Computational Visual Media
https://doi.org/10.1007/s41095-021-0244-6 Vol. 8, No. 2, June 2022, 303–315

Research Article

# Neighborhood co-occurrence modeling in 3D point cloud segmentation

**Jingyu Gong[1], Zhou Ye[2], and Lizhuang Ma[1,3] (✉)**



**Abstract** A significant performance boost has been achieved in point cloud semantic segmentation by utilization of the encoder–decoder architecture and novel convolution operations for point clouds. However, co-occurrence relationships within a local region which can directly influence segmentation results are usually ignored by current works. In this paper, we propose a *neighborhood co-occurrence matrix* (NCM) to model local co-occurrence relationships in a point cloud. We generate target NCM and prediction NCM from semantic labels and a prediction map respectively. Then, Kullback–Leibler (KL) divergence is used to maximize the similarity between the target and prediction NCMs to learn the co-occurrence relationship. Moreover, for large scenes where the NCMs for a sampled point cloud and the whole scene differ greatly, we introduce a reverse form of KL divergence which can better handle the difference to supervise the prediction NCMs. We integrate our method into an existing backbone and conduct comprehensive experiments on three datasets: Semantic3D for outdoor space segmentation, and S3DIS and ScanNet v2 for indoor scene segmentation. Results indicate that our method can significantly improve upon the backbone and outperform many leading competitors.

**Keywords** 3D vision; point cloud; co-occurrence relation modeling; semantic segmentation

1 Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: J. Gong, gongjingyu@sjtu.edu.cn; L. Ma, ma-lz@cs.sjtu.edu.cn (✉).

2 Shanghai CLS Fintech Co., LTD, Shanghai 200030, China. E-mail: yezhou@cls.cn.

3 MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University, Shanghai 200240, China.

Manuscript received: 2021-04-01; accepted: 2021-05-28

## 1 Introduction

With advances in scanning devices, much 3D data has been produced and widely used in augmented and virtual reality, 3D games, and robotics. As a basic form of 3D data, the point cloud is very popular and can be easily converted into meshes or voxels [1]. Semantic segmentation of point clouds is an essential 3D scene comprehension task yet remains challenging due to its inherent irregularity [2].

PointNet [3] was the first neural network to directly process point clouds for 3D segmentation. They proposed to apply shared multi-layer perceptrons (MLPs) to point clouds to learn point-wise features and utilized max/mean pooling to aggregate global features. These were concatenated with point-wise features before a few MLPs were used for final semantic segmentation. Later, PointConv [4] and KPConv [5] used novel 3D convolution operations to extract informative point features and achieved good performance in 3D scene segmentation. An encoder–decoder framework is usually used to gradually extract global features and fuse them with local features to predict the semantic labels. While global contextual information and local region information are both used for point-wise labeling, local co-occurrence relationships are usually ignored or used in a implicit way.

In spite of the rich categories of objects in real world, there is a strong relation between the categories of neighboring objects (i.e., occurrence of specific category pairs). We note that this neighborhood co-occurrence relationship can be used during semantic prediction to rule out neighboring pairs that cannot co-occur in a local region. For example, whiteboards are always adjacent to walls, and chairs are usually close to tables as shown in Fig. 1. These category

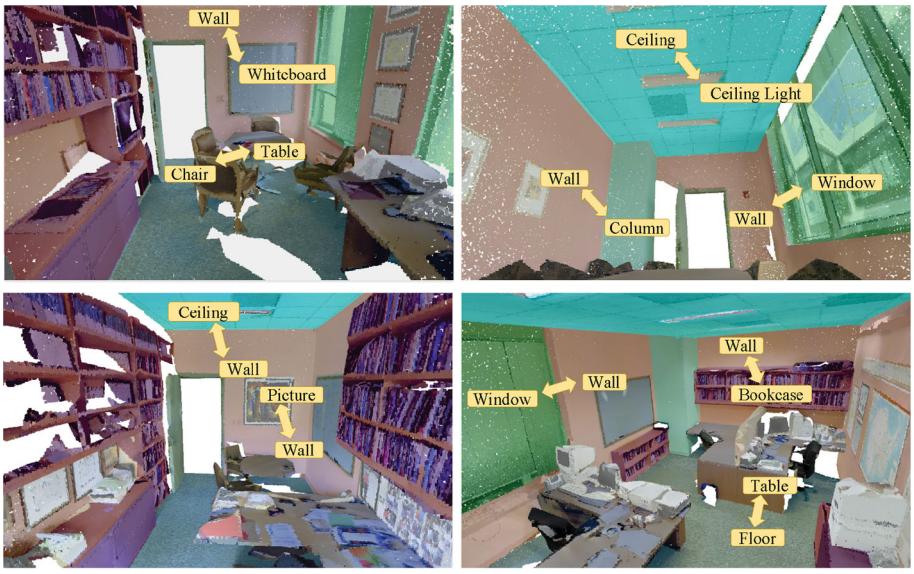


**Fig. 1** Co-occurring category pairs in local regions from scenes in S3DIS Area-5.

pairs of objects can co-occur in a local region of real scene while other pairs (e.g., whiteboards and tables) cannot. However, this idea is usually ignored in point cloud segmentation.

Patch-level co-occurrence relationships have been exploited and used to optimize semantic labeling results, but optimization problems had to be solved in the testing stage [6]. In order to obtain segmentation results which follow real-world co-occurrence relationships without extra computations during inferencing, we propose a neighborhood co-occurrence matrix (NCM) to model this relation. The NCM is a two-dimensional matrix. Each row represents one category of center points, and each column represents one category for the neighbors of center points. Each element $(i, j)$ of the NCM shows the probability that the semantic label of the center point is $i$ and the semantic label of neighboring points is $j$. Based upon this definition, the whole NCM is a joint distribution over categories of center points and neighbors.

In our method, the network can directly learn the local co-occurrence relationship from the neighborhood co-occurrence matrix. In the training stage, we randomly sample points from the original point cloud as center points for simplicity and lack of bias, and collect the category labels of these center points and their $K$ nearest neighbors. Next, these category labels are used to generate the target NCM. Meanwhile, prediction maps of center points and their neighbors will be collected to generate the prediction NCM. To learn the local co-occurrence relationship and make the prediction NCM approximate to target NCM, we introduce Kullback–Leibler (KL) divergence to estimate the distance between these two distributions.

For large-scale scenes, especially outdoor scenes, the neighborhood co-occurrence matrix of a sampled point cloud may not fully reflect the real-world neighbor relationships, and this can be treated as noise in the target NCM. Therefore, we introduce the reverse form of KL divergence into NCM learning to handle the scalar difference between sampled point clouds and whole scenes. In this way, the network can learn to make predictions that are in better accordance with local co-occurrence relationships in real world.

Our major contributions can be summarized as follows:

- a neighborhood co-occurrence matrix to model local co-occurrence of point-wise semantic labels, utilizing KL divergence to minimize difference between the prediction NCM and target NCM;
- introduction of the reverse form of KL divergence into NCM learning to handle the difference between the target NCM generated from sampled data and the real-world NCM for large-scale scenes;
- integration of our method into an existing backbone and experiments on three challenging

benchmark datasets demonstrating significantly improved performance over the backbone for point cloud segmentation task, and outperformance of many state-of-the-art competitors.

