Weekly Report – Point Cloud Group

Wenhe Xu

Contents

RandLA

- LFA
- Attentive Pooling
- Dilated Residual Block

Point-Voxel…

- Point
- Voxel
- Point Voxel

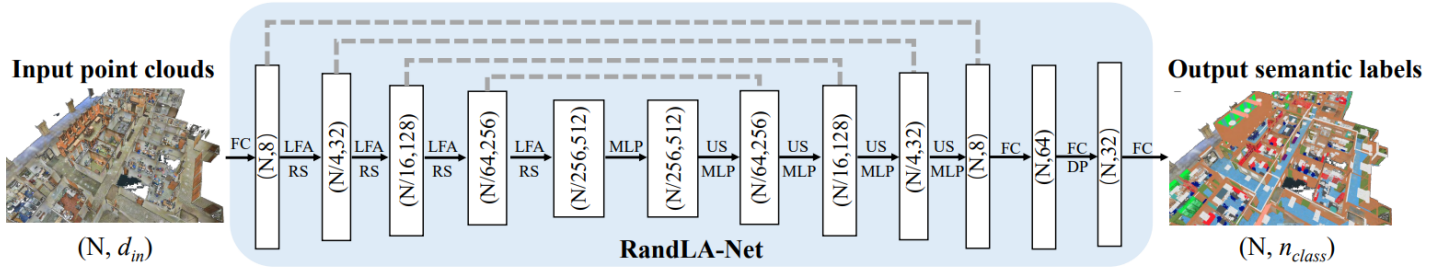
RandLA

Challenge

- **Downsampling Strategy**: Existing algorithms either have high computational costs or consume a lot of memory. For instance, the widely-used farthest-point sampling takes over 200 seconds to downsample a point cloud of 1 million points to 10% of its original size.
- **Feature Learning**: Many methods rely on computationally expensive processes like kernelization or graph construction.
- **Limited Receptive Fields**: Most existing methods have limited receptive fields, making it difficult to efficiently and accurately capture complex geometric structures in large-scale point clouds.

Goal

- Lightweight, computationally-efficient, and memory-efficient network
- Directly handle large-scale 3D point clouds without voxelization, block partition, or graph construction.



RandLA - Downsampling

- Not only significantly reduces the memory footprint of the model, but also allows the model to extract structural features of the point cloud at different scales.
- Existing Downsampling Methods, including:
    1. Heuristic sampling methods Farthest point sampling (FPS)
    2. Inverse Density Importance Sampling (IDIS)
    3. Random sampling (RS)
    4. Learning-based sampling methods Generator-based Sampling (GS)
    5. Continuous Relaxation based Sampling (CRS)
    6. Policy Gradient based Sampling (PGS).

