As the recognition of a scene can directly influence the result of point-level semantic segmentation, we propose to use a scene descriptor to represent the global scene information. Similar to [Gupta *et al.*, 2013], we also use the objects existing in the scene to represent global information. By contrast, our scene descriptor focuses on whether the corresponding object exists rather than the proportion and work as a mask to filter out impossible categories rather than an additional feature. For scene semantic segmentation task with n object categories, we design a multi-hot vector of length n with each element representing the probability of existence of the corresponding category in the point cloud. Specifically, let $\widetilde{\mathbf{g}}$ denote the predicted descriptor, the i-th element of this descriptor $\widetilde{\mathbf{g}}_i$ will be 0 if the i-th category does not exist in this point cloud. Then, in semantic segmentation, we could apply softmax on the output to obtain the classification probability map $\widetilde{P} \in \mathbb{R}^{N \times n}$, where N denotes the number of points, n is the number of classes, and $\widetilde{P}(i,j)$ represents the probability that point i belongs to class j. To make the scene descriptor directly guide the semantic segmentation, we directly multiply the probability map $\widetilde{P}$ with the predicted descriptor $\widetilde{\mathbf{g}}$ to achieve the refined probability map $\widetilde{P}_{ref}$ as follow.

$$\widetilde{P}_{ref}(i,j) = \frac{\widetilde{\mathbf{g}}_j \cdot \widetilde{P}_{(i,j)}}{\sum_{j=1}^{n} \widetilde{\mathbf{g}}_j \cdot \widetilde{P}_{(i,j)}} \tag{1}$$

where $\widetilde{P}_{ref}(i,j)$ is the final predicted probability that point i belongs to class j, and $\widetilde{\mathbf{g}}_j$ indicates the probability of existence of the class j in the whole scene.

In this way, global information plays an important and influential role of prior knowledge in semantic segmentation by filtering out impossible results. Obviously, two scenes with the same descriptor are likely to be the same type of scene, and this makes our descriptor be able to substitute the scene label. Furthermore, compared with the manually annotated label, our scene descriptor could subdivide each scene according to the object composition and provide a more concrete global information.

Supervision for Scene Descriptor

Compared with the SpiderCNN [Xu *et al.*, 2018] that requires the ground truth of global label during both training and testing, we generate the scene descriptor through the SceneEncoder module. To effectively train the SceneEcoder module,

we regularize the predicted scene descriptor $\widetilde{\mathbf{g}}$. Instead of labeling the ground truth of each scene descriptor $\mathbf{g}$ manually, we generate it on-the-fly from the labels of all points in the point cloud during the training process. Note that, the ground truth of the scene descriptor is a binary vector of length n, and each element represents the existence of corresponding category in this scene. Therefore, through checking the labels of all the points, we can confirm which classes appear in this point cloud easily. Then, the regularization is as follow:

$$\mathcal{L}_{des} = -\sum_{j=1}^{n} \mathbf{g}_j log(\widetilde{\mathbf{g}}_j). \tag{2}$$

In the training phase, the SceneEncoder module could learn to represent the type of each scene by encoding the global information into the scene descriptor. As for the validation and testing process, the scene descriptor could be predicted by the well-trained SceneEncoder module and directly aid point-level prediction through Eq. (1).

3.3 Region Similarity Loss

Compared with global information, local details could help to generalize to complex scenes. However, during feature aggregation process, previous works ignore the difference among points with different labels in each local context. Such a kind of label inconsistency will lead to the lack of distinguishing ability of point features. Consequently, there are usually poor contours and noisy regions in the segmentation results. Therefore, we propose a novel region-based loss to propagate distinguishing features in local regions, thus those nondistinguishing features of neighbors with the same label can be refined.

Distinguishing Feature Selection

Compared with features achieved from intermediate layers, point features of the last layer encode more local and global information, and directly affect the segmentation results. Hence, our proposed loss directly refines point features of the last feature abstraction layer.

As shown in Figure 2, to select distinguishing point features, we first choose a set of correctly classified points $\mathcal{P}$. Then, among these points, we analyze two strategies to select a fixed number M of points. The first strategy is randomly selection, while the second one is to pick points with the top M classification confidence from $\mathcal{P}$ and classification confidence refers to the predicted probability. However, the first strategy is too arbitrary so that some correctly classified points with low confidence may also be selected, leading to incorporate indistinguishable features.

By contrast, the second strategy is more reasonable, which will be confirmed in Sec. 4.3. In practice, if the amount of correctly classified points is less than the given number M, which usually happens in the beginning of training process, the point feature with the highest classification confidence is sampled repeatedly.

