CGA-Net: Category Guided Aggregation for Point Cloud Semantic Segmentation

Tao Lu Limin Wang* Gangshan Wu
State Key Laboratory for Novel Software Technology, Nanjing University, China

Abstract

Previous point cloud semantic segmentation networks use the same process to aggregate features from neighbors of the same category and different categories. However, the joint area between two objects usually only occupies a small percentage in the whole scene. Thus the networks are well-trained for aggregating features from the same category point while not fully trained on aggregating points of different categories. To address this issue, this paper proposes to utilize different aggregation strategies between the same category and different categories. Specifically, it presents a customized module, termed as Category Guided Aggregation (CGA), where it first identifies whether the neighbors belong to the same category with the center point or not, and then handles the two types of neighbors with two carefully-designed modules. Our CGA presents a general network module and could be leveraged in any existing semantic segmentation network. Experiments on three different backbones demonstrate the effectiveness of our method.

1. Introduction

3D sensor plays an important role in perceiving the environment geometry, it's widely equipped in home service robots, autonomous cars and even some mobile devices. Point cloud is an efficient data type to represent the 3D scene. Recently, lots of scene understanding works [22, 23, 30, 15, 27, 34, 35, 3] are committed to designing point-cloud-based neural networks to analyze the semantic label for each point. They usually use an ”Encoder-Decoder” architecture. The encoder extracts features for every point by aggregating features from neighbors progressively. And the decoder combines the low-level features and the propagated neighboring features to parse the representations to semantic labels.

In a real scene, some objects are distinctive, like tables, chairs, and planes. We can classify its category by aggregating the features from the object itself. Some objects are easily confused, like the cup and vase, windows and doors. Because they are very similar in 3D contour. Since in a real scene, there is a certain dependency between multiple objects, we can leverage the neighbor context to identify those categories.

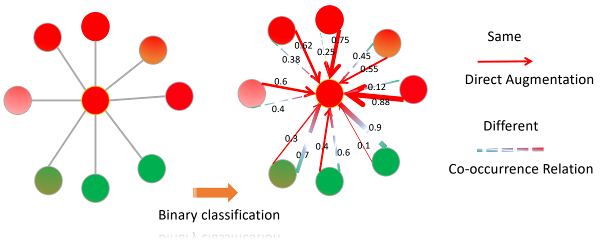

Figure 1. An illustration of our method. We find k neighbors for the center point, and identify the probability that they belong to the same or different categories with the center point. Red arrows implies a directly augmentation to the center point. ”Green-Red” dash line means a co-occurrence relation between neighbors and the center.

For those points located in the center area of an object, its context mainly focus on the object itself. In the joint area of two different categories objects, each point gathers features from both the same and different categories. The features of the same category should be similar and serve as components for object classification together. Meanwhile, the features from the neighbor objects of different categories serve as relation to infer co-occurrence. The learning process for the two types features are different in principle. However, previous works [23, 30, 15, 3] prefer to use a unified operator to cover the two types feature learning. Although the network has strong fitting ability, joint areas only occupy a small percentage in the whole scene, the network tends to learn a group of biased parameters which are more friendly to those areas who keep away from the joint areas, which leading to more ambiguous features in the joint areas. JSENet [6] emphasizes the particularity of edges and designs modules to detect the edge area. It's useful and obtains better results. However, the features for edge detection are different from the semantic segmentation in essence, we think that only detecting the edge is not enough.

*Corresponding author.

In this paper, we propose a neighbor category-guided-aggregation method to augment the representation from any backbones for semantic segmentation. It explicitly models inter-relations between two objects of different categories and enhance intra-consistency in one object. The input are the point-wise representations from any semantic segmentation backbones. First, we find k nearest neighbors for each point. With the local context, we identify whether a point and its neighbors belong to the same category. For those who belonging to the same category, we augment the feature of the center point by a weighted sum of neighbors. For those with different classes, we design a new module to learn how the information from different objects supports the feature learning of the center point. We have conducted several experiments on different datasets and different backbones. All of the results are improved and we achieve new state of the art in these datasets. Visualized results demonstrate that our method improves both the joint areas and object with internal noise. **The key contributions of this paper are as follows**,

- We propose a two-path feature augmentation architecture to handle the information from the same category and the different categories separately.
- A relational module is proposed to explicitly gather support from objects of different categories.
- Several experiments prove the effectiveness of our method.

2. Related Works

Semantic segmentation is a fundamental task in point cloud, aiming to densely predict a semantic label for each point. Since the PointNet [22] is proposed, it becomes a convention for every new point cloud backbone to test on semantic segmentation. Because this task has high demands on the robust feature extraction, local details information preserving and global scene context understanding. According to the type of point cloud backbone, we introduce some related works as follows.

2.1. Point-wise MLP Methods

This type of method is constructed by shared MLP layers. Benefiting from the parallel computing, the MLP module is quite efficient. However, MLP module lacks the ability of describing the local geometry structure of point cloud. Thus, many works are dedicated in designing a local aggregation module to gather features in local area. PointNet++ [23] queries point within multi-scale spheres. ShellNet [40] simplifies it into concentric spherical shells. To handle the imbalance of point cloud in different directions, PointSIFT [9] proposes to query neighbors in 8 octants. Results on indoor semantic segmentation show that this is quite suitable for capturing the local structure. Except for these methods that well-design a search area, other methods just find neighbors using a naive KNN or fixed-radius ball query to aggregate local features. Max pooling is the most commonly used symmetric function to aggregate features from neighbors. Some works [5, 42, 17, 2, 41] propose to learn the weights for aggregating local features. Li et al. [16] proposes a simple PosPool module in the CloserLook3D, which achieves comparable results with those well-designed methods.