## 2 Related work

### 2.1 Semantic segmentation of point clouds

PointNet [3] proposed use of shared multi-layer perceptrons (MLPs) to extract point-wise features and pooling to obtain global features. Then, global features would be duplicated and concatenated with point features before using a few MLPs for semantic segmentation. Later, PointNet++ [7] introduced the encoder–decoder architecture into the point cloud segmentation problem to better fuse local and global information. GACNet [8] utilized graph convolution and paid more attention to neighbors with similar features during feature aggregation. Later, KPConv [5] mapped features of neighboring points onto anchored points, and implemented convolution on those points. RandLA [9] utilized a random sampling strategy which is more efficient for large-scale point cloud segmentation. JSNet [10] and JSIS [11] took instance segmentation and semantic segmentation as joint tasks for improved semantic segmentation. JSENet [12] and BAGEM [13] introduced boundary information into the semantic segmentation task for better contours in the prediction map. Fusion-Aware Conv [14] extracted semantic features both spatially and temporally from RGBD scans for online scene segmentation. On the other side, patch-based methods were proposed to cluster patches with similar features to segment point clouds [15]. However, these methods usually ignored the category-based co-occurrence relationship in point cloud segmentation.

Compared to the methods aforementioned, we propose a neighborhood co-occurrence matrix (NCM) to model the neighborhood co-occurrence relationship. Then, two forms of KL divergence are introduced to minimize the difference between prediction NCM and target NCM, leading to predictions that are more consistent with realistic neighborhood co-occurrence relationships.

### 2.2 Neighborhood context learning

Neighborhood contexts in the vicinity of an object have proved useful for 2D semantic segmentation [16]. RMI [17] utilized region mutual information to model the local relationship between neighboring pixels, and achieved high consistency in the final predictions for image segmentation. Conditional random fields (CRF) were introduced into point cloud segmentation to model the relationships between neighboring labels [18], leading to better segmentation. However, CRF is a post-processing method and extra computations are required during inferencing. In point cloud segmentation, 3P-RNN [19] utilized RNNs to explore long-range spatial context. HPEIN [20] extracted features of edges between neighboring points to implicitly model the neighborhood relation. Region similarity loss was proposed to propagate distinguishing features of center points to neighbors with the same categories in a local neighborhood [2].

Compared to these methods, our method focuses on explicitly learning the neighborhood category co-occurrence relationship for point cloud segmentation.

### 2.3 Co-occurrence modeling

Given the target features, CFNet [21] predicted the probability of co-occurring features and used them as weights to fuse co-occurrent contexts. A global co-occurrence constraint was introduced by Ref. [22] to eliminate configurations that violate common sense or physical law. However, these methods failed to exploit the semantic label co-occurrence relationship in a neighborhood. Segment-based and patch-based contextual relationships were exploited to optimize the label assignment problem for semantic labeling during inferencing [6, 23]. However, extra time is needed then to obtain the segmentation results. Co-occurrence matrices have usually been used to describe the co-occurrence of words in natural language processing [24].

Unlike previous methods, we design a neighborhood co-occurrence matrix to directly model the local category co-occurrence relationship to eliminate impossible neighboring pairs in point cloud segmentation. Additionally, our method can train the network in an end-to-end manner, and it does not require extra time during inferencing.

## 3 Method

In this section, we first introduce the overall architecture of our method in Section 3.1. Then, we describe the proposed neighborhood co-occurrence

matrix (NCM) used to model local co-occurrence relationships in Section 3.2. Finally, we describe how the target NCM supervises the output prediction and makes the network learn local co-occurrence relationships as well as the reverse form of KL divergence in Section 3.3.

### 3.1 Overview

Figure 2 shows the overall framework of our method. First of all, we use a common encoder–decoder network to extract features and make category predictions. Then, ground truth semantic labels are directly used to supervise the segmentation results through cross entropy loss. Meanwhile, we generate the prediction neighborhood co-occurrence matrix (prediction NCM) from the prediction maps and generate the target NCM from point-wise semantic labels. Later, KL divergence is used to supervise the prediction NCM and make it approximate the target NCM. In this way, anomalous co-occurring neighboring pairs will be punished and the co-occurrence relationships in our prediction will be more reasonable.

### 3.2 Neighborhood co-occurrence matrix

Co-occurrence relationships in the local neighborhood can directly influence the results of point cloud semantic segmentation. For instance, a whiteboard usually co-occurs with a wall in a local region and is unlikely to be adjacent to other categories such as the ceiling or floor (see Fig. 1). Based upon this observation, we hope our final semantic prediction to accord with real-world neighborhood co-occurrence relationships.

While co-occurrence relationships are seldom explicitly exploited in segmentation tasks, they are usually modeled by a co-occurrence matrix in natural language processing to find co-occurring words within a sentence. Inspired by the design of co-occurrence matrices for words, we propose a neighborhood co-occurrence matrix to model the relationship of neighboring co-occurring categories in local regions of point clouds. Here, we attempt to exploit the category relationship between a randomly selected point (referred to as the center point) and its neighbors. For a semantic segmentation task where we need to categorize each point as one of $C$ classes, our designed NCM will be a $C \times C$ matrix. Each row of NCM represents a category for center points and each column represents a category of their neighboring points. Specifically, the $ij$-th element indicates the probability that the center point belongs to the $i$-th class and a neighboring point belongs to the $j$-th class.

In order to effectively utilize computational resources and storage, we only sample a fixed ratio of center points from the original point clouds, naming $N' = \alpha N$ as shown in Fig. 2. To generate the target NCM, we first collect the one-hot labels of

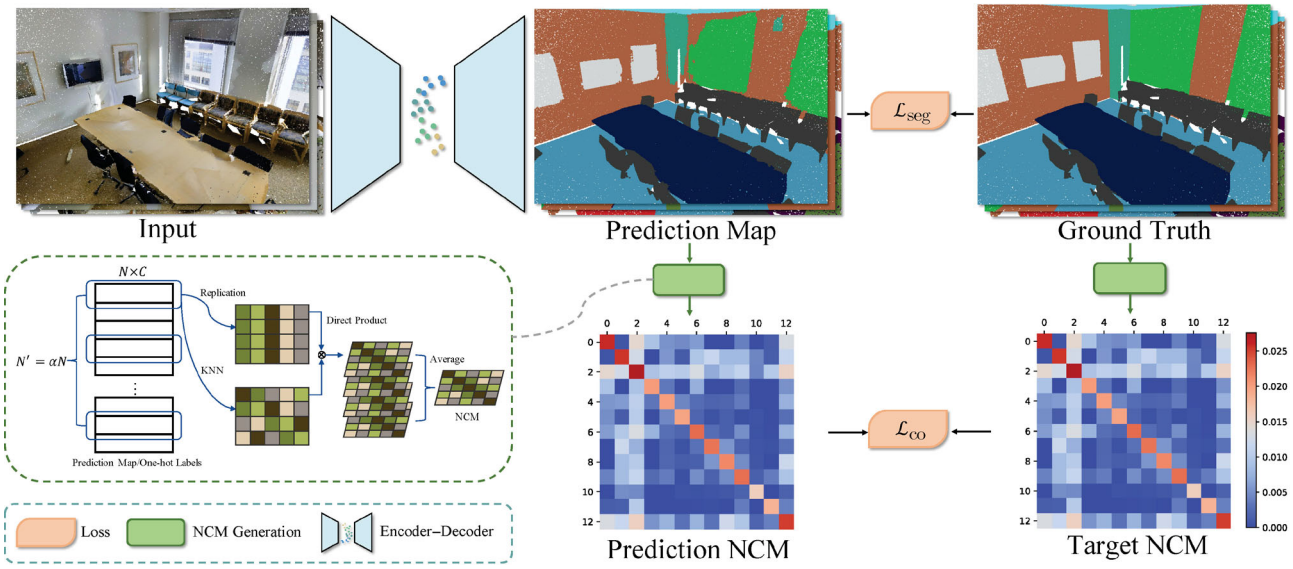


**Fig. 2** Architecture of our proposed method. An encoder–decoder network is used to produce the prediction map. Then, prediction NCM and target NCM are generated from the prediction map and ground truth respectively. The KL divergence of these two distributions is minimized to learn local co-occurrence relationships in the real world.

