Global-Local Bidirectional Reasoning for Unsupervised Representation Learning of 3D Point Clouds
Supplementary Material

A. Visualization

To have an intuitive understanding of our models, we visualized the unsupervised learn features on the test set of ModelNet40 in Figure 1. The features are mapped to 2D space by applying t-SNE [4]. We see features from different categories are naturally separated without explicit supervision, which reflects the strong discriminative power of our representation.

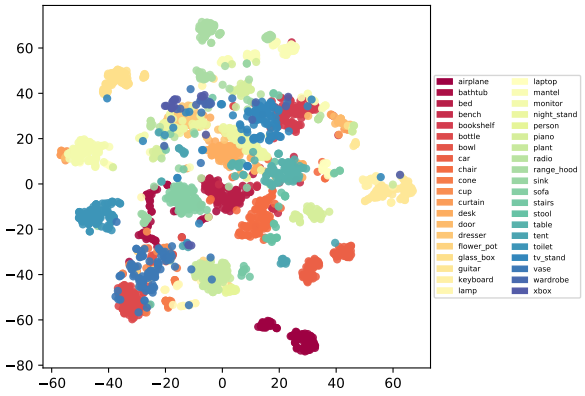

Figure 1: Visualization of unsupervisedly learned representations on the test set of ModelNet40 using t-SNE. Best viewed in color.

B. Network Configuration Details

We first present the details of the single-scale grouping PointNet++ [5] used in our experiments. To improve the efficiency of the original PointNet++, we divide the multi-layer perceptron (MLP) used in each set abstraction (SA) layer of PointNet++ into two fully connected layers and use them before and after the aggregation operation, which can reduce more than 50% computational cost compared to the original SSG model. The details of the new SSG-SA layer is presented in Table 1.

For clearness, we use the following notations to describe the layer and corresponding setting format:

- SSG-SA(N, K, r, [C_{in}, C_{mid}, C_{out}]) is a single-scale grouping set abstraction layer with N local regions of ball radius r and the number of sampled neighboring points K using channel with configuration [C_{in}, C_{mid}, C_{out}].

- MLP([C_1, ..., C_d]) is a $d-1$ layer multi-layer perceptron with channel width C_1, ..., C_d.

Table 1: The detailed architecture of our SSG-SA layer. C_{in}, C_{mid} and C_{out} are the channel widths. N_{in} and N_{out} are the numbers of input and output points. K is the number of sampled neighboring points.

input size	layer type	output size
(N_{in}, C_{in})	Ball Query	(N_{out}, C_{mid}, K)
(N_{out}, C_{mid}, K)	Conv+BN+ReLU	(N_{out}, C_{mid}, K)
(N_{out}, C_{mid}, K)	Max Pooling	(N_{out}, C_{mid})
(N_{out}, C_{mid})	Conv+BN+ReLU	(N_{out}, C_{out})

The overall network architecture used in our experiments is summarized in Table 2, where M is the channel width multiplier. We use the same hyper-parameters of SA layers as [3].

For experiments based on Relation-shape CNN [3], we use the SSG version of this model following the official implementation.

C. Experiment Details

ScanNet Experiments: ScanNet [1] is a richly annotated dataset of 3D reconstructed meshes of indoor scenes, which contains 1513 scanned and reconstructed scenes. To obtain the 3D object classification dataset from the original ScanNet annotations, we use the instance segmentation labels to extract point clouds of each instances from the complete scenes. We use the ScanNetV2 annotations and splits for training and evaluation, where there are 1201 and 312 scenes for training and testing respectively. Following [2], we select objects from 17 categories. 12060 and 3416 objects are extracted as the training and testing sets for object classification task.

Table 2: The detailed architecture of our SSG PointNet++ model and the auxiliary networks with channel width multipler M.

input	input size	layer type	output size	output name
backbone network				
points	(1024, 3)	SSG-SA(512, 48, 0.23, [3, $64M$, $128M$])	(512, $128M$)	sa1
sa1	(512, $128M$)	SSG-SA(128, 64, 0.32, [3, $128M$, $512M$])	(128, $512M$)	sa2
sa2	(128, $512M$)	MLP([512, 1024]) + Max Pooling	(1, $1024M$)	sa3
prediction networks				
sa1	(512, $128M$)	MLP([$128M$, min($128M$, 512), 512])	(512, 512)	pred1
sa2	(128, $512M$)	MLP([$512M$, 512, 512])	(128, 512)	pred2
sa3	(1, $1024M$)	MLP([$1024M$, 512, 512])	(1, 512)	pred3
aggregated representation				
pred1, pred2, pred3	-	Max Pooling + Concatenation	(1, 1536)	agg
self-reconstruction networks				
agg	(1, 1536)	MLP([1536 + 2, 512, 256, 3])	(1024, 3)	recon_mid
agg, recon_mid	(1024, 1539)	MLP([1536 + 3, 512, 256, 3])	(1024, 3)	recon
normal estimation networks				
agg, points	(1024, 1539)	MLP([1536 + 3, 512, 256, 3])	(1024, 3)	normal

Linear SVM: We use the linear SVM classifier provided by scikit-learn library[1] to evaluate the unsupervised learning algorithms. We use the default parameters in all our experiments. We only extract one feature for each object to form the training and evaluation data. Data augmentation techniques are not used to train the linear SVM.

References

[1] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017. 1

[2] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *NeurIPS*, pages 828–838, 2018. 1

[3] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *CVPR*, pages 8895–8904, 2019. 1

[4] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(Nov):2579–2605, 2008. 1

[5] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, pages 5099–5108, 2017. 1

[1] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html