SoftPoolNet: Shape Descriptor for Point Cloud Completion and Classification

Yida Wang[1], David Joseph Tan[2], Nassir Navab[1], and Federico Tombari[1,2]

[1] Technische Universität München
[2] Google Inc.

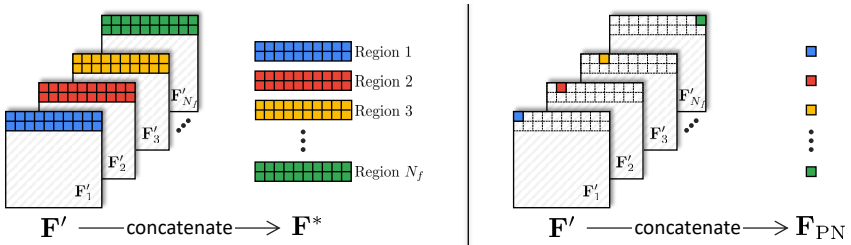
1 Comparison of SoftPoolNet to PointNet and PCN

Our architecture is composed by two parts: *encoder* and *decoder*. The encoder takes the partial scan as input. We process the partial scans with our novel soft pooling to produce the ordered feature F^*. Then, the decoder takes the feature F^* as input and apply our regional convolution twice to produce the point clouds with resolutions of 256 and 16,384 successively.

Notably, there are some similar components between our encoder and PointNet [1], as well as our decoder and PCN [3]. The following sections discuss the distinction in more detail.

1.1 Distinction of our encoder from PointNet

Each point on the cloud goes through the multi-layer perceptron (MLP) to accumulate the feature vectors into the matrix $\mathbf{F}$. Then, we sort the feature vectors in a descending order based on the k-th element of each vector. The sorted matrix is denoted as $\mathbf{F}'_i$. After independently sorting all N_f elements, we collect the matrices to form the tensor $\mathbf{F}'$ as shown in Fig. 1(a). We then build our softpool feature by taking the first N_r elements of each matrix and concatenate them to $\mathbf{F}^*$.



(a) *Soft*-pooling from the proposed (b) *Max*-pooling from PointNet

Fig. 1: Comparison between (a) our soft-pool operation and (b) max-pooling from PointNet, where the feature from PointNet is only a subset of our feature.

When comparing our softpool feature $\mathbf{F}^*$ with the feature from PointNet [1] denoted as $\mathbf{F}_{\text{PN}}$, PointNet executes a max-pooling operation on $\mathbf{F}'$ as illustrated in Fig. 1(b). Assuming that both features are derived from the same $\mathbf{F}'$ produced by an MLP, we can conclude that $\mathbf{F}_{\text{PN}}$ is a subset of our feature where

$$\mathbf{F}_{\text{PN}} = \Big[\mathbf{F}'_1[1],\ \mathbf{F}'_2[2],\ \mathbf{F}'_3[3],\ \ldots\ \mathbf{F}'_{N_f}[N_f]\Big] \tag{1}$$

only takes the one value of each matrix while our method takes the first N_r rows. Due to this, the dimensionality of the feature are then distinct. PointNet takes a vector with 1,024 values while we take $N_r \times N_f \times N_f$.

Notably, both our softpool feature and the PointNet feature are permutation invariant, which means that $\mathbf{F}^*$ and $\mathbf{F}_{\text{PN}}$ are the same irrelevant of the order of the input points. This is one of the most important aspect when handling point clouds since this data is unordered.

1.2 Distinction of our decoder from PCN

Based on our decoder architecture in the paper, the resulting feature from the encoder undergoes two successive regional convolution operations. The first converts the features to a course point cloud $\mathbf{P}'_{\text{out}}$ with 256 points. From there, the second regional convolution interpolates from the coarse to a fully-packed point cloud with 16,384 points which is denoted as $\mathbf{P}_{\text{out}}$.

Compared to PCN [3], both approaches execute a coarse-to-fine approach which is performed by our second regional convolution. However, the architecture and the method are different.

Given $\mathbf{P}'_{\text{out}}$, PCN [3] duplicates $\mathbf{P}'_{\text{out}}$ 64 times and appends a 2D coordinates of an 8×8 grid. Then, they use MLP to produce $\mathbf{P}_{\text{out}}$ that locally deforms the 2D grids around each point similar FoldingNet [2]. In contrast, we interpolate 63 samples between every 2 points of $\mathbf{P}'_{\text{out}}$ and use the proposed regional convolution to produce $\mathbf{P}_{\text{out}}$. Compared to MLP in PCN, our regional convolution takes more local samples into account to produce a point in the higher resolution.

2 Ablation study on the softpool feature $\mathbf{F}^*$

Using our architecture trained with $N_r = 32$, we present the qualitative results when only a subset of the rows is selected. The objective is to investigate which parts of the object each region reconstructs first. In Fig. 2, we start by limiting with the first two rows of the feature matrix then increasing N_r to reach 32. By selecting the first 2 features, we observe that the softpool feature focuses on a skeleton of the object without large surfaces. Although the regions reconstruct different parts of the object, they tend to cover the important components like the wings of plane and the wheels of car. As we increase N_r from 2 to 32, the object is slowly completed without huge overlaps between different regions.