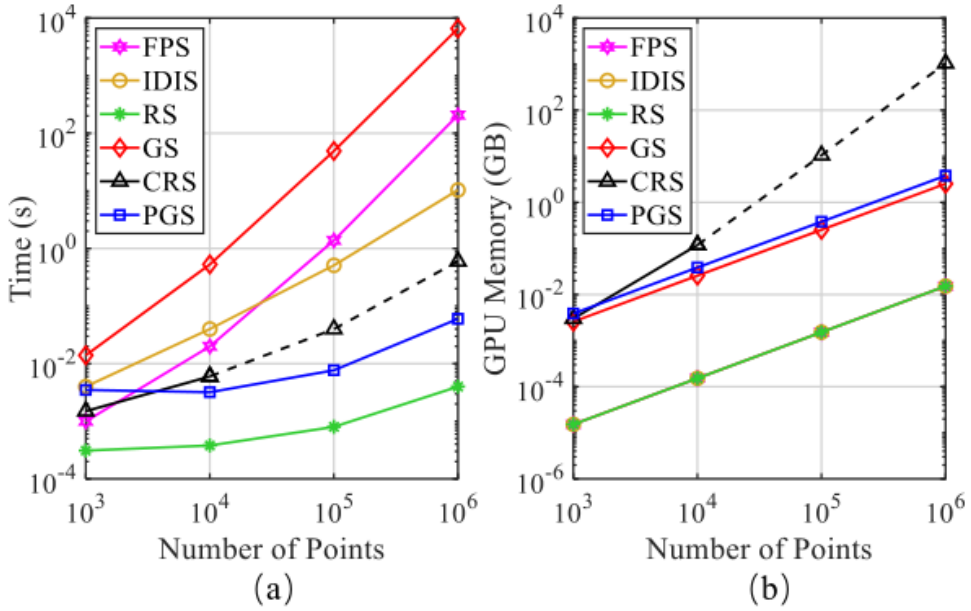
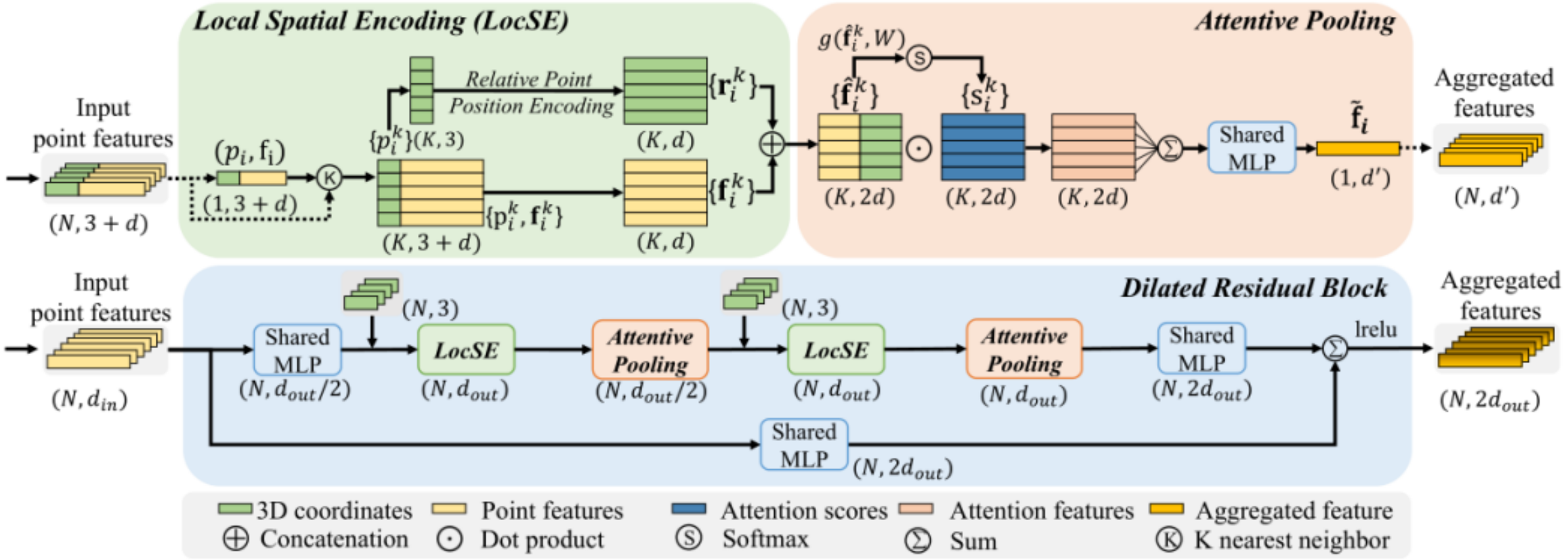


Figure 5. Time and memory consumption of different sampling approaches. The dashed lines represent estimated values due to the limited GPU memory.

RandLA – LFA


Local Spatial Encoding (LocSE)
Input point features
$(N, 3+d)$
(p_i, f_i)
$(1, 3+d)$
K
$\{p_i^k\}(K,3)$
Relative Point Position Encoding
$\{r_i^k\}$
(K,d)
$(K, 3+d)$
$\{p_i^k, f_i^k\}$
$\{f_i^k\}$
(K,d)
Attentive Pooling
$g(\hat{f}_i^k, W)$
S
$\{\hat{f}_i^k\}$
$\{s_i^k\}$
$(K,2d)$
$(K,2d)$
$(K,2d)$
Shared MLP
$\tilde{f}_i$
$(1,d')$
Aggregated features
(N,d')
Input point features
(N, d_{in})
Dilated Residual Block
Shared MLP
$(N, d_{out}/2)$
$(N,3)$
LocSE
(N, d_{out})
Attentive Pooling
$(N, d_{out}/2)$
$(N,3)$
LocSE
(N, d_{out})
Attentive Pooling
(N, d_{out})
Shared MLP
$(N, 2d_{out})$
lrelu
Shared MLP
$(N, 2d_{out})$
Aggregated features
$(N, 2d_{out})$
3D coordinates
Point features
Attention scores
Attention features
Aggregated feature
Concatenation
Dot product
Softmax
Sum
K nearest neighbor

