

Supplementary Material - Learning Progressive Point Embeddings for 3D Point Cloud Generation

Cheng Wen Baosheng Yu Dacheng Tao

School of Computer Science, Faculty of Engineering,

The University of Sydney, 6 Cleveland St, Darlingtown, NSW 2008, Australia

{cwen6671@uni., baosheng.yu@, dacheng.tao@}sydney.edu.au

1. Details in Our Network

In our paper, we have shown the main framework of proposed point cloud generative network. In this section, we provide more details of its three components.

Discriminator As mentioned in our paper, we attach a point convolution layer or the set abstraction (SA) layer [1] to extract local features after each MLP block. In the main framework, we have shown details of each MLP block. Here, we give the detail parameters of each SA layer used in our experiments. The structure and working principle of SA layer please refer to [1]. When embedding the input points into high dimensional space pointwisely, the output of each MLP block is a set of point features with the size $N \times C_{out}$, where N is the numbers of input points and C_{out} denotes the channels in the output features. For the following SA layer, the input feature size is $N \times C'_{in}$ and the output size is $N_s \times C'_{out}$, where N_s indicates the number of sampled points, and C'_{in} , C'_{out} are the input and output feature channels respectively. Obviously, $C'_{in} = C_{out}$ and we set $C'_{out} = C'_{in}$ for simplicity. In our experiment, we choose $N_s = 450$ sample points and each point locates 5 neighboring points for local feature extraction. Overall, we list these parameters of our discriminator in table 1. For the shape-wise branch in the discriminator, the sizes of three fully-connected layers are 1024, 512 and 128 respectively.

Generator The z and z' in our experiments are both Gaussian noise with mean $\mu = 0$, variance $\sigma = 0.2$ and the size of z is 1×128 . In the upsampling module of the first generator, there are five upsampling layers, and the oversampling layer in our architecture is another upsampling layer. For a better overview, we list their parameters together in table 2. For the second generator, we embed the dense input points from generator one into high dimensional space using a way similar to the discriminator and the sizes of four fully-connected layers are 256, 256, 256 and 2048×3 respectively. In the loss \hat{L}_{gen2} , we set $\lambda_1 = \lambda_2 = 1.0$.

Index	MLP Block		SA Layer			
	C_{in}	C_{out}	C'_{in}	C'_{out}	N_s	n_k
1	3	128	128	128	450	5
2	128	256	256	256	450	5
3	256	512	512	512	450	5
4	512	1024	1024	1024	450	5

Table 1. The details of discriminator parameters in our experiments. There are four combinations of MLP block and SA layer in the architecture, thus the “index” is from 1 to 4. The n_k denotes the number of neighbors for each sampled point in the SA layer.

Generator one	inupt	output	k
upsampling layer 1	1×128	1×128	1
upsampling layer 2	1×128	2×256	2
upsampling layer 3	2×256	16×512	8
upsampling layer 4	16×512	256×128	16
upsampling layer 5	256×128	2048×64	8
oversampling layer	2048×64	8192×3	4

Table 2. The details of the first generator parameters in our experiments. The “input” and “output” stand for the input and output feature size of this layer. k denote the upsampling rate. The product of first five upsampling rates is 2048, which is the same as the number of input points, and the overall product of all layer upsampling rates in the table is 8192.

2. Results Gallery

In this section, we show more generated results of our method in Fig.1.

References

- [1] 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 (NeurIPS)*, pages 5099–5108, 2017.

1