Feature Adjustment

Based on the selected distinguishing point features, we adjust other features to boost their distinguishing ability. First, since points of different categories should have different features, we only propagate distinguishing features to points of the same category. Then, we achieve the feature propagation by reducing the difference between each distinguishing feature and features of points in its local neighborhood instead of the whole scene. Because in most cases, features of points that are far away from each other are likely to be dissimilar even if they belong to the same category. If we force to reduce the difference between two features but their corresponding points are far apart, it would hinder the feature abstraction layers from learning the correct point features, making the deep network hard to train.

Therefore, as shown in Figure 2(b), we utilize a KNN with label strategy to select neighboring points of same category, and only propagate each distinguishing feature in corresponding neighborhood. Concretely, for each point p_i with distinguishing feature, its k nearest neighbors $p_{n_{i1}}, p_{n_{i2}}, ..., p_{n_{ik}}$ with the same label are adjusted. To minimize the difference between each neighboring point feature and the center distinguishing point feature, we use the cosine similarity to define the region similarity loss as follows:

$$\begin{aligned} \mathcal{L}_{rs} &= -\frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{k} cos\_sim(f_{p_i}, f_{p_{n_{ij}}}) \\ &= -\frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{k} \frac{f_{p_i} \cdot f_{p_{n_{ij}}}}{max(\|f_{p_i}\|_2 \cdot \|f_{p_{n_{ij}}}\|_2, \epsilon)} \end{aligned} \tag{3}$$

where M is the number of distinguishing point features and k is the number of selected neighboring points. f_{p_i} and $f_{p_{n_{ij}}}$ are features of center point and neighboring point, respectively.

In addition to cosine similarity, Euclidean distance can also be used as a distance measurement. However, it is sensitive to the value of each dimension of each point feature, reducing generalization ability of the loss. On the contrary, cosine similarity could make our proposed loss more robust to point clouds in different scales.

3.4 Total Loss

In summary, the total loss function consists of three parts: $\mathcal{L}_{cls}$, $\mathcal{L}_{des}$, and $\mathcal{L}_{rs}$. $\mathcal{L}_{cls}$ is the cross entropy loss applied to constrain the point-level predictions. $\mathcal{L}_{des}$ is defined in Eq. (2) to improve the performance of SceneEncoder module. Furthermore, $\mathcal{L}_{rs}$ is the proposed region similarity loss to boost the distinguishing ability of point features as in Eq. (3). Therefore, the total loss function is as follows:

$$\mathcal{L} = \lambda_1\mathcal{L}_{cls} + \lambda_2\mathcal{L}_{des} + \lambda_3\mathcal{L}_{rs}, \tag{4}$$

where λ_1, λ_2 and λ_3 adjust ratio of the three losses.

In the training process, we inhibit the gradient back propagation from $\mathcal{L}_{cls}$ to $\widetilde{g}$, that is to prevent the scene descriptor from being regularized by $\mathcal{L}_{cls}$. Because, $\mathcal{L}_{cls}$ is a point-level classification loss while our scene descriptor is a global descriptor, which would lower the training performance. In our experiments, λ_1 and λ_2 are set the same, but we dynamically change λ_3 during training. In the beginning of training process the selected features are not distinguishing enough, so

Method	xyz	rgb	mIoU
PointNet++ [Qi *et al.*, 2017a]	✓		33.9
PointCNN [Li *et al.*, 2018b]	✓	✓	45.8
3DMV [Dai and Nießner, 2018]	✓	✓	48.4
PointConv [Wu *et al.*, 2019]	✓		55.6
TextureNet [Huang *et al.*, 2019]	✓	✓	56.6
HPEIN [Jiang *et al.*, 2019]	✓	✓	61.8
Ours	✓		**62.8**

Table 1: Semantic segmentation results on ScanNet v2.

Method	mcIoU	mIoU
PointNet++ [Qi *et al.*, 2017b]	81.9	85.1
SO-Net [Li *et al.*, 2018a]	81.0	84.9
PCNN by Ext [Atzmon *et al.*, 2018]	81.8	85.1
SpiderCNN [Xu *et al.*, 2018]	82.4	85.3
ShellNet [Zhang *et al.*, 2019]	82.8	-
PointConv [Wu *et al.*, 2019]	82.8	**85.7**
Ours	**83.4**	**85.7**

Table 2: Part segmentation results on ShapeNet.