2.2. Point Convolution Methods

Inspired by the 2D convolution on image [18], some works design point-cloud-based convolution operations [14, 24, 21]. In Pointwise CNN [7], a convolution kernel is centered at each point of a point cloud, and neighbors within the kernel support can contribute to the center point. KPConv [30] designs a set of kernel points uniformly distributed in Euclidean space. It has excellent performance in capturing local structure and robust to the varying density. The positions of kernel points is determined by an optimization process. For specific task, the position of those kernel points can be shifted through learning, which is a deformable version. Another deformable work is Deformable-Filter [37], it can concentrate on the most related area. According to the image-based segmentation, a large receptive field beneficial for capturing semantic information. DPC [4] designs a dilated convolution operator for point cloud and gets good result. PointConv [36] learns a dynamic filter to generate weights for convolution. It provides detailed math process for how the convolution works and how to implement the proposed process efficiently.

2.3. Other Methods

Except for the above point-based methods, some works project the point onto a structural grid [27, 26, 3] and conduct a grid-based convolution. Some works [10, 11, 43, 32, 19] connect the point to construct a graph and learn feature with graph-based methods. ASIS [33] propose to jointly learn semantic segmentation along with the instance segmentation. JSENet [6] proposes to jointly learn an edge detection and fuses the feature from the edge path to the semantic path. Point cloud sequence semantic segmentation task emerges with the popularity of 3D sensors. MinkowskiNet [3] and MeteorNet [15] are able to handle point cloud sequence directly. And we conduct experiments on such 4D data, which proves that our proposed module is not only suitable for 3D data but also applicable in 4D data.

3. Our Method

Point cloud semantic segmentation task assigns each point in a point cloud with a semantic label l_i, where l_i is one of K pre-defined classes. Lots of neural networks are

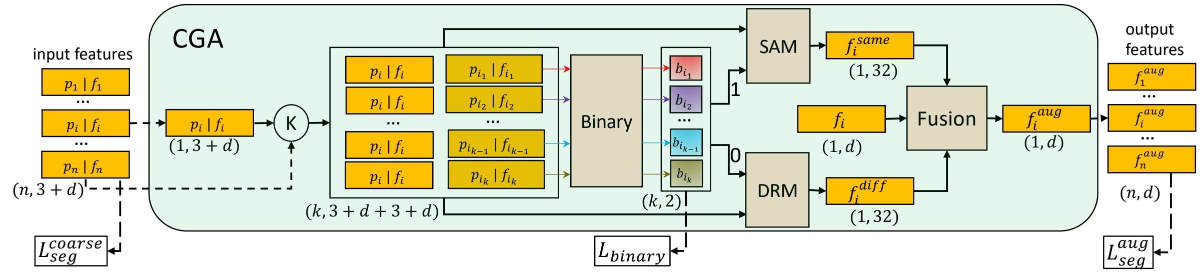

Figure 2. The full architecture of our CGA. Inputs are the representations from the backbone and the xyz positions. For each point, the KNN finds k nearest neighbors and gathers the associated features. The binary module identifies whether neighbors belong to the same category with its center point. '1' implies the same category, and the features will be sent to the SAM module to augment the center point. '0' implies a different category and will be sent to the DRM module to build a relation with its center point. The Fusion module concatenates the outputs of the SAM module, DRM module, and the coarse representation from the backbone, and calculate a final augmented feature for each point. Both the coarse representations and the augmented representations are supervised by the segmentation loss, and the Binary module is supervised by a binary classification loss.

designed to learn representations for each point. We propose the category-guided aggregation (CGA for short), to handle the imbalance inside this backbones. Our method is independent of these backbones, can be plug-in any point cloud semantic segmentation neural network in theory.

We find the k nearest neighbors of each point, and identify whether those neighbors have the same label with the center point. For those with the same label, we aggregate their feature vectors to get a local consistent feature for the center point. For those with different labels, we construct a relation-based module to leverage those different-category points to find clues to support the classification of the center point. By aggregating features from both the same and the different categories, we get an enhanced representation for each point. On the one hand, the enhanced representation reduces the inconsistency among the same category in a local area, especially for those edge areas. On the other hand, it discovers the contextual relations across different categories. The full pipeline is depicted in Fig 2. All supervision signals we use are the semantic labels of all points. We will explain each module in detail in the following sections.

