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

Abstract. The rapid development of point cloud learning has driven point cloud completion into a new era. However, the information flows of most existing completion methods are solely feedforward, and high-level information is rarely reused to improve low-level feature learning. To this end, we propose a novel Feedback Network (**FBNet**) for point cloud completion, in which present features are efficiently refined by rerouting subsequent fine-grained ones. Firstly, partial inputs are fed to a Hierarchical Graph-based Network (HGNet) to generate coarse shapes. Then, we cascade several Feedback-Aware Completion (FBAC) Blocks and unfold them across time recurrently. Feedback connections between two adjacent time steps exploit fine-grained features to improve present shape generations. The main challenge of building feedback connections is the dimension mismatching between present and subsequent features. To address this, the elaborately designed point Cross Transformer exploits efficient information from feedback features via cross attention strategy and then refines present features with the enhanced feedback features. Quantitative and qualitative experiments on several datasets demonstrate the superiority of proposed FBNet compared to state-of-the-art methods on point completion task. The source code and model are available at https://github.com/hikvision-research/3DVision/.

Keywords: Point cloud completion, Feedback Network, Cross transformer

1 Introduction

With the rapid development of 3D sensors, point cloud has been widely used in 3D computer vision applications such as autonomous driving, augmented reality, and robotics. However, due to the limitations of the resolution, occlusion, and view angles, point clouds acquired from 3D sensors are usually sparse, partial, and noisy. Therefore, recovering the complete shape from its partial observation is desirable and vital for various downstream tasks (e.g. shape classification [20], object detection [11,19], and semantic/instance segmentation [34]).
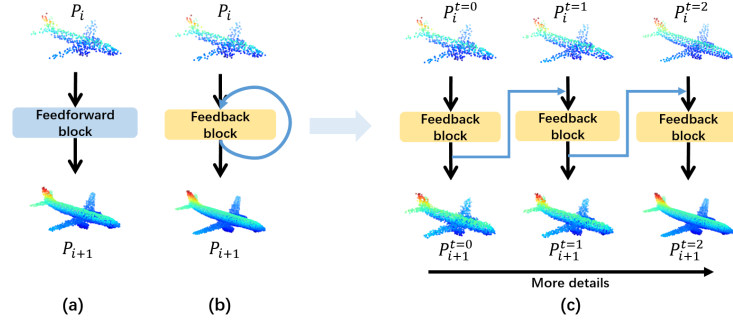

⋆ Equal contribution

Fig. 1: The illustrations of the feedforward and feedback blocks for point completion. (a) Feedforward block in which information only flows from low-level block to high-level block. (b) Feedback mechanism in our proposed FBNet which refines present information with high-level information via feedback connections (represented by the blue arrow). (c) Unfolding feedback block across time steps.

The pioneering work PCN [45] applied an encoder-decoder net to predict coarse shapes from partial inputs firstly, and then refined coarse results to fine-grained completions through a folding-based decoder. Enlightened by the success of PCN's two-stage generation fashion (coarse-to-fine), most recent point completion works [45,26,30,16,31,18] focus on enhancing the the performance of the network of coarse or fine stages to get more fruitful final results. More recent works [30,37,8] extended two-stage (coarse-to-fine) strategy to multistage point generation and achieved impressive completion performance. However, these completion networks are wholly feedforward: the information solely flows from lower stages to higher ones. It is reasonable to infer that the output of a regular completion network has more details and higher quality compared to its input, as shown in Fig. 1 (a). Similarly, for a two-stage or multistage network, higher stages also generate better results compared to lower ones. *Is it possible to utilize higher quality information to make the lower stage features more representative and informative?*

Motivated by the successful applications of the feedback mechanism on 2D vision [15,25,4,46,2,13], we propose the feedback network (FBNet) for point cloud completion. To the best of our knowledge, FBNet is the first feedback-based network for point completion task. The feedback mechanism aims to transmit high-level features to previous layers and refine low-level information. As shown in Fig. 1 (b), the feedback block reuses the output information to enhance its own feature learning via feedback connections. When unfolding it across time, the feedback block takes the state of the previous time step to enrich present feature representations, shown in Fig. 1 (c). As a result, the feedback block has the ability to refine its output recurrently across time steps.