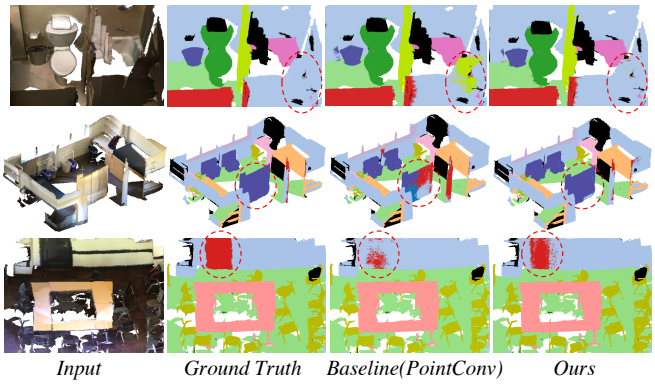

Figure 3: Visualization results for semantic segmentation on ScanNet. The images from left to right are the original scenes, the segmentation ground truth, predictions given separately by PointConv and the proposed methods.

the impact to other features should be weakened. Therefore, we assign a small weight to λ_3 in the beginning and increase the weight gradually until it is the same as λ_1 and λ_2. In the testing process, we do not generate the ground truth of scene descriptor any more.

4 Experiments

Experiments consist of three parts. First, we demonstrate the effectiveness of our method on semantic segmentation task in Sec 4.1. Second, we generalize our method to part segmentation and prove its effectiveness on part segmentation in Sec 4.2. Finally, we conduct ablation study to demonstrate the effectiveness of the proposed SceneEncoder module and region similarity loss in Sec 4.3.

Metric. Like previous works, we take the intersection-over-union (IoU) as our metric, *i.e.,* mean IoU over categories for semantic segmentation task, class average IoU and instance average IoU for part segmentation task.

4.1 Scene Semantic Segmentation

Dataset. We evaluate the performance of scene semantic segmentation task on ScanNet v2 [Dai *et al.*, 2017]. ScanNet v2 consists of 1,201 scanned scenes for training and 312 scenes for validation. Another 100 scenes are provided as the testing dataset. Different from most previous works using both geometry and color information, we just take the geometry information (*i.e.,* xyz) as the input.

Implementation details. We insert our SceneEncoder module into PointConv [Wu *et al.*, 2019], in which point features of the last layer are regularized by our region similarity loss during training process. Following [Wu *et al.*, 2019], training samples are generated by randomly sampling 3m×1.5m×1.5m cubes from rooms, and then we test over the entire scan. The network is trained by using Adam optimizer with batch size 8 on a single GTX 1080Ti GPU.

Results. We report mean IoU (mIoU) over categories in Table 1, where a clear improvement over backbone (PointConv) can be observed, *i.e.,* 7.2% in mIoU, and our method achieves a state-of-the-art performance in ScanNet benchmark. As aforementioned, we only take the location information as our input and it also performs better than lots of methods that use extra color information. Figure 3 visualizes the scene semantic segmentation results of PointConv [Wu *et al.*, 2019] and our methods.

4.2 Part Segmentation

Similarly, our methods can also improve the performance on part segmentation. In part segmentation task that purposes to segment the functional part of each object, we can treat each object as the entire scene point cloud, and the parts of each object can be considered as the different objects in the scene. Therefore, to confirm the generalization ability, we evaluate our method on part segmentation.

Dataset. We evaluate the performance on ShapeNet part data [Chang *et al.*, 2015]. ShapeNet part data consists of 16,881 point clouds from 16 categories and totally contains 50 parts, with xyz and norm vectors being taken as the input.

Implementation details. We still choose PointConv as the baseline and integrate our method into it. We directly take the raw point cloud as input without preprocessing. The Adam optimizer is employed to train the model with our region similarity loss on a single GTX 1080Ti GPU.

Results. We report class average IoU (mcIoU) and instance average IoU (mIoU) for part segmentation in Table 2. The 0.6% increment over PointConv in term of mcIoU shows the great potential of the proposed method in part segmentation. Overall, the improvements on both scene semantic segmentation task and part segmentation task show the generality of our method. However, even obvious improvements have been shown in mcIoU, there is little increase in mIoU, seeing the rightmost column in Table 2. The reason might be attributed to the fact that: the target scene descriptors are very similar for instance of the same categories, while the predict scene