3.1. Details of Each Sub-module

3.1.1 Coarse Feature

We denote the input point cloud of n points as $P \in \mathbb{R}^{n\times(3+c)}$, with position (x, y, z) and other auxiliary information, including the color (r, g, b). A certain backbone assigns each point p_i in P with a coarse feature vector f_i. We denote the whole feature vector of P as $F \in \mathbb{R}^{n\times d}$. According to the feature vector, a coarse classifier predicts the probability of p_i belonging to each category. We choose the category with the highest probability as the final prediction. The predicted labels of P is denoted as $\hat{\mathcal{L}} \in \mathbb{N}^{n\times 1}$. The above processes are formulated as follows:

$$\begin{aligned} &F = \textbf{Net}(P), \\ &Prob = Softmax(\textbf{Cls}^{coarse}(F)), \\ &\hat{\mathcal{L}} = Argmax(Prob), \end{aligned} \tag{1}$$

where **Net(*)** is the backbone and $\textbf{Cls}^{coarse}(*)$ is the coarse classifier. The output probability $Prob \in \mathbb{R}^{n\times K}$ is supervised with a cross-entropy loss as follows,

$$L_{seg}^{coarse} = -\sum_{i=1}^{K} y_i log(Prob_i), \tag{2}$$

where y is the ground truth label.

3.1.2 Binary Module

This module identifies whether the local neighbors belongs to the same category with the center point or not. We first finds k nearest neighbors for all points, denoted as N_p, and gathers the features of neighbors, denoted as N_F. We concatenate the feature F of the center point, neighboring features N_F, and the position difference between the center point and the neighboring points, e.g. $P - N_P$ to obtain an edge descriptor $E^s \in \mathbb{R}^{n\times k\times(d+d+3)}$. A binary classification function **BC** outputs the possibility of each point belonging to the same category with the center point according to the edge descriptor as follows,

$$\begin{aligned} &E^s = Concat(F, N_F, P - N_P), \\ &B = SoftMax(\textbf{BC}(E^s)), \end{aligned} \tag{3}$$

$B \in \mathbb{R}^{n\times k\times 2}$. **BC** is a fully connected layer with no activate function. The neighboring points N_P are then partitioned into two soft collections: $\{N^{same}, N^{diff}\}$, as depicted in Fig 1. This module is supervised by the neighboring category consistency mask, which is generated from the

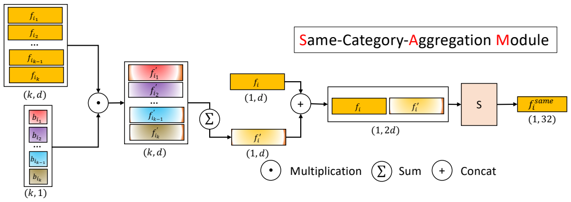

Figure 3. Details of the Same-Category-Aggregation Module.

ground truth label. '1' denotes the neighboring point has the same label with the center point, '0' denotes different labels. This binary classification loss is as follows,

$$L_{binary} = -\sum_{i=0}^{1} m_i log(B_i), \tag{4}$$

where m is the ground truth mask.

3.1.3 Same-Category-Aggregation Module

The soft collections N_{same} is selected by the

$$B^{same} = B[:, :, 1], \tag{5}$$

The values of B_{same} measure how many common parts every neighbor has with the center point. The features of the same category in a local area should not change rapidly. So the common parts in the neighbors should be similar to the center points. We strengthen the feature of the center by collecting the influence of those same category features. The influence on the center point is a weighted sum as follows,

$$F^{same} = \frac{\sum B^{same} * N_F}{\sum B^{same}}, \tag{6}$$

The denominator $\sum B^{same}$ is for scaling the influence to a unified range with the center feature to avoid unstable training. Then, the feature been influenced by the soft same-category collections is computed by concatenating the aggregated feature and the original feature and using a conv1×1 layer **S** to transform the concatenated vector to a new feature as follows,

$$F^{sameaug} = \mathbf{S}(Concat(F, F^{same})), \tag{7}$$

$F^{sameaug}$ represents how the center features are influenced by the soft collections N^{same}. In the view of self-attention, it's a process of local same-category-oriented self-attention. And the soft collections B^{same} serves as the similarity matrix. Details are in Fig 3.

3.1.4 Different-Category-Relation Module

Different category objects provide with co-occurrence information to help classifying the center point. Thus we aim to model the relationship between the center point and the neighboring point with different labels. The soft collections N^{diff} is selected by

$$B_{diff} = B[:, :, 0]. \tag{8}$$

To measure how the neighbors contribute to the center, we construct a new descriptor E^d for the edge between neighbor and center point by subtracting their features and positions. A global relation layer **R** lifts such descriptor to a relation vector F^{rel} to describe how the relations of neighboring points influence the center point as follows,

$$\begin{aligned} E^d &= Concat(F - N_F, P - N_P), \\ F^{rel} &= \mathbf{R}(E^d). \end{aligned} \tag{9}$$

Point-pairs in different positions or between different categories shares the same parameters. Aided by the relation, neighboring points offer strong support in identifying the categories to each other. We compute the total surrounding support by a weighted sum of the individual relation since the more confident that the two points belong to different categories, the more reliable the learned relation should be.