The proposed FBNet consists of a Hierarchical Graph-based Network (HGNet) and the feedback refinement module which stacks three Feedback-Aware Com-

pletion (FBAC) blocks, as shown in Fig. 2. The HGNet is an encoder-decoder structure that encodes partial inputs to global features and then decodes them to coarse complete point clouds. The existing point cloud pooling methods [21,50] downsample point clouds via FPS algorithm and focus on pooled point feature learning. Due to unlearnability, the FPS algorithm may fail to correctly capture local structures when the input is partial and sparse. Hence, the Adaptive Graph Pooling (AdaptGP), the learnable downsampling method, is applied in the HGNet to get more fruitful and accurate global features. After getting coarse complete shapes from HGNet, the feedback refinement module with stacked FBAC blocks refine the coarse complete point clouds to fine-grained and dense ones. For each FBAC block, the high-level information is rerouted to the low layer through the feedback connection and helps to make low-layer features more representative and informative. The main challenge of fusing high-layer information with low-layer ones is the dimension mismatching between two layers. To address this problem, the proposed Cross Transformer builds the relationships between two mismatched point features, and adaptively selects useful information from rerouted features to enhance low-layer features via cross attention strategy. The stacked FBAC blocks gradually refine their outputs across time steps and finally get impressive complete shapes. The proposed FBNet achieves state-of-the-art performance on several benchmarks with various resolutions. Our key contributions are as follows:

- We propose a novel feedback network (FBNet) for point completion, which recurrently refines completion shapes across time steps. To the best of our knowledge, the proposed FBNet is the first feedback-based network for point completion task.
- We introduce the feedback-aware completion (FBAC) block to refine coarse complete point clouds to fine-grained ones. Compared with previous feedforward completion works, feedback connections in FBAC blocks reroute high-level point information back to the low layer and make low-level features more representative.
- We design the Cross Transformer to overcome the feature mismatching problem and adaptively select valuable information from feedback features for enhancing low-level features.
- Experiments on several datasets show that the proposed FBNet achieves superior performance compared to state-of-the-art methods.

2   Related Work

2.1   Point Cloud Processing

The pioneering work PointNet [20] used the shared Multi-layer Perceptions (MLPs) and symmetrical max-pooling operation to extract global features on the unorderedness point cloud, which did not take the relationships of local points into count. To solve this problem, PointNet++ [21] introduced a hierarchical architecture with local PointNet layers to capture regional information.

Following point-based works [12,5,43,7,24,49,38,23] focused on how to learn local features more effectively. Recently, PCT [7] and Point Transformer [49] introduced the self-attention strategy of Transformer [28] for point feature learning and achieved impressive results. For achieving convolution-like operation in the point cloud, many convolution-based works [3,14,42,17,36,27,40] built the relationship between local centers and their neighborhoods to learn dynamic weights for convolution. Besides, graph-based methods [10,32,41,50] achieved notable performance for the local aggregation of geometric features, where DGCNN [32] proposed the EdgeConv module to dynamically update point features by learning edge features between two nodes. In this paper, we aim to design a hierarchical graph-based encoder to learn multi-scale geometric features of the partial point cloud.

2.2   Point Cloud Completion

The target of point cloud completion is to recover a complete 3D shape based on its partial observation. Recent advances based on 3D point cloud processing techniques have boosted the research of point cloud completion. PCN [45] first proposed an explicit coarse-to-fine completion framework, which generates a coarse point cloud based on learned global features from the partial inputs and then refines coarse results to fine-grained completions through a folding-based decoder. Enlightened by the success of PCN, the following methods [26,39,16,34,31,9,47,30,33,35,44,8] especially focused on the local feature exploitation, and the decoding operation were applied to refine their 3D completion results. Most recently, VRCNet [18] introduced a novel variational relation point completion dual-path network, which used the probabilistic modeling network and relation enhancement network to capture local point features and structural relations between points. SnowflakeNet [37] stacked several SPD modules to generate a complete shape like the snowflake growth of points in 3D space. Although these methods achieved impressive completion performance, all of them are feedforward networks that the information flows only from low layer to high layer, and ignore that the information from fine-grained shapes can reroute back to correct previous states.