Method	mcIoU	mIoU	aero	bag	cap	car	chair	eph.	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate.	table
SpiderCNN	81.8	84.8	82.5	77.2	85.7	76.9	90.4	77.0	90.9	**87.6**	81.7	**95.8**	68.3	94.6	81.2	59.4	76.6	82.8
SpiderCNN + category	82.4	85.3	83.5	81.0	87.2	77.5	**90.7**	76.8	**91.1**	87.3	83.3	**95.8**	70.2	93.5	**82.7**	59.7	75.8	82.8
SpiderCNN + SceneEncoder	82.4	85.3	**83.9**	80.5	83.9	78.6	90.6	**80.5**	90.7	87.2	**83.8**	95.7	69.7	94.4	81.6	58.4	76.5	82.6
SpiderCNN + category + SceneEncoder	**83.1**	**85.4**	**83.9**	**83.3**	**87.7**	**78.7**	90.4	79.9	**91.1**	87.2	83.0	95.7	**71.3**	**94.7**	81.3	**61.7**	**77.3**	**83.1**

Table 3: Compare the effectiveness of ground truth of category label with SceneEncoder module based on SpiderCNN. The top line removes the ground truth of category label used in SpiderCNN. Then, category label and SceneEncoder module are added into SpiderCNN separately. The bottom line shows the result where both ground truth of category label and SceneEncoder module are used in SpiderCNN.

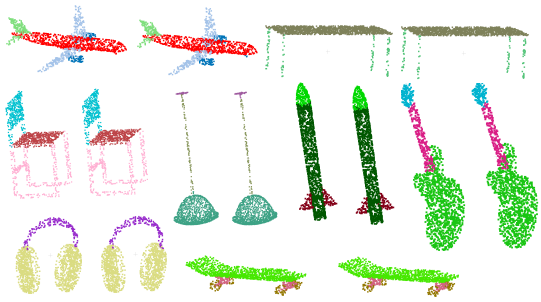

Figure 4: Visualization of part segmentation on ShapeNet. For each pair of objects, the left one is the ground truth while the right one is the predicted result.

descriptor is able to learn the pattern for one categories but not good at distinguishing the subtle difference among instances. The visualization of some results is shown in Figure 4.

4.3 Ablation Study

To prove that scene descriptor could describe the global information and substitute the global category label, we conduct experiments on ShapeNet with SpiderCNN [Xu *et al.*, 2018] as the backbone, since it directly adopts category label. Then, without loss of generality, we conduct more ablation study on semantic segmentation task on ScanNet and choose PointConv as the baseline.

SceneEncoder module versus category information. For part segmentation, category labels are not always available, as for scene semantic segmentation, it is even hard to give a label to a scene. By contrast, our SceneEncoder module learns to represent the global information with the scene descriptor extracted from the point cloud, even with no extra information of category label. To show the effectiveness of scene descriptor in representing global information, we conduct the ablation study for different combinations with SpiderCNN as backbone. [Xu *et al.*, 2018] releases the result of SpiderCNN with category information, and we conduct three extra experiments under the same setting, *i.e.,* SpiderCNN with no category information, SpiderCNN with SceneEncoder module, and SpiderCNN with both category information and SceneEncoder module. As shown in Table 3, SpiderCNN with SceneEncoder module but without category information perform as well as SpiderCNN with only category information. Therefore, it indicates that the scene descriptor could also represent the global information and substitute the global category label. Additionally, compared with [Xu *et al.*, 2018], combining the SceneEncoder and category label can increase

Method	mIoU
PointConv	55.6
PointConv + SceneEncoder	58.6
PointConv + RSL	58.7
PointConv + SceneEncoder + RSL	62.8

Table 4: Ablation study for impact of SceneEncoder module and region similarity loss.

Selection Strategy	mIoU
Select points randomly	60.2
Select points with high confidence	62.8

Table 5: Ablation Study that whether choosing points randomly or according to the maximum classification confidence.

the class average IoU by 0.7%. This result illustrates that the SceneEncoder enhances the effect of global information.

SceneEncoder module and region similarity loss. To evaluate the impact of the SceneEncoder module and the region similarity loss, we conduct two experiments on ScanNet. First, we insert our SceneEncoder module into the PointConv and train the model without region similarity loss. Second, we just introduce the region similarity loss into training process of the PointConv without the SceneEncoder module. The results are shown in Table 4, where we can observe that the SceneEncoder module and region similarity loss individually improve the performance. The combination can deal with both the global and local feature in a better way, hence further improve the performance of PointConv by a large margin.