these center points, denoted $A_c \in \mathcal{R}^{N' \times C}$. Then, for each center point, we search for its $K$ nearest neighbors and collect their corresponding one-hot labels. The collected neighbors' one-hot labels are denoted $A_n \in \mathcal{R}^{N' \times K \times C}$. Then, the target NCM $M \in \mathcal{R}^{C \times C}$ is given by

$$M[i,j] = \frac{1}{N'K} \sum_{n}^{N'} \sum_{k}^{K} A_c[n,i] A_n[n,k,j] \quad (1)$$

Note that $\sum_m^C \sum_n^C M[m,n] = 1$, so $M$ is a normalized probability density function which models the target co-occurrence relationship in local regions of real 3D scenes.

### 3.3 Learning neighborhood co-occurrence relationship

In order to learn the real-world neighborhood co-occurrence relationship, we directly utilize the target NCM $M$ to supervise the prediction NCM generated from the output prediction maps. Our prediction NCM will approach the target NCM to learn a more reasonable co-occurrence relationship in the local region.

Unlike generating the target NCM, we directly utilize the prediction map of these $N'$ center points $\hat{A}_c \in \mathcal{R}^{N' \times C}$ where $\hat{A}_c[n,i]$ represents the predicted probability that the $n$-th center point belongs to the $i$-th category. As in the counterpart of target NCM, we also aggregate the prediction maps of center points' neighbors $\hat{A}_n \in \mathcal{R}^{N' \times K \times C}$ where $K$ is the number of neighbors. The prediction NCM can be calculated by the following formula:

$$\hat{M}[i,j] = \frac{1}{N'K} \sum_{n}^{N'} \sum_{k}^{K} \hat{A}_c[n,i] \hat{A}_n[n,k,j] \quad (2)$$

We again have that $\sum_i^C \sum_j^C \hat{M}[i,j] = 1$ because

$$\sum_{i}^{C} \hat{A}_c[n,i] = 1 \quad \text{and} \quad \sum_{j}^{C} \hat{A}_n[n,k,j] = 1 \quad (3)$$

according to the definition of probability distribution. In order to make the prediction NCM approach the target NCM, KL divergence, which measures the distance between two probability distributions, is introduced to narrow the difference between target NCM and prediction NCM.

A high KL divergence indicates a large difference between two distributions. A common way of formulating KL divergence is

$$D(p||q) = \int_x p(x) \log\left(\frac{p(x)}{q(x)}\right) \quad (4)$$

where $p$ and $q$ are two distributions over variable $x$. In our method, integration over continuous $x$ is replaced by summation over the discrete pairs $(i,j)$. According to the choice of $p$ and $q$, we have two forms of KL divergence loss.

In the common form of KL divergence, we can simply set $M$ to be $p$ and $\hat{M}$ to be $q$. Based upon this, we can reformulate the KL divergence of these two NCM distributions as

$$D(M||\hat{M}) = \sum_{i}^{C} \sum_{j}^{C} (M[i,j] \log M[i,j] - M[i,j] \log \hat{M}[i,j]) \quad (5)$$

Note that $M[i,j]\log M[i,j]$ is not differentiable with respect to hidden parameters in the network because the category labels of points are fixed. Thus we only need to optimize the second term to minimize the KL divergence. Thus, our loss for NCM is

$$\mathcal{L}_{co} = -\sum_{i}^{C} \sum_{j}^{C} M[i,j] \log(\hat{M}[i,j] + \epsilon) \quad (6)$$

where $\epsilon$ is a small quantity to prevent invalid numerical operations.

In order to handle the difference between the target NCM generated from the sampled point cloud and the real-world NCM, especially for large-scale scenes, we treat this difference as a kind of noise and introduce the reverse form of KL divergence which conversely sets $\hat{M}$ to be $p$ and $M$ to be $q$; we call this the reverse KL divergence. Then, the loss for the neighborhood co-occurrence matrix can be reformulated as

$$\mathcal{L}_{co} = \sum_{i}^{C} \sum_{j}^{C} (\hat{M}[i,j] \log(\hat{M}[i,j] + \epsilon) - \hat{M}[i,j] \log(M[i,j] + \epsilon)) \quad (7)$$

where both terms are differentiable with respect to hidden parameters in this case.

Compared to Eq. (6), this form of KL divergence is more complex because both terms are differentiable. The first term maximizes the entropy of the prediction NCM which can be treated as a constraint to give a prior on a uniform distribution. The second term is the reverse cross entropy which has been shown to be more tolerant to noise in labels [25]. The differences in performance between these two forms of loss and analysis of KL divergence will be discussed in detail in Section 4.3.

The total loss for the network consists of two parts: $\mathcal{L}_{seg}$ and $\mathcal{L}_{co}$:

$$\mathcal{L} = \mathcal{L}_{seg} + \lambda \mathcal{L}_{co} \quad (8)$$

where $\mathcal{L}_{\text{seg}}$ represents the cross entropy loss for point-wise segmentation.

In our implementation, $\alpha$ is set to 0.3, and 8 nearest neighbors are collected to generate the target NCM and prediction NCM. $\epsilon$ and $\lambda$ are set to $10^{-8}$ and 1 respectively, giving good results.

## 4 Experiments

Our experiments consist of five parts. First, we test our method on the large-scale outdoor semantic segmentation task Semantic3D reduced-8 [26] in Section 4.1. Then, we evaluate the performance of our method on the indoor scene semantic segmentation benchmarks S3DIS [27] and ScanNet v2 [28] in Section 4.2. Next, we conduct studies to analyze the two forms of KL divergence in Section 4.3. Later, we analyse the influence of number of neighbor and sampling density on segmentation performance in Section 4.4. Finally, we visualize the prediction NCMs and target NCMs for some real scenes in Section 4.5.

### 4.1 Large-scale outdoor space semantic segmentation

#### *4.1.1 Dataset*

We evaluate the effectiveness of our method for outdoor space semantic segmentation on the Semantic3D task [26]. This dataset contains 15 large-scale outdoor areas for training and another 15 areas for testing. There are more than 4 billion points in this dataset and all points can be divided into 8 categories. For easier evaluation, Semantic3D proposed another segmentation task with fewer points in the test set: Semantic3D reduced-8. Only labels for training data are available, and predictions on the test set must be submitted to their online servers for evaluation.

#### *4.1.2 Implementation*

We utilize KPConv *deform* [5] as our backbone and embed our method into it. In the training stage, we randomly sample spheres of 3 m in radius from the outdoor scenes and feed them into the network for training following Refs. [5, 29]. Eqs. (1) and (2) are used to generate the target NCM and prediction NCM. Eq. (8) is used to optimize the whole network, and Eq. (7) is used as the KL divergence loss for NCM. In the test stage, we utilize the trained model to predict the results on all points in the test set and submit the results to the Semantic3D server [26] for evaluation. The momentum optimizer is utilized to train the network, and the batch size is set to 10 on a single GTX 1080Ti GPU.