$$F^{diff} = \frac{\sum B^{diff} * F^{rel}}{\sum B^{diff}}, \tag{10}$$

The denominator $\sum B^{diff}$ scales the relation-based feature to a unified range, it makes this module stable during training. B^{diff} reflects how the context influences the center point. And the final influenced feature is transformed as follows,

$$F^{diffaug} = \mathbf{D}(Concat(F, F^{diff})). \tag{11}$$

3.1.5 Fusion Module

The final feature is computed with a conv1×1 layer **A** by concatenating the original feature and the augmented features from the two soft collections as follows,

$$F^{aug} = \mathbf{A}(F, F^{sameaug}, F^{diffaug}), \tag{12}$$

F^{aug} has the same dimensions with the coarse representation F. Then, we re-compute the probability of semantic label by

$$Prob^{aug} = SoftMax(\mathbf{Cls}^{aug}(F^{aug})). \tag{13}$$

$\mathbf{Cls}^{aug}$ is a classifier for the augmented feature. The output probability $Prob^{aug} \in \mathbb{R}^{n \times K}$ is also supervised by the ground truth label with a cross entropy loss as follows

$$L_{seg}^{aug} = -\sum_{i=1}^{K} y_i log(Prob_i^{aug}). \tag{14}$$

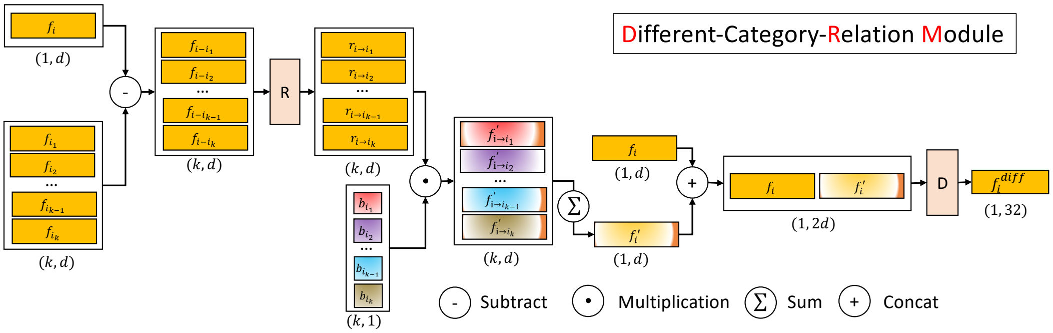

Figure 4. Details of the Different-Category-Relation Module.

3.2. Implementation details

3.2.1 Backbone

The proposed module is independent of a certain backbone, it can be appended after any backbones who can output the feature of each point for semantic segmentation. Mainstream backbones include pointwise-MLP-based network and pseudo-grid-based point convolution. The former directly learn feature with shared MLP layers, which is very efficient. To aggregate local context, some techniques like neighboring feature pooling and attention-based aggregation are proposed. However, MLP fails to incorporate spatial prior in its design. The point convolution methods aim to design an operator similar to the image convolution for point clouds. It often has an advantage in grasping the geometry of the input space while they rely on a well-designed pseudo grid and the hyper-parameters of the grid often have a big impact. To test the generality of our method, we do experiments on both types of backbones. To further prove the effectiveness, we do experiments on a point cloud sequence semantic segmentation task. During the whole process, the backbones only output a coarse representation for each point, with which we predict the initial category probability for each point. Then, we concatenate the initial representation with the position of each point and feed them into our proposed feature augmentation module.

3.2.2 CGA Module

In order to avoid using some tricks in the CGA module, we implement all the sub-modules as simple as possible. For each fully connected layer, we directly adopt the implementation from the backbone itself. For those layers using an activation function, we use the same one (ReLU, Leaky ReLU etc.) as the one used most frequently in the backbone. As to the number of neighbors, for a dense dataset, we only choose half of the number adopted by the backbone to aggregate local features, for a sparse dataset we use the same number. This is for avoiding this augmentation module occupying too much computation resources. Such simple design criteria is for ensuring that the augmentation results only depend on the architecture, rather than some carefully adjusted details. Furthermore, it shows that this method can be easily migrated to other backbones as a feature augmentation module.

3.2.3 Losses

The output of the binary module is directly put into the SoftMax to get the category consistency probability, without any activations. Both the initial representation and the augmented representation are processed by a linear function for the semantic predictions. They are supervised by the same ground truth labels. The labels for the binary module is generated from the ground truth labels. For all backbones, the total loss is as follows,

$$L_{total} = L_{seg}^{coarse} + L_{seg}^{aug} + L_{binary}, \tag{15}$$

for simplicity, we fix the weights for all losses to 1.0.

4. Experiments

All experiments are conducted on a server with 4 RTX 2080Ti GPUs. And we use python and C to code all of our projects. We choose TensorFlow as our deep learning platform. To study the advantages and limitations of our method, we conduct lots of experiments on a 3D indoor point cloud dataset and an outdoor point cloud sequence dataset. We totally test 3 backbones: RandLA-Net [5], CloserLook3D [16] and MeteorNet [15]. The first two are designed for single frame 3D data. The MeteorNet [15] is used for point cloud sequence task. We will first introduce the experiments on a 3D dataset. Results show that both backbones benefit from the feature augmentation module. And we achieve a new state of the art among all fully supervised methods on the dataset. In section 4.2, we will introduce the experiment on a sequence segmentation dataset. we also achieve a new state of the art in this dataset. In section 4.3, we will present detailed ablation studies to show the effect of each sub-module.