Selection strategy of region similarity loss. In order to show the effectiveness of our selection strategy of distinguishing point features, we design experiments to study the performance of the region similarity loss using another selection strategy. Namely, we select distinguishing point features among correctly classified points randomly. As shown in Table 5, selecting distinguishing point features with high confidence could help to improve 2.5% on mIoU, which proves the effectiveness of our proposed selection strategy.

5 Conclusion

In this paper, we propose a scene-aware semantic segmentation method with a SceneEncoder module and a region similarity loss. The SceneEncoder module makes full utilization of global information to predict a scene descriptor and this scene descriptor can aid the point level semantic segmentation by filtering out impossible results. To further refine

the contour of segmentation, we propose the region similarity loss to propagate distinguishing features by forcing points in each local region with same labels to have similar final features. Overall, the proposed methods achieve the state-of-the-art performance on ScanNet dataset for semantic segmentation and ShapeNet dataset for part segmentation.

Acknowledgments

We thank for the support from National Natural Science Foundation of China (61972157, 61902129, 61772524, 61876161, 61701235, 61373077, 61602482), National R&D Program (SQ2019YFC150159), Shanghai Pujiang Talent Program (19PJ1403100), Beijing Municipal Natural Science Foundation (4182067), Economy and Information Commission of Shanghai (XX-RGZN-01-19-6348), Fundamental Research Funds for the Central Universities associated with Shanghai Key Laboratory of Trustworthy Computing. Xin Tan is also supported by the Postgraduate Studentship (by Mainland Schemes) from City University of Hong Kong. Jingyu Gong is also supported by Wu Wen Jun Honorary Doctoral Scholarship, AI Institute, Shanghai Jiao Tong University.

References

[Atzmon *et al.*, 2018] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *ACM Transactions on Graphics(TOG)*, 37(4):71, 2018.

[Chang *et al.*, 2015] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[Dai and Nießner, 2018] Angela Dai and Matthias Nießner. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In *ECCV*, pages 452–468, 2018.

[Dai *et al.*, 2017] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017.

[De Brabandere *et al.*, 2017] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551*, 2017.

[Engelmann *et al.*, 2018] Francis Engelmann, Theodora Kontogianni, Jonas Schult, and Bastian Leibe. Know what your neighbors do: 3d semantic segmentation of point clouds. In *ECCV*, pages 0–0, 2018.

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

[Gupta *et al.*, 2013] Saurabh Gupta, Pablo Arbelaez, and Jitendra Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *CVPR*, pages 564–571, 2013.

[Huang *et al.*, 2019] Jingwei Huang, Haotian Zhang, Li Yi, Thomas Funkhouser, Matthias Nießner, and Leonidas J Guibas. Texturenet: Consistent local parametrizations for learning from high-resolution signals on meshes. In *CVPR*, pages 4440–4449, 2019.

[Jiang *et al.*, 2019] Li Jiang, Hengshuang Zhao, Shu Liu, Xiaoyong Shen, Chi-Wing Fu, and Jiaya Jia. Hierarchical point-edge interaction network for point cloud semantic segmentation. In *ICCV*, pages 10433–10441, 2019.

[Lei *et al.*, 2019] Huan Lei, Naveed Akhtar, and Ajmal Mian. Octree guided cnn with spherical kernels for 3d point clouds. In *CVPR*, pages 9631–9640, 2019.

[Li *et al.*, 2018a] Jiaxin Li, Ben M Chen, and Gim Hee Lee. So-net: Self-organizing network for point cloud analysis. In *CVPR*, pages 9397–9406, 2018.

[Li *et al.*, 2018b] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *NIPS*, pages 820–830, 2018.

[Qi *et al.*, 2017a] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017.

[Qi *et al.*, 2017b] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, pages 5099–5108, 2017.

[Su *et al.*, 2015] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *ICCV*, pages 945–953, 2015.

[Tchapmi *et al.*, 2017] Lyne Tchapmi, Christopher Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3d point clouds. In *3DV*, pages 537–547. IEEE, 2017.

[Wang *et al.*, 2019] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In *CVPR*, pages 10296–10305, 2019.

[Wu *et al.*, 2019] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *CVPR*, pages 9621–9630, 2019.

[Xu *et al.*, 2018] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In *ECCV*, pages 87–102, 2018.

[Zhang *et al.*, 2019] Zhiyuan Zhang, Binh-Son Hua, and Sai-Kit Yeung. Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics. In *ICCV*, pages 1607–1616, 2019.

[Zhao *et al.*, 2019] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. Pointweb: Enhancing local neighborhood features for point cloud processing. In *CVPR*, pages 5565–5573, 2019.