#### *4.1.3 Results*

We report the mean IoU (mIoU) over categories and IoUs for different categories on Semantic3D reduced-8 task in Table 1. Our method achieves 76.6% mIoU in this task, outperforming many existing methods. Our method also brings a satisfying 3.5% mIoU improvement over the backbone which shows its effectiveness. We also provide the category-wise IoUs, but IoUs for KPConv *deform* are not listed because they are not available in their paper and the benchmark. Figure 3 visualizes outdoor segmentation results of KPConv *deform* and our method on the validation set of Semantic3D reduced-8 split by KPConv *deform* [5]. The red dashed-dotted circles indicate obvious qualitative improvements, more clearly seen in the close-up views.

### 4.2 Indoor scene semantic segmentation

#### *4.2.1 Dataset*

We evaluate the performance of our method for indoor semantic segmentation on the S3DIS [27] and

**Table 1** Results of outdoor space semantic segmentation on Semantic3D (reduced-8)

| Method | mIoU (%) | man-made. | natural. | high veg. | low veg. | buildings | hard scape | scanning. | cars |
|---|---|---|---|---|---|---|---|---|---|
| SegCloud ('17) [30] | 61.3 | 83.9 | 66.0 | 86.0 | 40.5 | 91.1 | 30.9 | 27.5 | 64.3 |
| RF_MSSF ('18) [31] | 62.7 | 87.6 | 80.3 | 81.8 | 36.4 | 92.2 | 24.1 | 42.6 | 56.6 |
| SPG ('18) [32] | 73.2 | **97.4** | **92.6** | 87.9 | 44.0 | 93.2 | 31.0 | 63.5 | 76.2 |
| ShellNet ('19) [33] | 69.4 | 96.3 | 90.4 | 83.9 | 41.0 | 94.2 | 34.7 | 43.9 | 70.2 |
| GACNet ('19) [8] | 70.8 | 86.4 | 77.7 | **88.5** | 60.6 | 94.2 | 37.3 | 43.5 | 77.8 |
| KPConv *deform* ('19) [5] | 73.1 | — | — | — | — | — | — | — | — |
| FGCN ('20) [34] | 62.4 | 90.3 | 65.2 | 86.2 | 38.7 | 90.1 | 31.6 | 28.8 | 68.2 |
| PointGCR ('20) [35] | 69.5 | 93.8 | 80.0 | 64.4 | **66.4** | 93.2 | 39.2 | 34.3 | **85.3** |
| Ours | **76.6** | 95.8 | 91.7 | 82.6 | 50.6 | **94.8** | **40.0** | **75.4** | 81.7 |

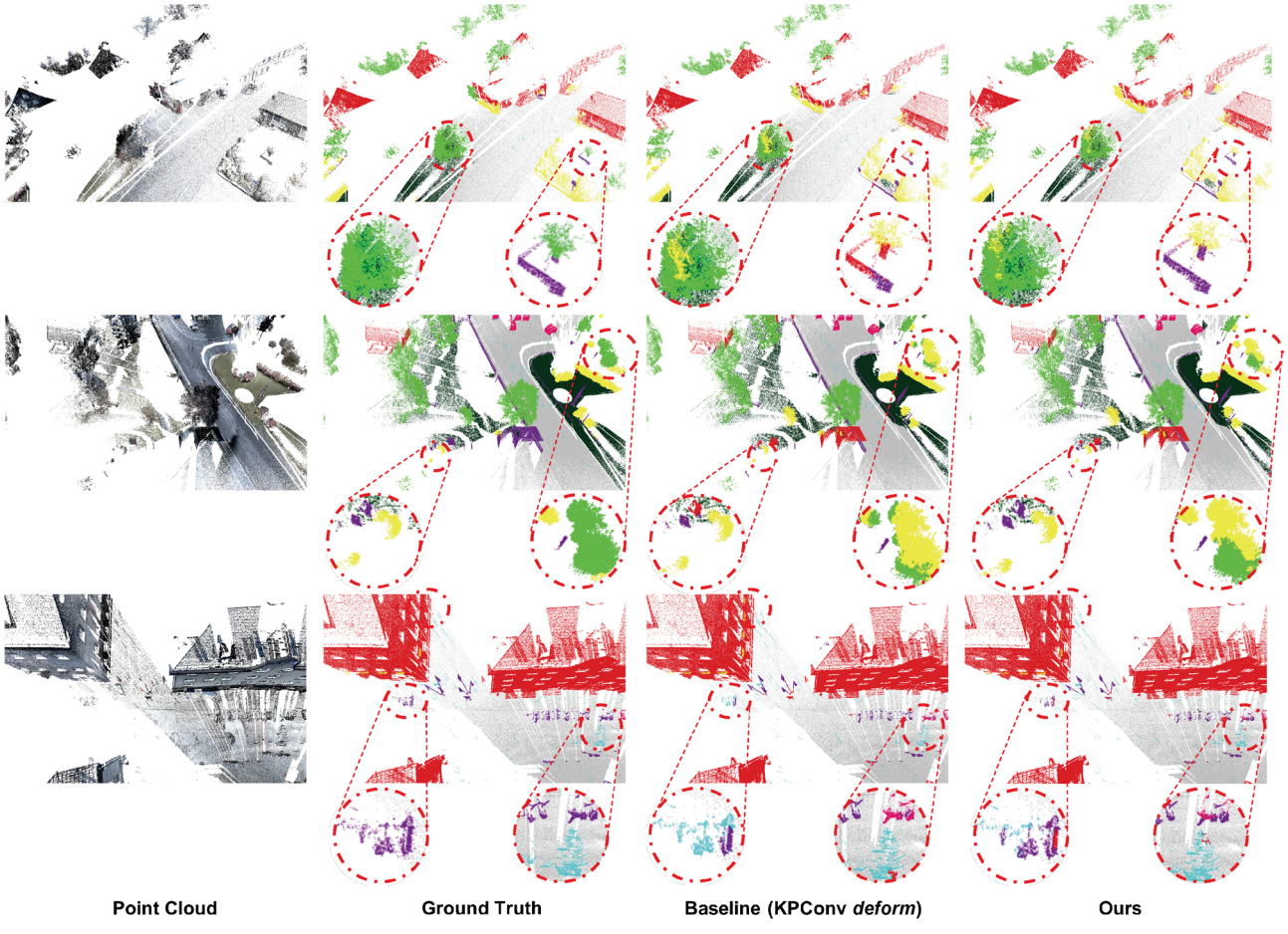


**Fig. 3** Visual results for semantic segmentation on the validation set of Semantic3D.

ScanNet v2 [28] datasets.

S3DIS contains 271 rooms in 6 large indoor areas from three different buildings. About 273 million points are collected and annotated in this dataset; all points are categorized into 13 classes. Following previous work [5, 13, 36], we take rooms in Area-5 as the test set and samples from the other areas as the training set.

ScanNet v2 contains 1513 cluttered indoor scenes with annotations. 1201 scenes are used for training and 312 scenes are used for validation. All annotated points are categorized into 20 classes or unlabeled. Additionally, another 100 scenes are published without label annotations as the test set.