RandLA - LocSE

1. Finding Neighbouring Points (KNN): for the i-th point, which has coordinates p_i and feature f_i, use KNN to find this point. , use KNN to find K neighbors of this point {}, the feature of each neighbor point is:

$$\{f_i^1 \cdots f_i^k \cdots f_i^K\}$$

2. Relative Point Position Encoding (RPPE): For each neighbor point p^k_i of p_i, put together a bunch of information (center point coordinates, neighbor point coordinates, coordinate difference, distance from the center point to the neighbor point), and process it with an MLP to get a new feature:

$$\mathbf{r}_i^k = MLP\left(p_i \oplus p_i^k \oplus \left(p_i - p_i^k\right) \oplus \left\|p_i - p_i^k\right\|\right)$$

3. Neighbor Point Feature Augmentation (Point Feature Augmentation): the original feature f^k_i of each neighbor point and the feature r^k_i after Relative Point Position Encoding are spliced together to form a new feature $\hat{f}_i^k$

$$\hat{F} = \{\hat{f}_i^1 \cdots \hat{f}_i^k \cdots \hat{f}_i^K\}$$

inside the feature of each neighbor point, the position information with respect to the center point is added.

RandLA – Attentive Pooling

This module is used to aggregate the features generated by the local spatial coding module. This module uses the ATTENTION mechanism to feed the point features into the MLP and softmax layers to get the ATTENTION score.

$$\mathbf{s}_i^k = g\left(\hat{\mathbf{f}}_i^k, \boldsymbol{W}\right)$$

Neighborhood point features are weighted and summed using ATTENTION SCORE to get a new centroid feature.

$$\tilde{\mathbf{f}}_i = \sum_{k=1}^{K}\left(\hat{\mathbf{f}}_i^k \cdot \mathbf{s}_i^k\right)$$

RandLA – DRB (Dilated Residual Block)

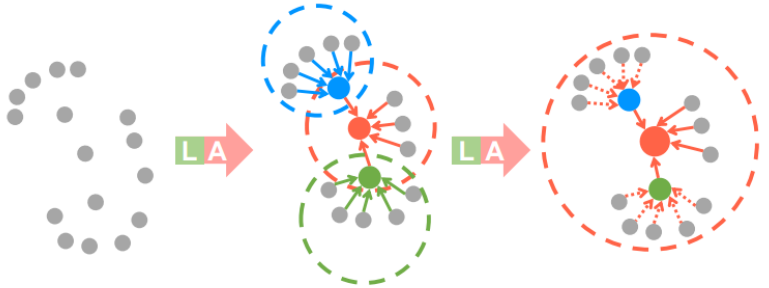
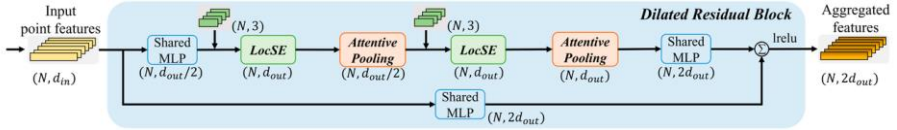


Figure 4. Illustration of the dilated residual block which significantly increases the receptive field (dotted circle) of each point, colored points represent the aggregated features. L: Local spatial encoding, A: Attentive pooling.

- The main idea of this module is to compensate for the loss of information due to random sampling by expanding the receptive field of each point. In this module, two sets of local spatial coding and attentive pooling modules are used to make the receptive field of each point increase from its k nearest neighbors to up to k^2 points.