2.3   Feedback mechanism

The feedback mechanism has been widely employed in various 2D image vision tasks [15,25,4,46,2,13]. With feedback connections, high-level features are rerouted to the low layer to refine low-level feature representations. Building feedback connections in 2D vision tasks are convenient because image-based feature maps are regular and ordered. The resolutions of high-layer feature maps can strictly align with lower ones easily. Several works [15,13,46] directly concatenate feedback features with lower ones and then employ convolution layers to fuse features. Similarly, in our work, the high-resolution point features are transmitted back to enrich low-resolution point features. Due to the unorderedness
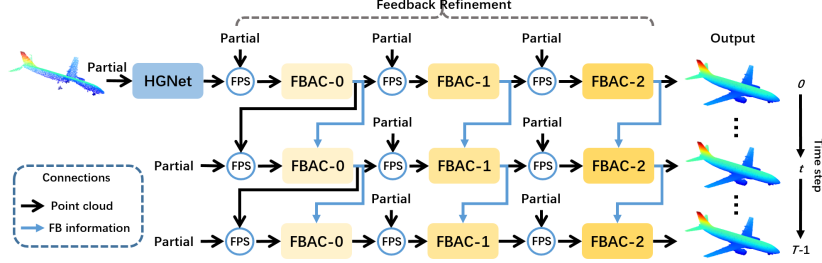
Fig. 2: The overall architecture of our FBNet consists of the Hierarchical Graph-based Network (HGNet) and the feedback refinement module that stacks three Feedback-Aware Completion (FBAC) Blocks. The HGNet aims to generate coarse completions from partial inputs. The cascaded FBAC blocks in the feedback refinement module aim to reconstruct complete and dense point clouds from partial inputs and coarse outputs. Note that, the FBAC block's weight parameters are shared across time steps.

of the point cloud, it is difficult to align two point cloud features with different resolutions.

3   Methodology

The overall architecture of FBNet is shown in Fig. 2, which consists of two modules: one Hierarchical Graph-based Network (HGNet) and the feedback refinement module with three Feedback-Aware Completion (FBAC) blocks. The HGNet generates sparse but complete shapes from partial inputs. The stacked FBAC blocks reconstruct complete and fine-grained point point clouds based on both partial inputs and HGNet's outputs. The feedback connections on these FBAC blocks reroute high resolution point information (points and features) to enrich low resolution point features. With the help of feedback mechanism, FBAC blocks can gradually refine their outputs across time steps and finally get impressive complete shapes.

3.1   Hierarchical Graph-based Network

Hierarchical Graph-based Network (HGNet) is an encoder-decoder structure, which encodes partial inputs to global features and then decodes them to coarse complete point clouds, as shown in Fig. 3. To get more representative global features, we introduce the hierarchical encoder which stacks EdgeConv [32] and proposed Adaptive Graph Pooling alternately.

Adaptive Graph Pooling. The existing point cloud pooling methods [21,50] usually downsampled point clouds via FPS algorithm and aggregated sampled
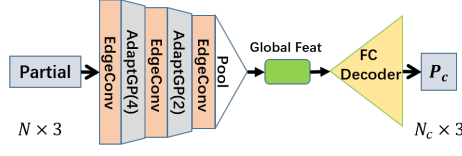
Fig. 3: The architecture of Hierarchical Graph-based Network.

point feature by pooling their neighborhood features. However, the FPS algorithm is unstable, which may fail to pool geometric information efficiently when the input is incomplete and sparse.

To overcome above drawback, we propose a novel Adaptive Graph Pooling (AdaptGP) to pool points and features for partial inputs adaptively. Specifically, AdaptGP generates the pooled points and features by weighting the neighbors of sampled points in geometric and feature space respectively. The weight matrix w is learned from the relation of key point and its neighbors, which can be defined as:

$$w = Softmax(\mathcal{M}((f_i - f_j) + \mathcal{K}(p_i - p_j))) \tag{1}$$

where $\mathcal{M}$, $\mathcal{K}$ are linear mapping functions (i.e., MLPs), which can learn both the point and feature relations between point p_i and point p_j.