#### *4.2.2 Implementation*

We apply our method to KPConv *deform* [5] and take it as our baseline. Following KPConv *deform*, we randomly sample spheres with a 2 m radius from rooms in the training set and feed them into the network for training. Again, Eqs. (1) and (2) are used to generate the target NCM and prediction NCM. Eq. (8) is used to optimize the whole network. However, Eq. (6) is used as the KL divergence loss for NCM. In the testing stage, spheres are sampled regularly and all points are included in at least one sphere. The network is trained by a momentum optimizer, with batch size 5 for S3DIS and 10 for ScanNet v2, using a single GTX 1080Ti GPU.

#### *4.2.3 Results*

We report the results of our method and many state-of-the-art competitors on S3DIS Area-5 in Table 2; mean IoU (mIoU) is taken as a metric to evaluate segmentation performance. Our method achieves a 68.29% mIoU (1.19% higher than the backbone) and outperforms many existing methods. We also list the IoU scores for different categories in this table. No methods perform well in the beam category because beams in S3DIS Area-5 are tilted while beams in other areas are horizontal. Our method improves IoU for most categories except the sofa class. This

**Table 2** Indoor semantic segmentation results on S3DIS Area-5

| Method | mIoU (%) | ceil. | floor | wall | beam | col. | wind. | door | chair | table | book. | sofa | board | clut. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PointNet ('17) [3] | 41.09 | 88.80 | 97.33 | 69.80 | **0.05** | 3.92 | 46.26 | 10.76 | 58.93 | 52.61 | 5.85 | 40.28 | 26.38 | 33.22 |
| RSNet ('18) [37] | 51.93 | 93.34 | 98.36 | 79.18 | 0.00 | 15.75 | 45.37 | 50.10 | 65.52 | 67.87 | 22.45 | 52.45 | 41.02 | 43.64 |
| KPConv *deform* ('19) [5] | 67.1 | 92.8 | 97.3 | 82.4 | 0.0 | 23.9 | 58.0 | 69.0 | 91.0 | 81.5 | 75.3 | **75.4** | 66.7 | 58.9 |
| FPConv ('20) [38] | 62.8 | **94.6** | 98.5 | 80.9 | 0.0 | 19.1 | 60.1 | 48.9 | 80.6 | 88.0 | 53.2 | 68.4 | 68.2 | 54.9 |
| Point2Node ('20) [39] | 62.96 | 93.88 | 98.26 | 83.30 | 0.00 | **35.65** | 55.31 | 58.78 | 79.51 | 84.67 | 44.07 | 71.13 | 58.72 | 55.17 |
| SegGCN ('20) [36] | 63.6 | 93.7 | **98.6** | 80.6 | 0.0 | 28.5 | 42.6 | **74.5** | 80.9 | **88.7** | 69.0 | 71.3 | 44.4 | 54.3 |
| DCM-Net ('20) [40] | 64.0 | 92.1 | 96.8 | 78.6 | 0.0 | 21.6 | 61.7 | 54.6 | 78.9 | **88.7** | 68.1 | 72.3 | 66.5 | 52.4 |
| FusionNet ('20) [41] | 67.2 | — | — | — | — | — | — | — | — | — | — | — | — | — |
| JSENet ('20) [12] | 67.7 | 93.8 | 97.0 | 83.0 | 0.0 | 23.2 | 61.3 | 71.6 | 89.9 | 79.8 | 75.6 | 72.3 | 72.7 | 60.4 |
| Ours | **68.29** | 94.43 | 98.25 | **83.60** | 0.00 | 25.74 | **62.01** | 70.32 | **91.51** | 82.58 | **75.98** | 73.04 | **69.51** | **60.81** |

is because NCM learning punishes those pairs that seldom appear, thus making the network less likely to categorize a point into a minor class. Thus, we do not observe score improvement for the sofa category which has least points. Additionally, we visualize the improvement over our backbone (KPConv *deform*) in Fig. 4, with yellow dashed-dotted circles indicating obvious improvements.

For ScanNet v2, we report the results in Table 3, and mean IoU over category is also used to estimate the performance. In this dataset, our method achieves 69.0% mIoU which is 0.6% higher than our backbone, and our method achieves a state-of-the-art performance in this benchmark. Category-wise scores are also shown in this table. We can also see that our method improves the performance for most categories, but degrades the performance for categories with few points like sofa and refrigerator. The reason is the same to S3DIS that NCM learning punishes pairs that appear less frequently, thus degrading the performance of minor categories.

### 4.3 Choice of KL divergence

In this section, we conduct experiments to compare the difference in performance between the usual and reverse forms of KL divergence for both indoor scene and outdoor area semantic segmentation.

As shown in Section 3.3, there are two forms of KL divergence for NCM learning. Although both forms lead the prediction NCM to approximate the target

