Supplementary for “FBNet: Feedback Network for Point Cloud Completion”

Xuejun Yan[1,3*], Hongyu Yan[2,1⋆], Jingjing Wang[1], Hang Du[1], Zhihong Wu[2], Di Xie[1], Shiliang Pu[1(✉)], and Li Lu[2(✉)]

[1]Hikvision Research Institute [2]Sichuan University [3]Zhejiang University
{yanxuejun,wangjingjing9,duhang,xiedi,pushiliang.hri}@hikvision.com
hongyuyan@stu.scu.edu.cn, luli@scu.edu.cn

1 Source Code

The source code is available at https://github.com/hikvision-research/3DVision/.

2 Detailed Settings

2.1 Network Implementation Details

HGNet. The hierarchical graph-based encoder stacks 3 EdgeConv and 2 AdaptGP layers. Both EdgeConv and AdaptGP use k-nearest neighbors (kNN) as the grouping operation, where we set $k = 16$ throughout the paper. The k of kNN strategy in our cross transformer is set to 16.

We use 3 full-connected layers $(1024 \times 1024, 1024 \times 1024, 1024 \times 128 * 3)$ to decode the global feature to coarse shape P_c with $N_c = 128$. Besides, we aggregate P_c and the partial input, and downsample it to a new coarse output $P_c^{'}$ with size 512×3, which is the input of 0-th FBAC block at first time step($t = 0$).

FBAC block. The FBAC block in the FBNet is a lightweight sub-network and its weight parameters are shared across time steps. For feature extraction, feedback exploitation and feature expansion modules, channel dimensions of these modules' output features are set to 128. The NodeShuffle layer [2] is used to expand features to higher resolution ones. The EdgeConv and MLPs are used to expand input features by r times on channel dimension, and then a shuffle operation is used to rearrange the feature map.

Time Steps and Up-sampling Ratios. We set the input size of first FBAC block to 512×3, and the upsampling ratio in each FBAC block is set to expand point clouds and generate fine-grained shapes. The detailed upsampling ratios of different resolution completion task is shown in Table 1. We set $T = 3$ for 2048 points completion task, and for other higher resolution tasks, we set $T = 2$ to shorten training time and get competitive inference cost compared with recent SOTA works.

⋆ Equal contribution

Table 1: The number of time steps and upsampling ratios for various resolution completion tasks in our FBNet.

Resolutions	Time Steps (T)	Up-sampling Ratios of FBAC blocks
2048	3	1,2,2
4096	2	1,2,4
8192	2	1,2,8
16384	2	1,2,16

Table 2: The effect of feedback refinement mechanism on MVP dataset (2048 points). The FBNet ($T = 3$) is unfolded across time and the output of each time step is evaluated.

t-th Time Step	CD	F1
0	5.79	0.510
1	5.17	0.527
2	5.06	0.530

2.2 Training details

We implement our method with PyTorch and use the Adam optimizer [1] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ to train it.

MVP Dataset. For MVP dataset, we set initial learning rate to 10^{-3} with a decay of 0.1 every 30 epochs for our method. We train our method on NVIDIA V100 16G GPU with batch size 48 and take 100 epochs to converge.

PCN Dataset. In PCN dataset, there are 8 incomplete shapes that are captured from 8 different views for each object. In each training epoch, we use all 8 views data for our method as PCN did [3]. We set initial learning rate to 10^{-3} with a decay of 0.7 every 16 epochs for our method. We train our method on NVIDIA P40 24G GPU with batch size 80 and take 100 epochs to converge.

3 More Results

3.1 The effectiveness of feedback refinement mechanism

We unfold our trained model across time steps and the output of each time step is evaluated. Quantitative results is reported in Table 2, the outputs at present step t gets lower reconstruction error compared to the ones at previous time step $t-1$. The qualitative comparison of different time step is shown in Fig 1, FBNet gradually refines the completion results across time steps via feedback refinement mechanism. Take the third column for example, the completed lamp becomes more complete and less noise with the increasing of time step t. Therefore, our FBNet has the ability to train once and dynamically adjust time step t to balance model inference efficiency and effectiveness for various devices with different computational resources.

Table 3: The time and space complexity on MVP (16K) dataset.

models	PCN	VRCNet	SnowflakeNet	PoinTr	FBNet($T = 1$)	FBNet($T = 2$)
Params (M)	6.86	16.30	19.32	31.27	4.96	4.96
Time (ms)	0.63	29.17	3.47	4.98	2.81	4.58
CD	6.02	3.06	2.73	3.74	2.59	2.29