The final pooled point and feature of in our AdaptGP can be represented as:

$$p'_i = \sum_{j:(i,j)\in\mathcal{E}} w_j p_j \tag{2}$$

$$f'_i = \sum_{j:(i,j)\in\mathcal{E}} w_{f,j} f_j \tag{3}$$

where $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set of the directed point graph $G(\mathcal{V}, \mathcal{E})$ built by k-nearest neighbors(kNN) grouping operation. In our encoder, we use the two AdaptGP with the pooling rate (4, 2) and three EdgeConv with dimensions (64, 128, 512) to extract multi-scale point features. Finally, we employ max and avg pooling operations to generate the global feature.

Coarse Point Generator. We decode the global feature to predict a coarse output P_c via three fully-connected (FC) layers, as shown in Fig. 3.

With the help of the representative global feature generated by our hierarchical graph-based encoder, the FC layers can recover the coarse shape with more geometric details. Besides, we use FPS to gain new coarse output P'_c from aggregation of P_c and partial input, which provides a better initial input for the following feedback refinement stage.

3.2   Feedback Refinement Mechanism

The feedback refinement module aims to refine the HGNet's coarse result to the complete and fine-grained point cloud. The feedback refinement module consists
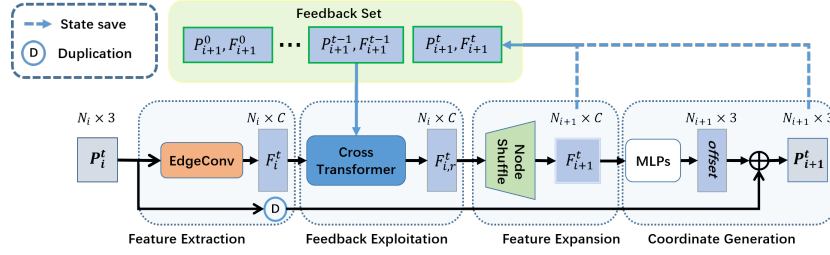
Fig. 4: The detailed architecture of proposed Feedback-Aware Completion Block which consists of four parts: feature extraction, feedback exploitation, feature expansion and coordinate generation.

of three stacked Feedback-Aware Completion (FBAC) blocks and builds multiple feedback connections at various scales to learn more effective features. As shown in Fig. 2 (left-to-right), the information firstly flows from coarse HGNet to stacked FBAC blocks in a feedforward fashion. Each FBAC block takes the output of previous block as input, then refines and upsamples the shape to the denser one with more details.

In particular, for getting better input to i-th FBAC block, we aggregate partial input with feedforward output of $i-1$-th block to a new point cloud, and then downsample it to the same size of previous output via FPS algorithm. Through this initialization operation, the $i-1$-th block's outputs are refined with the original geometric information from partial shapes to get the new inputs, which make present block easier to generate fruitful results. For 0-th block, we aggregate partial input with HGNet's output to initialize its input at $t = 0$ step, and aggregate partial input with its own output at $t-1$ step when $t > 0$.

In our refinement module, the information also flows from high layer to low layer of same FBAC block in the feedback fashion. We unfold FBAC blocks across time steps, shown in Fig. 2 (top-to-down). The weight parameters of FBAC blocks are shared across time steps. For the i-th FBAC block at t step, its high layer features at $t-1$ step are rerouted and used for present step feature learning via the feedback connection. It is reasonable to infer that high layer features at $t-1$ step contain fine-grained information that can refine the low layer features to be more representative at present time t step. As a result, stacked FBAC blocks gradually refine their outputs across time steps and finally fruitful shapes are generated via the feedback refinement mechanism.