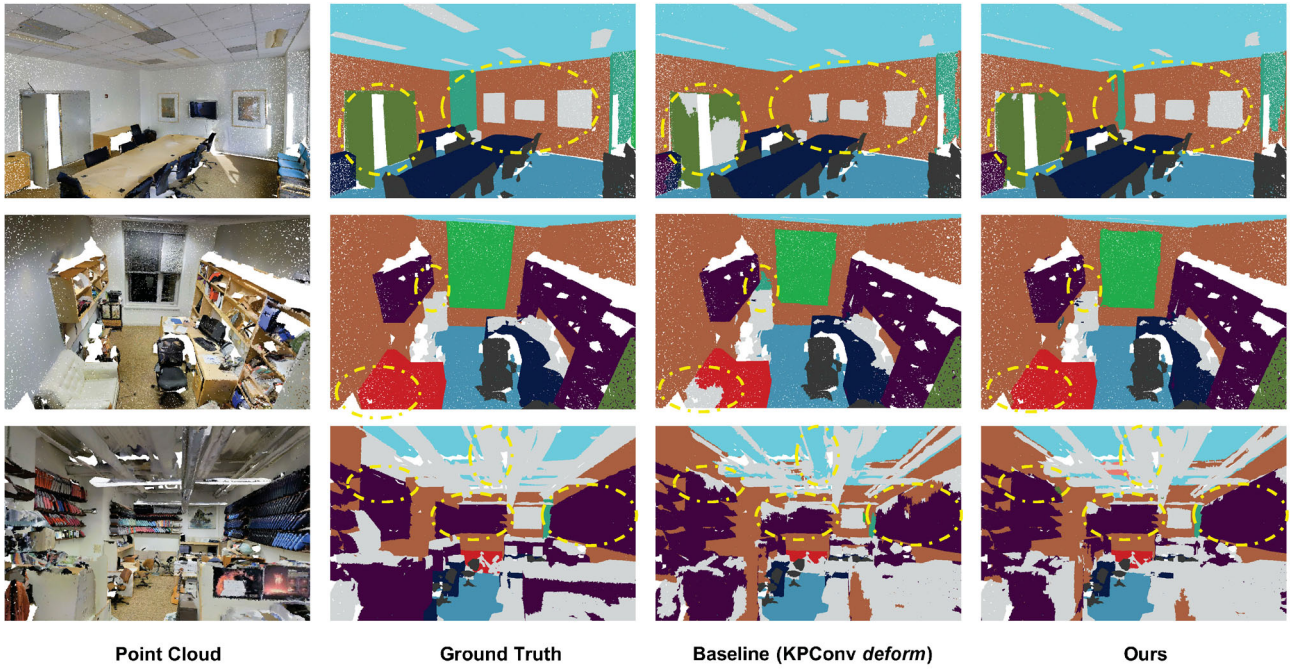


**Fig. 4** Visual results for semantic segmentation on S3DIS Area-5.

**Table 3** Semantic segmentation results on ScanNet v2

| Method | mIoU (%) | bth. | bed | bksf. | cab. | chair | ctr. | curt. | desk | door | floor | oth. | pic. | ref. | shw. | sink | sofa | tab. | toil. | wall | win. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PointNet++ ('17) [3] | 33.9 | 58.4 | 47.8 | 45.8 | 25.6 | 36.0 | 25.0 | 24.7 | 27.8 | 26.1 | 67.7 | 18.3 | 11.7 | 21.2 | 14.5 | 36.4 | 34.6 | 23.2 | 54.8 | 52.3 | 25.2 |
| PointCNN ('18) [42] | 45.8 | 57.7 | 61.1 | 35.6 | 32.1 | 71.5 | 29.9 | 37.6 | 32.8 | 31.9 | 94.4 | 28.5 | 16.4 | 21.6 | 22.9 | 48.4 | 54.5 | 45.6 | 75.5 | 70.9 | 47.5 |
| TextureNet ('19) [43] | 56.6 | 67.2 | 66.4 | 67.1 | 49.4 | 71.9 | 44.5 | 67.8 | 41.1 | 39.6 | 93.5 | 35.6 | 22.5 | 41.2 | 53.5 | 56.5 | 63.6 | 46.4 | 79.4 | 68.0 | 56.8 |
| HPEIN ('19) [20] | 61.8 | 72.9 | 66.8 | 64.7 | 59.7 | 76.6 | 41.4 | 68.0 | 52.0 | 52.5 | 94.6 | 43.2 | 21.5 | 49.3 | 59.9 | 63.8 | 61.7 | 57.0 | **89.7** | 80.6 | 60.5 |
| KPConv *deform* ('19) [5] | 68.4 | 84.7 | 75.8 | 78.4 | 64.7 | 81.4 | 47.3 | 77.2 | 60.5 | 59.4 | 93.5 | 45.0 | 18.1 | 58.7 | 80.5 | 69.0 | **78.5** | 61.4 | 88.2 | 81.9 | 63.2 |
| SPH3D-GCN ('20) [44] | 61.0 | **85.8** | 77.2 | 48.9 | 53.2 | 79.2 | 40.4 | 64.3 | 57.0 | 50.7 | 93.5 | 41.4 | 4.6 | 51.0 | 70.2 | 60.2 | 70.5 | 54.9 | 85.9 | 77.3 | 53.4 |
| FAConv ('20) [14] | 63.0 | 60.4 | 74.1 | 76.6 | 59.0 | 74.7 | **50.1** | 73.4 | 50.3 | 52.7 | 91.9 | 45.4 | 32.3 | 55.0 | 42.0 | 67.8 | 68.8 | 54.4 | 89.6 | 79.5 | 62.7 |
| FPConv ('20) [38] | 63.9 | 78.5 | 76.0 | 71.3 | 60.3 | 79.8 | 39.2 | 53.4 | 60.3 | 52.4 | 94.8 | 45.7 | 25.0 | 53.8 | 72.3 | 59.8 | 69.6 | 61.4 | 87.2 | 79.9 | 56.7 |
| DCM-Net ('20) [40] | 65.8 | 77.8 | 70.2 | **80.6** | 61.9 | 81.3 | 46.8 | 69.3 | 49.4 | 52.4 | 94.1 | 44.9 | 29.8 | 51.0 | 82.1 | 67.5 | 72.7 | 56.8 | 82.6 | 80.3 | 63.7 |
| PointASNL ('20) [45] | 66.6 | 70.3 | **78.1** | 75.1 | 65.5 | **83.0** | 47.1 | 76.9 | 47.4 | 53.7 | **95.1** | 47.5 | 27.9 | **63.5** | 69.8 | 67.5 | 75.1 | 55.3 | 81.6 | 80.6 | 70.3 |
| FusionNet ('20) [41] | 68.8 | 70.4 | 74.1 | 75.4 | 65.6 | 82.9 | **50.1** | 74.1 | 60.9 | 54.8 | 95.0 | **52.2** | **37.1** | 63.3 | 75.6 | **71.5** | 77.1 | **62.3** | 86.1 | 81.4 | **65.8** |
| Ours | **69.0** | 85.3 | 75.5 | 78.9 | **66.0** | 80.5 | 47.1 | **79.3** | **61.8** | **59.6** | 94.7 | 47.9 | 22.5 | 56.3 | **84.5** | 68.2 | 74.9 | 61.3 | 89.6 | **82.7** | 63.9 |

NCM, the gradients are quite different. Thus, we conduct a study to analyze the difference between these two forms of KL divergence on Semantic3D reduced-8 and S3DIS Area-5 tasks. The results are reported in Table 4. It demonstrates that these two forms of KL divergence achieve similar improvements on the S3DIS Area-5 task. However, reverse KL divergence achieves a 0.8% higher mIoU on the Semantic3D reduced-8 task. This results from differences between the target NCM generated from the sampled data and the real-world NCM. Specifically, the scale of the whole scene is much greater than that of the sampled point cloud in the Semantic3D dataset. The second term in Eq. (7), $-\hat{M}[i,j]\log(M[i,j]+\epsilon)$, is the reverse cross entropy which is more tolerant to such discrepancy noise [25], thus providing better performance. Furthermore, the first term $\hat{M}[i,j]\log(\hat{M}[i,j]+\epsilon)$ is the negative entropy, and minimizing it will maximize the entropy of the prediction NCM. This will give a preference to a uniform distribution over the prediction NCM, thus alleviating imbalance between the number of points in each category during NCM learning.

## 4.4 Hyper-parameter analysis

Here, we first conduct experiments to analyse the influence of number of neighbor used for NCM generation. Then, we study how the sampling density impacts segmentation performance, using the Semantic3D reduced-8 task and reverse KL divergence.

**Table 4** Comparison of forms of KL divergence used as loss function

| Method | Semantic3D | S3DIS |
|---|---|---|
| Ours (using Eq. (6)) | 75.8 | 68.29 |
| Ours (using Eq. (7)) | 76.6 | 68.22 |

### *4.4.1 Number of neighbors in NCM*

In this section, we conduct a study on changing the number of neighbors in the neighborhood co-occurrence matrix (NCM), which affects the size of neighborhood. We set the number of neighbors to 4, 8, and 12, respectively, with all other settings remaining unchanged from our original method. The experimental results are reported in Table 5 which shows the neighborhood consisting of 8 nearest neighbors brings the largest improvement to point cloud segmentation.

### *4.4.2 Sampling density in NCM*

We also conduct experiments to study the influence of sampling density on segmentation performance. We attempt to control the sampling density by changing the hyper-parameter $\alpha$. We set $\alpha$ to 30%, 10%, and 3% in turn, with all other settings remaining the same. Results are reported in Table 6. A higher sampling density results in better segmentation results because more samples in the NCM lead to more stable co-occurrence relationship learning.