RandLA – Result:

SotA (2020)

SemanticKITTI

Methods	Size	**mIoU(%)**	Params(M)	road	sidewalk	parking	other-ground	building	car	truck	bicycle	motorcycle	other-vehicle	vegetation	trunk	terrain	person	bicyclist	motorcyclist	fence	pole	traffic-sign
PointNet [43]		14.6	3	61.6	35.7	15.8	1.4	41.4	46.3	0.1	1.3	0.3	0.8	31.0	4.6	17.6	0.2	0.2	0.0	12.9	2.4	3.7
SPG [26]		17.4	**0.25**	45.0	28.5	0.6	0.6	64.3	49.3	0.1	0.2	0.2	0.8	48.9	27.2	24.6	0.3	2.7	0.1	20.8	15.9	0.8
SPLATNet [49]	50K pts	18.4	0.8	64.6	39.1	0.4	0.0	58.3	58.2	0.0	0.0	0.0	0.0	71.1	9.9	19.3	0.0	0.0	0.0	23.1	5.6	0.0
PointNet++ [44]		20.1	6	72.0	41.8	18.7	5.6	62.3	53.7	0.9	1.9	0.2	0.2	46.5	13.8	30.0	0.9	1.0	0.0	16.9	6.0	8.9
TangentConv [51]		40.9	0.4	83.9	63.9	33.4	15.4	83.4	90.8	15.2	2.7	16.5	12.1	79.5	49.3	58.1	23.0	28.4	**8.1**	49.0	35.8	28.5
SqueezeSeg [58]		29.5	1	85.4	54.3	26.9	4.5	57.4	68.8	3.3	16.0	4.1	3.6	60.0	24.3	53.7	12.9	13.1	0.9	29.0	17.5	24.5
SqueezeSegV2 [59]	64*2048	39.7	1	88.6	67.6	45.8	17.7	73.7	81.8	13.4	18.5	17.9	14.0	71.8	35.8	60.2	20.1	25.1	3.9	41.1	20.2	36.3
DarkNet21Seg [3]	pixels	47.4	25	91.4	74.0	57.0	26.4	81.9	85.4	18.6	**26.2**	26.5	15.6	77.6	48.4	63.6	31.8	33.6	4.0	52.3	36.0	50.0
DarkNet53Seg [3]		49.9	50	**91.8**	74.6	64.8	**27.9**	84.1	86.4	25.5	24.5	32.7	22.6	78.3	50.1	64.0	36.2	33.6	4.7	55.0	38.9	52.2
RangeNet53++ [40]		52.2	50	**91.8**	**75.2**	**65.0**	27.8	**87.4**	91.4	25.7	25.7	**34.4**	23.0	80.5	55.1	64.6	38.3	38.8	4.8	**58.6**	47.9	**55.9**
RandLA-Net (Ours)	50K pts	**53.9**	1.24	90.7	73.7	60.3	20.4	86.9	**94.2**	**40.1**	26.0	25.8	**38.9**	**81.4**	**61.3**	**66.8**	**49.2**	**48.2**	7.2	56.3	**49.2**	47.7

Table 3. Quantitative results of different approaches on SemanticKITTI [3]. Only the recent published methods are compared and all scores are obtained from the online single scan evaluation track. Accessed on 31 March 2020.

Semantic3D

	mIoU (%)	OA (%)	man-made.	natural.	high veg.	low veg.	buildings	hard scape	scanning art.	cars
SnapNet_ [4]	59.1	88.6	82.0	77.3	79.7	22.9	91.1	18.4	37.3	64.4
SEGCloud [52]	61.3	88.1	83.9	66.0	86.0	40.5	91.1	30.9	27.5	64.3
RF_MSSF [53]	62.7	90.3	87.6	80.3	81.8	36.4	92.2	24.1	42.6	56.6
MSDeepVoxNet [46]	65.3	88.4	83.0	67.2	83.8	36.7	92.4	31.3	50.0	78.2
ShellNet [69]	69.3	93.2	96.3	90.4	83.9	41.0	94.2	34.7	43.9	70.2
GACNet [56]	70.8	91.9	86.4	77.7	**88.5**	**60.6**	94.2	37.3	43.5	77.8
SPG [26]	73.2	94.0	**97.4**	**92.6**	87.9	44.0	83.2	31.0	63.5	76.2
KPConv [54]	74.6	92.9	90.9	82.2	84.2	47.9	94.9	40.0	**77.3**	**79.7**
RandLA-Net (Ours)	**77.4**	**94.8**	95.6	91.4	86.6	51.5	**95.7**	**51.5**	69.8	76.8

Table 2. Quantitative results of different approaches on Semantic3D (reduced-8) [17]. Only the recent published approaches are compared. Accessed on 31 March 2020.

Appendices of RandLA

Point-Voxel

- Refer to my note on GitHub
- Notes8813/Voxel-point at main · LiliumJadez/Notes8813 (github.com)