Method	mIoU(%)	ceil.	floor	wall	beam	col.	wind.	door	chair	table	book.	sofa	board	clut.
PointNet [22]	41.1	88.8	97.3	69.8	0.1	3.9	46.3	10.8	52.6	58.9	40.3	5.9	26.4	33.2
SegCloud [29]	48.9	90.1	96.1	69.9	0.00	18.4	38.4	23.1	75.9	70.4	58.4	40.9	13.0	41.6
Eff 3D Conv [39]	51.8	79.8	93.9	69.0	0.2	28.3	38.5	48.3	71.1	73.6	48.7	59.2	29.3	33.1
TangentConv [28]	52.6	90.5	97.7	74.0	0.0	20.7	39.0	31.3	69.4	77.5	38.5	57.3	48.8	39.8
RNN Fusion [38]	53.4	95.2	98.6	77.4	0.8	9.8	52.7	27.9	78.3	76.8	27.4	58.6	39.1	51.0
PointCNN [13]	57.3	92.3	98.2	79.4	0.0	17.6	22.8	62.1	74.4	80.6	31.7	66.7	62.1	56.7
SPGraph [11]	58.0	89.4	969	78.1	0.0	**42.8**	48.9	61.6	84.7	75.4	69.8	52.6	2.1	52.2
ParamConv [31]	58.3	92.3	96.2	75.9	0.3	6.0	**69.5**	63.5	66.9	65.6	47.3	68.9	59.1	46.2
SPH3D-GCN [12]	59.5	93.3	97.1	81.1	0.0	33.2	45.8	43.8	79.7	86.9	33.2	71.5	54.1	53.7
HPEIN [8]	61.9	91.5	98.2	81.4	0.0	23.3	65.3	40.0	75.5	87.7	58.5	67.8	65.6	49.4
MinkowskiNet [3]	65.4	91.8	**98.7**	**86.2**	0.0	34.1	48.9	62.4	89.8	81.6	74.9	47.2	**74.4**	58.6
KPConv(deformable) [30]	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
KPConv(rigid) [30]	65.4	92.6	97.3	81.4	0.0	16.5	54.5	69.5	90.1	80.2	74.6	66.4	63.7	58.1
CT2 [20]	67.4	93.6	97.5	83.6	0.0	34.5	54.5	**78.2**	89.1	79.5	73.4	69.1	64.6	58.5
JSENet [6]	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
RandLA-Net [5]	62.5	92.3	97.7	80.5	0.0	20.9	62.0	35.3	77.7	86.8	74.7	68.8	65.0	50.8
RandLA-Net(aug)	65.4	92.1	98.1	82.2	0.0	33.4	62.6	55.8	75.7	**88.4**	66.4	70.1	73.8	51.6
CloserLook3D(pseudo-grid) [16]	65.7	93.9	98.3	82.0	0.0	18.2	56.6	68.0	91.2	80.3	75.3	58.4	70.6	60.8
CloserLook3D(aug)	**68.6**	**94.5**	98.3	83.0	0.0	25.3	59.6	71.0	**92.2**	82.6	**76.4**	**77.7**	69.5	**61.5**

Table 1. Comparisons with state of the art on area 5 of S3DIS [1]. *(aug) denotes augmentation by the CGA module. Red numbers mean better results than the baseline, **Red** and **black** numbers denote the best results among all methods.

Method	#Frames	mIoU(%)	Bldng	Road	Sdwlk	Fence	Vegitn	Pole	Car	T.sign	Pdstr	Bycyc	Lane	T.light
PointNet++ [23]	1	79.35	96.88	97.72	86.20	92.75	97.12	97.09	90.85	66.87	78.64	0.00	72.93	75.17
MinkowskiNet [3]	3	82.76	91.99	**98.93**	92.30	87.49	**99.38**	**97.96**	95.15	**85.55**	**95.16**	0.00	**77.73**	71.45
MeteorNet [15]	3	81.80	98.10	97.72	88.65	94.00	97.98	97.65	93.83	84.07	80.90	0.00	71.14	77.60
MeteorNet(aug)	3	**83.75**	**98.24**	98.49	**93.26**	**96.14**	97.44	97.89	**96.32**	83.02	87.59	0.00	76.30	**80.33**

Table 2. Comparison with baseline and state of the art on Synthia. *(aug) denotes augmentation by the CGA module. Red numbers mean better results than the baseline, **Red** and **black** numbers denote the best results among all methods.

4.1. Semantic Segmentation on Point Cloud

4.1.1 Datasets

S3DIS [1] is a high quality indoor dataset, recorded by Matterport camera. It consists of 6 large areas. Each area contains several scenes, like offices, storages, hallway, conference rooms, and so on. The whole dataset has around 273 million points annotated with 13 semantic labels. We use the Area 5 as a testing set and the rest 5 areas as a training set.

4.1.2 Baselines

RandLA-Net [5] is a point-wise-MLP-based backbone, which is proposed to handle the large scale scene. It replaces the farthest point sampling module with random sampling, which significantly speeds up the processing on large-scale scenes. A novel attentive local feature aggregation module is used to increase the receptive field to overcome the drawbacks caused by the random sampling process.

CloserLook3D [16] is a comprehensive work which researches how the different local aggregators influence the feature learning of point cloud. It includes point-wise MLP, pseudo grid, adaptive weights and a novel PosPool methods. In this paper, we choose the pseudo-grid method as our backbone, because it's a good reimplementation of the classic KPConv [30] and the results are more stable.