## 4.5 Visualization of NCM

In order to reflect the improvement of segmentation performance in NCM, we visualize target NCMs and prediction NCMs of our baseline and our method for some scenes of S3DIS in Fig. 5. It shows that

**Table 5** Effect of number of neighbors used to generate NCM

| Neighbors | 4 | 8 | 12 |
|---|---|---|---|
| mIoU (%) | 75.8 | 76.6 | 75.9 |

**Table 6** Effect of varying sampling density $\alpha$

| $\alpha$ | 30% | 10% | 3% |
|---|---|---|---|
| mIoU (%) | 76.6 | 75.7 | 75.4 |

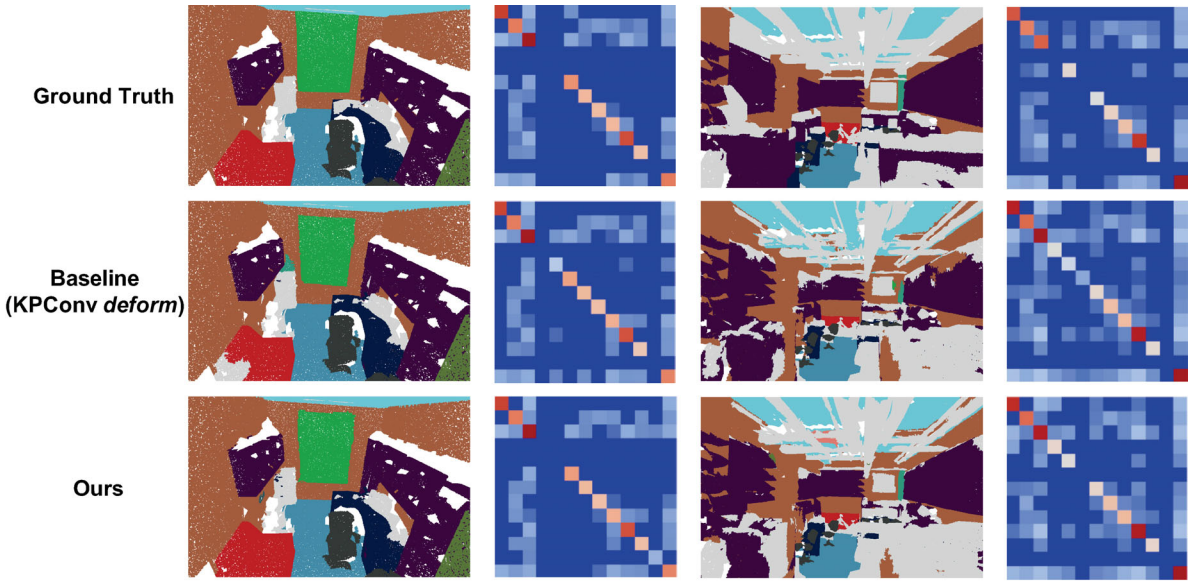


**Fig. 5** Visualization results of NCM for scenes in S3DIS.

our method removes many impossible pairs in the scenes as reflected by the NCM and improves the segmentation performance. For instance, there are no column–clutter pairs in the scenes and this is reflected in the NCM where segmentation improvement is achieved.

## 5 Conclusions

In this paper, we propose a neighborhood co-occurrence matrix to model the local category co-occurrence relationship and introduce it into the point cloud segmentation task. KL divergence is used to maximize the similarity of target NCM and prediction NCM. For better learning of local co-occurrence relationship for large-scale areas, we introduce the reverse form of KL divergence to NCM learning which is more robust to the difference between the NCM of a sampled point cloud and that of a whole scene. Additionally, our proposed method achieves state-of-the-art performance on Semantic3D for outdoor space segmentation as well as S3DIS and ScanNet v2 for indoor scene segmentation. Finally, we compare and analyze the difference in performance between the two forms of KL divergence used in our method, and conduct experiments to analyse the influence of number of neighbors and sampling density in NCM generation.

## Acknowledgements

All datasets on which the conclusions of the manuscript depend come from publicly available repositories: `http://semantic3d.net/`, `http://buildingparser.stanford.edu/dataset.html`, and `http://www.scan-net.org`.

We thank the support of the National Natural Science Foundation of China (61972157), the Natural Science Foundation of Shanghai (20ZR1417700), the National Key R&D Program of China (2019YFC1521104, 2020AAA0108301), Shanghai Municipal Commission of Economy and Information (XX-RGZN-01-19-6348), and the Art Major Project of National Social Science Fund (I8ZD22).

We thank Yao Li for server management.

## References

[1] Verdoja, F.; Thomas, D.; Sugimoto, A. Fast 3D point cloud segmentation using supervoxels with geometry and color for 3D scene understanding. In: Proceedings of the IEEE International Conference on Multimedia and Expo, 1285–1290, 2017.

[2] Xu, J. C.; Gong, J. Y.; Zhou, J.; Tan, X.; Xie, Y.; Ma, L. Z. SceneEncoder: Scene-aware semantic segmentation of point clouds with a learnable scene descriptor. In: Proceedings of the 29th International Joint Conference on Artificial Intelligence, 601–607, 2020.

[3] Charles, R. Q.; Hao, S.; Mo, K. C.; Guibas, L. J. PointNet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 77–85, 2017.

[4] Wu, W. X.; Qi, Z.; Fuxin, L. PointConv: Deep convo-

lutional networks on 3D point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 9613–9622, 2019.

[5] Thomas, H.; Qi, C. R.; Deschaud, J. E.; Marcotegui, B.; Goulette, F.; Guibas, L. KPConv: Flexible and deformable convolution for point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 6410–6419, 2019.

[6] Hu, S.-M.; Cai, J.-X.; Lai, Y.-K. Semantic labeling and instance segmentation of 3D point clouds using patch context analysis and multiscale processing. *IEEE Transactions on Visualization and Computer Graphics* Vol. 26, No. 7, 2485–2498, 2020.

[7] Qi, C. R.; Yi, L.; Su, H.; Guibas, L. J. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, 5105–5114, 2017.

[8] Wang, L.; Huang, Y. C.; Hou, Y. L.; Zhang, S. M.; Shan, J. Graph attention convolution for point cloud semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 10288–10297, 2019.

[9] Hu, Q. Y.; Yang, B.; Xie, L. H.; Rosa, S.; Guo, Y. L.; Wang, Z. H.; Trigoni, N.; Markham, A. RandLA-net: Efficient semantic segmentation of large-scale point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11105–11114, 2020.

[10] Zhao, L.; Tao, W. B. JSNet: Joint instance and semantic segmentation of 3D point clouds. *Proceedings of the AAAI Conference on Artificial Intelligence* Vol. 34, No. 7, 12951–12958, 2020.

[11] Pham, Q. H.; Nguyen, T.; Hua, B. S.; Roig, G.; Yeung, S. K. JSIS3D: Joint semantic-instance segmentation of 3D point clouds with multi-task pointwise networks and multi-value conditional random fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8819–8828, 2019.

[12] Hu, Z.; Zhen, M.; Bai, X.; Fu, H.; Tai, C. JSENet: Joint semantic segmentation and edge detection network for 3D point clouds. In: *Computer Vision–ECCV 2020. Lecture Notes in Computer Science, Vol. 12365*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 222–239, 2020.

[13] Gong, J.; Xu, J.; Tan, X.; Zhou, J.; Qu, Y.; Xie, Y.; Ma, L. Boundary-aware geometric encoding for semantic segmentation of point clouds. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2021.

[14] Zhang, J. Z.; Zhu, C. Y.; Zheng, L. T.; Xu, K. Fusion-aware point convolution for online semantic 3D scene segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4533–4542, 2020.