3.3 Feedback-Aware Completion Block.

The Feedback-Aware Completion (FBAC) block aims to refine and upsample the low resolution point cloud to high resolution and fine-grained ones via aggregating the points and features from the feedback connection. As shown in Fig. 4, the detailed architecture of proposed FBAC block consists of four parts: feature extraction, feedback exploitation, feature expansion and coordinate gen-
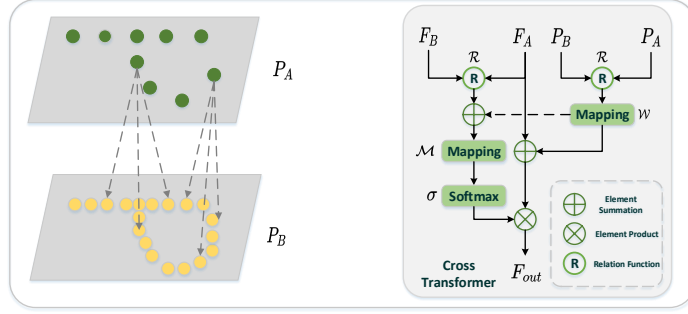
Fig. 5: The detailed structure of our Point Cross Transformer. The left figure visualize the perception process (noted as dashed arrows) of Cross Transformer between two point clouds P_A,P_B. The detail structure of Cross Transformer is shown on the right. P_A, F_A and P_B, F_B denote two different point clouds and their features. The $\mathcal{M}$ and $\mathcal{W}$ are linear mapping functions. $\mathcal{R}$ is a relation function (i.e., subtraction)

eration. We first use EdgeConv [32] to extract local geometric features F_i^t from P_i. Then, the Cross Transformer fuses present features F_i^t with feedback information $(P_{i+1}^{t-1}, F_{i+1}^{t-1})$ generated at $t-1$ step. Subsequently, the refined features $F_{i,r}^t$ is expanded r times via the NodeShuffle [22] block, where $r = N_{i+1}/N_i$. Finally, we gain the point displacements via a series of MLPs.

The final result is calculated as:

$$P_{i+1}^t = \left\{ \mathcal{D}(P_i^t, r) + \mathcal{M}(\eta(\mathcal{T}(P_i^t, \mathcal{G}(P_i^t), P_{i+1}^{t-1}, F_{i+1}^{t-1}), r)) \right\} \tag{4}$$

where $\mathcal{D}$ is the duplication operation. $\mathcal{M}$ is a series of linear functions. η is node-shuffle operation and r is up-sampling ratio. $\mathcal{T}$ is Cross Transformer function. $\mathcal{G}$ is the feature extractor based on EdgeConv.

Cross Transformer. The cross layer feature fusion strategy is widely used for point completion methods [34,18,37]. But in most of them, low-level information flows from a low layer to a high layer with same resolution in a feed-forward fashion and the alignment between two features is required. Take the skip-transformer in SnowflakeNet [37] for example, the two skip-connected features are the displacement feature of the previous SPD and the points feature extracted from the input of the present SPD (also the output of the previous SPD). These two features have the same resolution and a strict one-to-one mapping is built between them. As a result, it is convenient to fuse them via the concatenation operation in the Skip Transformer [37].

In our FBAC block, the feedback feature F_{i+1}^{t-1} has higher resolution compared with the extracted feature F_i^t. Regular fusion strategies (e.g., Skip Transformer, Concatenation) can not be applied directly since they only process the fusion of two point clouds that have the same resolution and their features are aligned.

To address this, we propose a novel point Cross Transformer to fuse features from two point clouds with different resolutions. As shown in Fig. 5, the P_A and

P_B are two point clouds with different resolutions. By constructing cross-points relations between P_A and P_B, our Cross Transformer can utilize P_B information to guide the network to learn better feature representation of P_A. The calculation of Cross Transformer can be formulated as:

$$f'_a = \sum_{b:(a,b)\in\mathcal{E}(\mathcal{A},\mathcal{B})} \sigma(\mathcal{M}(f_a - f_b + \delta)) \odot (f_b + \delta) \tag{5}$$

where $\mathcal{E}(\mathcal{A},\mathcal{B})$ is the directed edge set from point cloud P_A to P_B. the $\mathcal{M}$ is a mapping function i.e., $\{MLP \rightarrow ReLU \rightarrow MLP\}$. The $\delta = \mathcal{W}(p_a - p_b)$ which encoding the position embedding. The σ denotes $Softmax$ function. Through building the cross point cloud graph, the proposed Cross Transformer has the ability to query useful information from reference point cloud P_B and then enrich its own features, shown in Fig. 5. When P_A, P_B are the same point cloud, our Cross Transformer degrades to a point transformer [49] with self-attention mechanism.