4.1.3 Training Settings

For fair comparisons with the original backbones, we apply the same training settings with the backbones. Both of RandLA-Net [5] and CloserLook3D [16] downsample the S3DIS [1] with a grid size of 4cm. RandLA-Net uses the aligned version, it randomly samples 40960 points for each block to put into the network. The difference is that CloserLook3D [16] uses the non-aligned version of S3DIS [1], and randomly samples points within a sphere with 2m radius. Besides, a lot of data augmentation tricks, like randomly scaling, randomly rotation, Gaussian noise, are applied in CloserLook3D [16]. We use the same hyperparameters and optimizer with the baseline's original setting. As for the number of nearest neighbors for the proposed feature augmentation module, we use half of the k used in the original baseline's local aggregation module, e.g. for RandLA-Net [5] it's $\frac{16}{2} = 8$, for CloserLook3D [16] it's $\frac{26}{2} = 13$.

4.1.4 Testing Process

Area 5 contains about 80 million points, which cannot be put into the network together. We randomly select a point

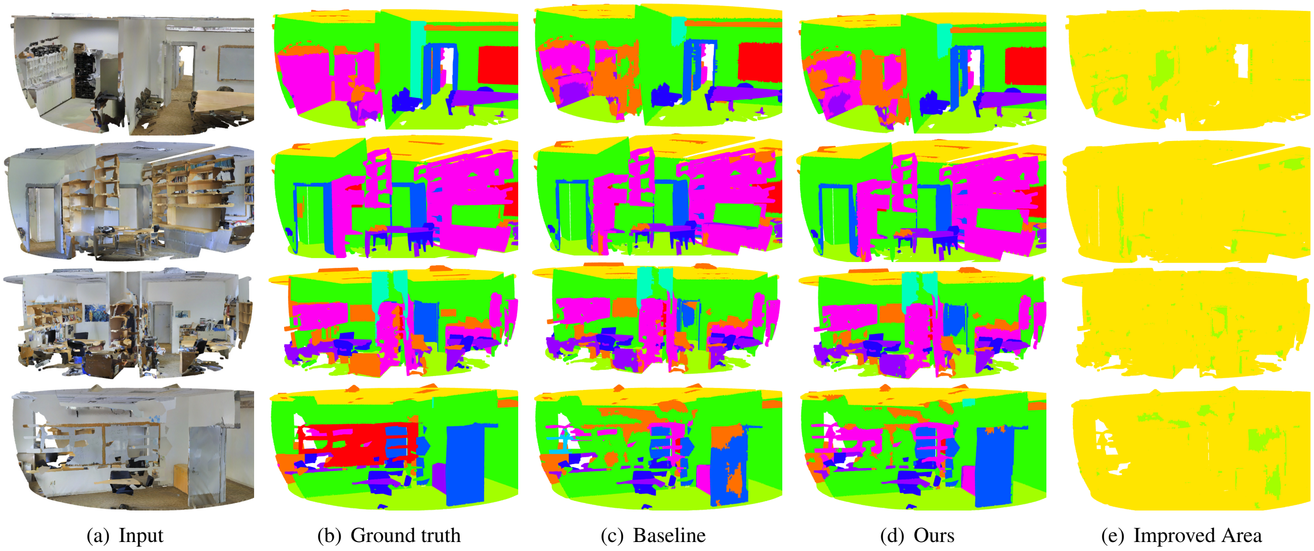

Figure 5. Qualitative results of S3DIS. From the left to the right is: input, ground truth, baseline(CloserLook3D), ours(CloserLook3D aug) and the improved areas. Green areas in the last column means that Ours have a better result than the baseline.

from the area as a center point, then we collect its neighboring points (for RandLA-Net [5] it's 40960 neighbors, for CloserLook3D [16] it's all points within a radius of 2m) to put into the network. Then, we find a new center point far away from the previous center points and collecting new block data. The process is repeated until all points have been fed into the network several times. Finally, a smooth combination module calculates the final probability of each point's category by considering all previous predictions.

4.1.5 Quantitative Results and Analysis

We compare the result of our methods with the baselines and some classical works in this task. And we adopt mIoU on S3DIS [1] Area5 as the evaluation metric.

Comparisons with baselines are presented in Table 1. According to the results, our feature augmentation module improves both the point-wise-MLP-based backbone and the pseudo-grid-based backbone, which prove its effectiveness and powerful generality. Since there are few points on small objects, it is difficult to learn discriminative features from the small object itself. According to the Table 1, most of the small objects get a better segmentation with the help of the local context.

Quantitative comparisons with state-of-the-art methods are also posted in Table 1. The results of RandLA-Net [5] and CloserLook3D [16] are obtained by re-implementation, and they achieve comparable results with the official version. The results of other methods are directly cited from public reports. We get new a state-of-the-art result among all fully supervised point-cloud-based methods. Qualitative results of S3DIS are illustrated in Fig 5. The improved areas are mainly located in the joint area between two objects.