[15] Mattausch, O.; Panozzo, D.; Mura, C.; Sorkine-Hornung, O.; Pajarola, R. Object detection and classification from large-scale cluttered indoor scans. *Computer Graphics Forum* Vol. 33, No. 2, 11–21, 2014.

[16] Mottaghi, R.; Chen, X. J.; Liu, X. B.; Cho, N. G.; Lee, S. W.; Fidler, S.; Urtasun, R.; Yuille, A. The role of context for object detection and semantic segmentation in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 891–898, 2014.

[17] Zhao, S.; Wang, Y.; Yang, Z.; Cai, D. Region mutual information loss for semantic segmentation. In: Proceedings of the 33rd Conference on Neural Information Processing Systems, 11117–11127, 2019.

[18] Wu, B. C.; Wan, A.; Yue, X. Y.; Keutzer, K. SqueezeSeg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud. In: Proceedings of the IEEE International Conference on Robotics and Automation, 1887–1893, 2018.

[19] Ye, X. Q.; Li, J. M.; Huang, H. X.; Du, L.; Zhang, X. L. 3D recurrent neural networks with context fusion for point cloud semantic segmentation. In: *Computer Vision–ECCV 2018. Lecture Notes in Computer Science, Vol. 11211*. Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 415–430, 2018.

[20] Jiang, L.; Zhao, H. S.; Liu, S.; Shen, X. Y.; Fu, C. W.; Jia, J. Y. Hierarchical point-edge interaction network for point cloud semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 10432–10440, 2019.

[21] Zhang, H.; Zhang, H.; Wang, C. G.; Xie, J. Y. Co-occurrent features in semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 548–557, 2019.

[22] Deng, Z.; Todorovic, S.; Latecki, L. J. Semantic segmentation of RGBD images with mutex constraints. In: Proceedings of the IEEE International Conference on Computer Vision, 1733–1741, 2015.

[23] Koppula, H. S.; Anand, A.; Joachims, T.; Saxena, A. Semantic labeling of 3D point clouds for indoor scenes. In: Proceedings of the 24th International Conference on Neural Information Processing Systems, 244–252, 2011.

[24] Zhao, Z.; Liu, T.; Li, S.; Li, B. F.; Du, X. Y. Ngram2vec: Learning improved word representations from ngram co-occurrence statistics. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 244–253, 2017.

[25] Wang, Y. S.; Ma, X. J.; Chen, Z. Y.; Luo, Y.; Yi, J. F.; Bailey, J. Symmetric cross entropy for robust learning with noisy labels. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 322–330, 2019.

[26] Hackel, T.; Savinov, N.; Ladicky, L.; Wegner, J. D.; Schindler, K.; Pollefeys, M. Semantic3D.net: A new large-scale point cloud classification benchmark. *arXiv preprint* arXiv:1704.03847, 2017.

[27] Armeni, I.; Sener, O.; Zamir, A. R.; Jiang, H.; Brilakis, I.; Fischer, M.; Savarese, S. 3D semantic parsing of large-scale indoor spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1534–1543, 2016.

[28] Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; Nießner, M. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2432–2443, 2017.

[29] Gong, J.; Xu, J.; Tan, X.; Song, H.; Qu, Y.; Xie, Y.; Ma, L. Omni-supervised point cloud segmentation via gradual receptive field component reasoning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11673–11682, 2021.

[30] Tchapmi, L.; Choy, C.; Armeni, I.; Gwak, J.; Savarese, S. SEGCloud: Semantic segmentation of 3D point clouds. In: Proceedings of the International Conference on 3D Vision, 537–547, 2017.

[31] Thomas, H.; Goulette, F.; Deschaud, J. E.; Marcotegui, B.; LeGall, Y. Semantic classification of 3D point clouds with multiscale spherical neighborhoods. In: Proceedings of the International Conference on 3D Vision, 390–398, 2018.

[32] Landrieu, L.; Simonovsky, M. Large-scale point cloud semantic segmentation with superpoint graphs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4558–4567, 2018.

[33] Zhang, Z. Y.; Hua, B. S.; Yeung, S. K. ShellNet: Efficient point cloud convolutional neural networks using concentric shells statistics. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 1607–1616, 2019.

[34] Khan, S. A.; Shi, Y. L.; Shahzad, M.; Zhu, X. X. FGCN: Deep feature-based graph convolutional network for semantic segmentation of urban 3D point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 778–787, 2020.

[35] Ma, Y. N.; Guo, Y. L.; Liu, H.; Lei, Y. J.; Wen, G. J. Global context reasoning for semantic segmentation of 3D point clouds. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, 2920–2929, 2020.

[36] Lei, H.; Akhtar, N.; Mian, A. SegGCN: Efficient 3D point cloud segmentation with fuzzy spherical kernel. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11608–11617, 2020.

[37] Huang, Q. G.; Wang, W. Y.; Neumann, U. Recurrent slice networks for 3D segmentation of point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2626–2635, 2018.

[38] Lin, Y. Q.; Yan, Z. Z.; Huang, H. B.; Du, D.; Liu, L. G.; Cui, S. G.; Han, X. FPConv: Learning local flattening for point convolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4292–4301, 2020.

[39] Han, W. K.; Wen, C. L.; Wang, C.; Li, X.; Li, Q. Point2Node: Correlation learning of dynamic-node for point cloud feature modeling. *Proceedings of the AAAI Conference on Artificial Intelligence* Vol. 34, No. 7, 10925–10932, 2020.

[40] Schult, J.; Engelmann, F.; Kontogianni, T.; Leibe, B. DualConvMesh-net: Joint geodesic and Euclidean convolutions on 3D meshes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8609–8619, 2020.

[41] Zhang, F. H.; Fang, J.; Wah, B.; Torr, P. Deep FusionNet for point cloud semantic segmentation. In: *Computer Vision–ECCV 2020. Lecture Notes in Computer Science, Vol. 12369*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 644–663, 2020.

[42] Li, Y. Y.; Bu, R.; Sun, M. C.; Wu, W.; Di, X. H.; Chen, B. Q. PointCNN: Convolution on $X$-transformed points. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 828–838, 2018.

[43] Huang, J. W.; Zhang, H. T.; Yi, L.; Funkhouser, T.; Nießner, M.; Guibas, L. J. TextureNet: Consistent

local parametrizations for learning from high-resolution signals on meshes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4435–4444, 2019.

[44] Lei, H.; Akhtar, N.; Mian, A. Spherical kernel for efficient graph convolution on 3D point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 43, No. 10, 3664–3680, 2021.

[45] Yan, X.; Zheng, C. D.; Li, Z.; Wang, S.; Cui, S. G. PointASNL: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5588–5597, 2020.

**Jingyu Gong** received his B.S. degree in physics from Shanghai Jiao Tong University, China, in 2019, where he is now a Ph.D. student. His research interests cover 3D point cloud segmentation.

**Zhou Ye** received his master degree in software engineering from Nanjing University. He is now the CTO of Shanghai CLS Fintech Co., Ltd. His research interests include computer vision, deep learning, and their applications to the stock market.

**Lizhuang Ma** received his B.S. and Ph.D. degrees from Zhejiang University, China, in 1985 and 1991, respectively. He is now a Distinguished Professor at the Department of Computer Science and Engineering, Shanghai Jiao Tong University and the School of Computer Science and Technology, East China Normal University. His research interests include computer vision, computer aided geometric design, computer graphics, scientific data visualization, computer animation, digital media technology, and theory and applications of computer graphics and CAD/CAM.