In addition to the first 32 rows when setting N_r, we also looked into the rows beyond 32. The lamp in Fig. 3 focuses on the following ranges: [33 : 64],

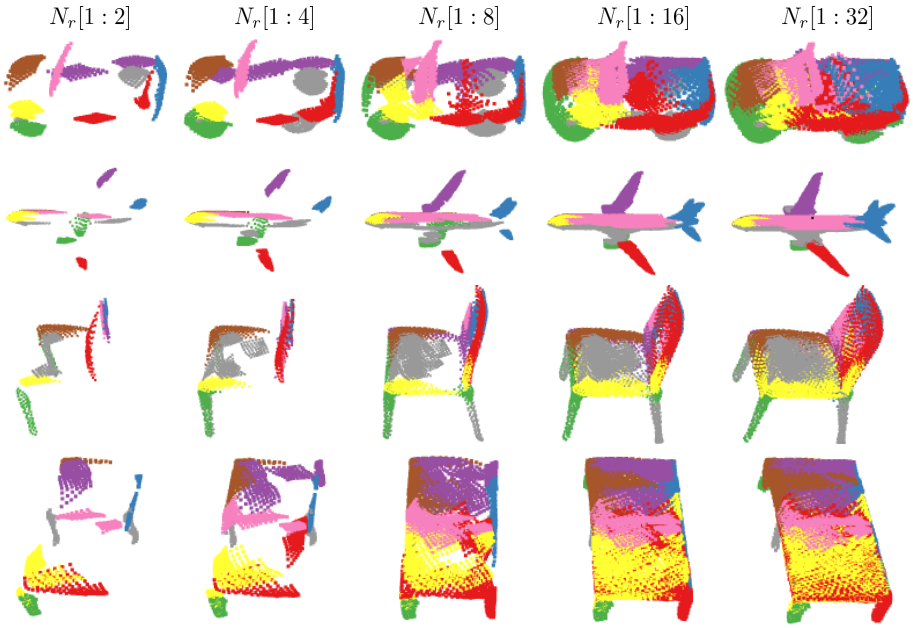


Fig. 2: Results when choosing the first subsets of N_r with the following ranges: $[1:2]$, $[1:4]$, $[1:8]$, $[1:16]$ and $[1:32]$ when the architecture is trained with $N_r = 32$.

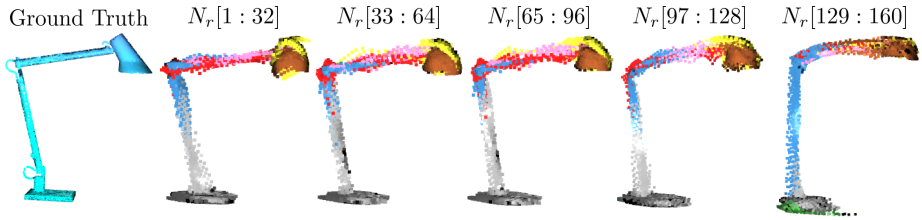


Fig. 3: Results when choosing different ranges of rows from $\mathbf{F}'$ to form $\mathbf{F}^*$ instead of selecting the first $N_r = 32$ rows.

$[65:96]$, $[97:128]$ and $[129:169]$. Although the shape of the lamp starts to deform as we go beyond 32, our reconstruction results still captures its overall shape even when we select the range $[129:169]$. Therefore, this proves that our feature is not constrained to the first 32 rows when sorting and demonstrates the robustness of our softpool feature.

3 Ablation study on τ

When computing for $\mathcal{L}_{\text{boundary}}$, we introduced the threshold τ to compute the sets. In Table 1, we then evaluate different values of τ and investigate its behavior with respect to the Chamfer distance. The table demonstrates that the results are not sensitive to the τ, where the thresholds between 0.2-0.9 generate a small difference in the Chamfer distance (with less than 1) from the chosen threshold of 0.3. Notably, compared to the related work, any threshold between 0.1 to 0.9 outperforms the other methods.

τ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Chamfer Distance	7.08	5.99	**5.94**	6.12	6.19	6.18	6.21	6.25	6.71

Table 1: Sensitivity of the average Chamfer distance (multiplied by 10^3) to the threshold τ.

4 Ablation study on N_r, N_r and $\mathbf{L_{boundary}}$

We investigate the influence of increasing the weight of $\mathcal{L}_{\text{boundary}}$ on the reconstruction as we change the number of regions N_f and the number selected rows N_r. While we chose the best option with N_r set to 8 and N_r set to 32, Table 2 also shows that a larger weight on $\mathcal{L}_{\text{boundary}}$ improves the performance when the number of regions is larger, *e.g.* when N_f is 32.

(N_f, N_r)	(2, 128)	(4, 64)	(8, 32)	(16, 16)	(32, 8)
1 $\times\mathcal{L}_{\text{boundary}}$	7.80	6.31	5.94	6.27	6.75
2 $\times\mathcal{L}_{\text{boundary}}$	7.80	6.31	**5.91**	6.25	6.72
10 $\times\mathcal{L}_{\text{boundary}}$	7.82	6.29	5.95	6.01	6.19

Table 2: Influence of N_f, N_r and the weight of $\mathcal{L}_{\text{boundary}}$ for object completion on the average Chamfer distance (multiplied by 10^3).

References

1. Qi, C.R., Su, H., Mo, K., 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. pp. 652–660 (2017) 1, 2

2. Yang, Y., Feng, C., Shen, Y., Tian, D.: Foldingnet: Point cloud auto-encoder via deep grid deformation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 206–215 (2018) 2
3. Yuan, W., Khot, T., Held, D., Mertz, C., Hebert, M.: Pcn: Point completion network. In: 2018 International Conference on 3D Vision (3DV). pp. 728–737. IEEE (2018) 1, 2