4.2. Semantic Segmentation on Point Cloud Sequences

4.2.1 Dataset

Synthia dataset [25] records 6 sequences of driving scenarios in different weather conditions. The raw data is RGB-D image sequences and it is reconstructed into a point cloud. The scene is cropped by 50m×50m×50m bounding box and each scene is downsampled to 16384 points. We use the same train/val/test split as MeteorNet [15], each split contains 19888, 815, 1886 frames separately. During the experiments, we use a sequence length of 3 frames, each frame with 8192 randomly sampled points.

4.2.2 Baseline

MeteorNet [15] is a point-wise MLP network for processing point cloud sequence. It's an enhanced version of PointNet++ [23], which enables the basic module of PointNet++ [23] to deal with time sequences. It aggregates features through a so-called meteor module, which groups point through direct-grouping or chained-flow-grouping. The direct grouping performs well in point cloud sequence semantic segmentation task. Thus we use it as our backbone.

4.2.3 Comparisons with Baseline

During the implementation of CGA for MeteorNet [15], we search 16 nearest neighbors for every point from the point cloud itself, without searching neighbors from the neighbor frame. The results are posted in Table 2. We get a mIoU of 83.75% for this dataset, which is a new state of the art. Compared with the baseline, we have a clear improvement

0	Coarse Seg	Binary Loss	SAM	DRM	Aug Seg	mIoU(%)
1	✓					65.7
2	✓	✓				66.7
3	✓	✓	✓		✓	67.5
4	✓	✓		✓	✓	67.8
5	✓		✓	✓	✓	66.5
6		✓	✓	✓	✓	64.9
7	✓	Cosine	✓	✓	✓	66.6%
8	✓	GT	✓	✓	✓	70.1%
9	✓	✓	✓	✓	✓	**68.6**

Table 3. Losses combinations.

mIoU(%) \ 0 1	SAM	DRM
SAM	66.2	68.6
DRM	-	67.2

Table 4. Replace module with the other.

Relation	Concat	Addition	Subtraction
mIoU(%)	67.4	67.5	68.6

Table 5. Analysis on DRM.

in the small category, like pedestrian and lane. Because the dataset is sparse, the contours of the small categories are not obvious. Thus they benefit more from the contextual feature augmentation module.

4.3. Ablation Study

To explore how each module influences the full architecture, we conduct some ablation studies on Area 5 of S3DIS with CloserLook3D [16]. All of the following experiments are conducted under the same configurations, except for the controlled conditions.

4.3.1 Module Combinations

The SAM is designed to gather features from the same category and the DRM is designed to model the support from neighbors of different categories. According to Table 3, the absence of any one of them will result in a lower result.

Although the results are improved, the two modules also introduce new parameters. A question emerges naturally: whether the improvement on mIoU comes from more parameters? Thus we design a comparison experiment by replacing the SAM with DRM because DRM even has more parameters than SAM. According to Table 4, the more parameters don't lead to a better result. In addition, we try to replace the DRM with SAM. The result also suffers a drop. The above two experiments prove that the improvements are not coming from the increase of parameters' number, but the well-designed structure.

4.3.2 Loss choices

We have 3 losses in total to train our network. In this part, we study how each loss influences the whole method. Results are posted in Table 3. According to the comparison, the initial segmentation module for coarse representation plays a very important role. It provides reasonable features to identify whether two points belong to the same category. When we remove the binary loss, the output of the binary module serves as an attention-like module, it slightly improves the result. Because the imbalance between joint area and the center area will make the naive attention to perform bad in the joint area. With the full losses combination, we achieve the best result. The "Cosine" means to replace the supervised Binary module with an unsupervised cosine similarity measure and its performance is worse than the supervised manner. To further prove the usefulness of our SAM and DRM module, we replace the Binary module with the ground truth binary label (i.e. no binary loss in training). We achieve a mIoU of 70.1%, which could serve as the performance upper bound.

4.3.3 More analysis on DRM

In the process of designing this module, we have tried several configurations, like replacing the subtraction with concatenation and addition. And the experiment results in Table 5 show that the current subtraction operation is slightly better than others. We think it is due to the subtraction operation explicitly extracts the difference between features of adjacent regions with different semantics. Thus, these contrast features will be helpful to yield more clear prediction boundary (i.e. different prediction across boundary). Besides, the concatenation operation causes more parameters, and for addition operation, it is hard to explain its physical meaning.

5. Conclusions

In this paper, we propose a feature augmentation architecture. It can select different paths for different neighbor points according to the consistency between the categories of neighbors and the center point. When with the same category, the neighbors augment the center point according to the similarity. When with different categories, a DRM module explicitly models the relation between the different categories. We conduct experiments on 2 datasets using 3 different backbones and achieve the state of the arts on the two datasets. This fully demonstrates the effectiveness and generality of our method. Our method only adjusts features according to the nearest neighbors, which is difficult to deal with areas where the error area is large. How to adaptively enlarge the neighboring area is our next goal.