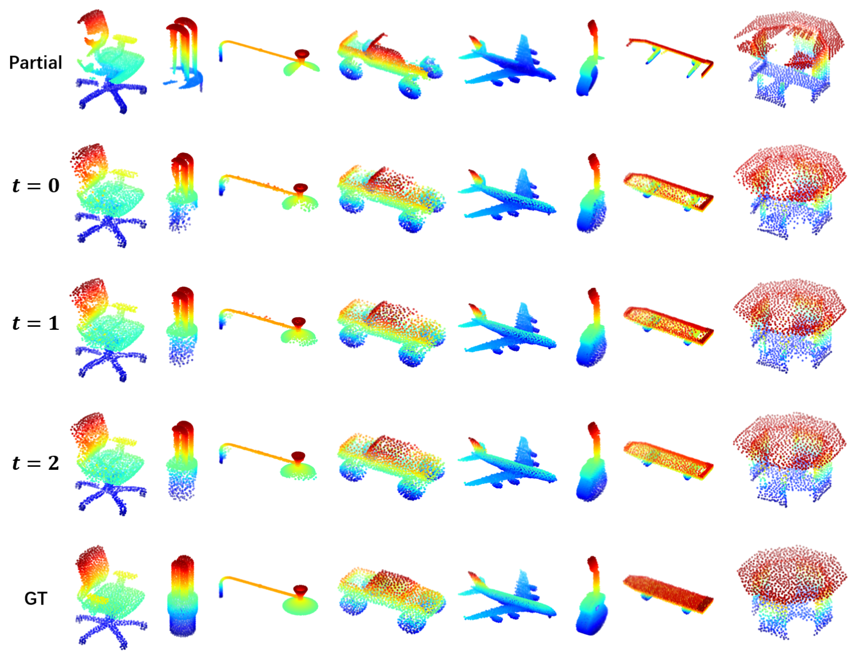

Fig. 1: Qualitative comparison of different time step results of our FBNet. The output is refined gradually across time steps via feedback refinement mechanism.

3.2 The time and space complexity

We compare our FBNet with other methods in terms of parameter size and inference cost on MVP dataset (16384 points). We test the infer time on NVIDIA TITAN X 12G GPU with batchsize 32, the results are shown in Table 3. The FBAC block in the FBNet is a lightweight sub-network and its weight parameters are shared across time steps. Although 3 FBACs are stacked and will be expanded in the time dimension, our FBNet still has the smallest parameter size and competitive inference cost compared with recent SOTA works.

3.3 More visualization results

In Fig 2, we provide more shape completion results on MVP dataset. FBNet has the ability to recover fine-grained details of targets with less noise, especially under challenging categories such as lamp and watercraft.

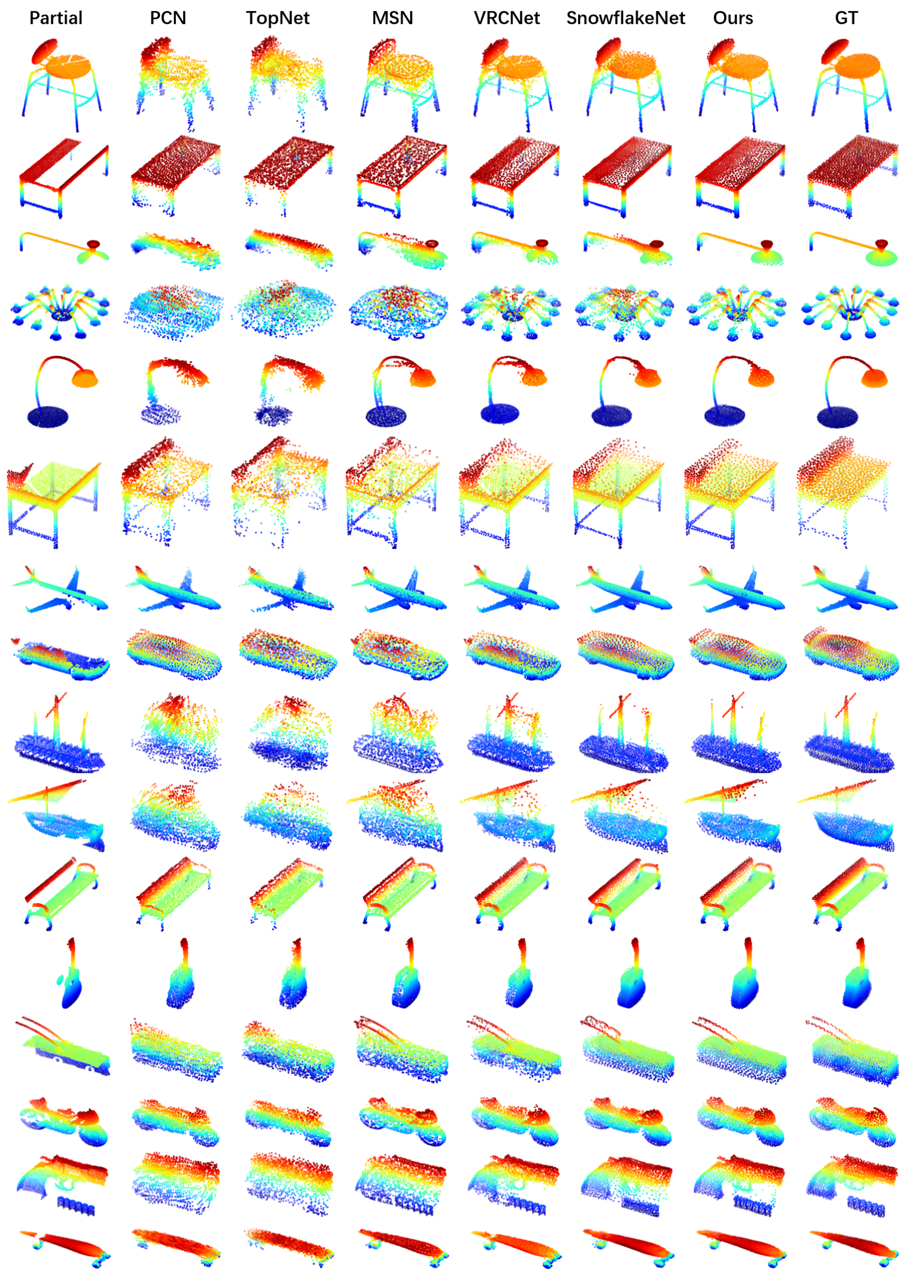

Fig. 2: More qualitative comparison of different methods on MVP dataset(2048 points).

References

1. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
2. Qian, G., Abualshour, A., Li, G., Thabet, A.K., Ghanem, B.: Pu-gcn: Point cloud upsampling using graph convolutional networks. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 11678–11687 (2021)
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)