Acknowledgements. This work is supported by the National Science Foundation of China (No. 62076119, No. 61921006), Program for Innovative Talents and Entrepreneur in Jiangsu Province, and Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

[1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1534–1543, 2016. 6, 7

[2] Lin-Zhuo Chen, Xuan-Yi Li, Deng-Ping Fan, Kai Wang, Shao-Ping Lu, and Ming-Ming Cheng. Lsanet: Feature learning on point sets by local spatial aware layer. *arXiv preprint arXiv:1905.05442*, 2019. 2

[3] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 1, 2, 6

[4] Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. Dilated point convolutions: On the receptive field of point convolutions. *arXiv preprint arXiv:1907.12046*, 2019. 2

[5] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11108–11117, 2020. 2, 5, 6, 7

[6] Zeyu Hu, Mingmin Zhen, Xuyang Bai, Hongbo Fu, and Chiew-lan Tai. Jsenet: Joint semantic segmentation and edge detection network for 3d point clouds. *arXiv preprint arXiv:2007.06888*, 2020. 1, 2, 6

[7] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 984–993, 2018. 2

[8] 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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 10433–10441, 2019. 6

[9] Mingyang Jiang, Yiran Wu, Tianqi Zhao, Zelin Zhao, and Cewu Lu. Pointsift: A sift-like network module for 3d point cloud semantic segmentation. *arXiv preprint arXiv:1807.00652*, 2018. 2

[10] Loic Landrieu and Mohamed Boussaha. Point cloud oversegmentation with graph-structured deep metric learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7440–7449, 2019. 2

[11] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4558–4567, 2018. 2, 6

[12] Huan Lei, Naveed Akhtar, and Ajmal Mian. Spherical kernel for efficient graph convolution on 3d point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 6

[13] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Advances in neural information processing systems*, pages 820–830, 2018. 6

[14] Zhidong Liang, Ming Yang, Liuyuan Deng, Chunxiang Wang, and Bing Wang. Hierarchical depthwise graph convolutional neural network for 3d semantic segmentation of point clouds. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8152–8158. IEEE, 2019. 2

[15] Xingyu Liu, Mengyuan Yan, and Jeannette Bohg. Meteornet: Deep learning on dynamic 3d point cloud sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9246–9255, 2019. 1, 2, 5, 6, 7

[16] Ze Liu, Han Hu, Yue Cao, Zheng Zhang, and Xin Tong. A closer look at local aggregation operators in point cloud analysis. *ECCV*, 2020. 2, 5, 6, 7, 8

[17] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. In *Advances in Neural Information Processing Systems*, pages 965–975, 2019. 2

[18] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 2

[19] Yanni Ma, Yulan Guo, Hao Liu, Yinjie Lei, and Gongjian Wen. Global context reasoning for semantic segmentation of 3d point clouds. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 2931–2940, 2020. 2

[20] Kirill Mazur and Victor Lempitsky. Cloud transformers. *arXiv preprint arXiv:2007.11679*, 2020. 6

[21] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4213–4220. IEEE, 2019. 2

[22] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1, 2, 6

[23] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 1, 2, 6, 7

[24] Dario Rethage, Johanna Wald, Jurgen Sturm, Nassir Navab, and Federico Tombari. Fully-convolutional point networks for large-scale point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 596–611, 2018. 2

[25] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016. 7

[26] Radu Alexandru Rosu, Peer Schütt, Jan Quenzel, and Sven Behnke. Latticenet: Fast point cloud segmentation using permutohedral lattices. *arXiv preprint arXiv:1912.05905*, 2019. 2

[27] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. Splatnet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2530–2539, 2018. 1, 2

[28] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3887–3896, 2018. 6

[29] Lyne Tchapmi, Christopher Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3d point clouds. In *2017 international conference on 3D vision (3DV)*, pages 537–547. IEEE, 2017. 6

[30] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6411–6420, 2019. 1, 2, 6

[31] Jiayun Wang, Rudrasis Chakraborty, and Stella X Yu. Spatial transformer for 3d points. *arXiv preprint arXiv:1906.10887*, 2019. 6

[32] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10296–10305, 2019. 2

[33] Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia. Associatively segmenting instances and semantics in point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4096–4105, 2019. 2

[34] 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 *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1887–1893. IEEE, 2018. 1

[35] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4376–4382. IEEE, 2019. 1

[36] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. 2

[37] Yuwen Xiong, Mengye Ren, Renjie Liao, Kelvin Wong, and Raquel Urtasun. Deformable filter convolution for point cloud reasoning. *arXiv preprint arXiv:1907.13079*, 2019. 2

[38] Xiaoqing Ye, Jiamao Li, Hexiao Huang, Liang Du, and Xiaolin Zhang. 3d recurrent neural networks with context fusion for point cloud semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 403–417, 2018. 6

[39] Chris Zhang, Wenjie Luo, and Raquel Urtasun. Efficient convolutions for real-time semantic segmentation of 3d point clouds. In *2018 International Conference on 3D Vision (3DV)*, pages 399–408. IEEE, 2018. 6

[40] Zhiyuan Zhang, Binh-Son Hua, and Sai-Kit Yeung. Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1607–1616, 2019. 2

[41] Chenxi Zhao, Weihao Zhou, Li Lu, and Qijun Zhao. Pooling scores of neighboring points for improved 3d point cloud segmentation. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1475–1479. IEEE, 2019. 2

[42] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. Pointweb: Enhancing local neighborhood features for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5565–5573, 2019. 2

[43] Kang Zhiheng and Li Ning. Pyramnet: Point cloud pyramid attention network and graph embedding module for classification and segmentation. *arXiv preprint arXiv:1906.03299*, 2019. 2