The Cross Transformer refines present point features with the reference point cloud information via the cross attention strategy. We apply the Cross Transformer to exploit feedback information and enrich present feature representations. In our FBAC block, we build P_A, P_B and their features as follows:

$$\begin{aligned} P_A &= P_i^t, \quad F_A = F_i^t \\ P_B &= [P_i^t, P_{i+1}^{t-1}], \quad F_B = [F_i^t, F_{i+1}^{t-1}] \end{aligned} \tag{6}$$

where $[,]$ is the merge operation.

As a result, present features F_i^t adaptively query valuable information from the merged feature set. It should be noted that there is no feedback features at 0 step , the F_i^t queries its own features and Cross Transformer degenerate to a Point Transformer.

3.4 Training Loss

In our implementation, we use Chamfer Distance (CD) as the loss function, which can be defined as follows:

$$\mathcal{L}_{CD}(P_1, P_2) = \frac{1}{|P_1|} \sum_{x\in P_1} \min_{y\in P_2} \|x - y\|^2 + \frac{1}{|P_2|} \sum_{y\in P_2} \min_{x\in P_1} \|y - x\|^2. \tag{7}$$

where x and y denote points that belong to two point clouds P_1 and P_2, respectively.

The total training loss is formulated as:

$$\mathcal{L} = \mathcal{L}_{CD}(P_c, Y_{gt}) + \sum_{t=0}^{T}\sum_{i=1}^{n} \mathcal{L}_{CD}(P_i^t, Y_{gt}), \tag{8}$$

where P_c denotes the coarse output of HGNet. The P_i^t and Y_{gt} denote the output of $i-1$-th FBAC block at t step and ground truth, respectively. T is the number of the time steps.

Table 1: Point cloud completion results on MVP dataset (16384 points) in terms of per-point L2 Chamfer distance ($\times 10^4$). The proposed FBNet achieves the lowest reconstruction errors in 14 categories. The best results are highlighted in bold.

Methods	airplane	cabinet	car	chair	lamp	sofa	table	watercraft	bed	bench	bookshelf	bus	guitar	motorbike	pistol	skateboard	Avg.
PCN [45]	2.95	4.13	3.04	7.07	14.93	5.56	7.06	6.08	12.72	5.73	6.91	2.46	1.02	3.53	3.28	2.99	6.02
TopNet [26]	2.72	4.25	3.40	7.95	17.01	6.04	7.42	6.04	11.60	5.62	8.22	2.37	1.33	3.90	3.97	2.09	6.36
MSN [16]	2.07	3.82	2.76	6.21	12.72	4.74	5.32	4.80	9.93	3.89	5.85	2.12	0.69	2.48	2.91	1.58	4.90
CRN [30]	1.59	3.64	2.60	5.24	9.02	4.42	5.45	4.26	9.56	3.67	5.34	2.23	0.79	2.23	2.86	2.13	4.30
GRNet [39]	1.61	4.66	3.10	4.72	5.66	4.61	4.85	3.53	7.82	2.96	4.58	2.97	1.28	2.24	2.11	1.61	3.87
NSFA [47]	1.51	4.24	2.75	4.68	6.04	4.29	4.84	3.02	7.93	3.87	5.99	2.21	0.78	1.73	2.04	2.14	3.77
VRCNet [18]	1.15	3.20	**2.14**	3.58	5.57	3.58	4.17	2.47	6.90	2.76	3.45	1.78	0.59	**1.52**	1.83	1.57	3.06
SnowflakeNet [37]	0.96	3.19	2.27	3.30	4.10	3.11	3.43	2.29	5.93	2.29	3.34	1.81	0.50	1.72	1.54	2.13	2.73
FBNet (Ours)	**0.81**	**2.97**	2.18	**2.83**	**2.77**	**2.86**	**2.84**	**1.94**	**4.81**	**1.94**	**2.91**	**1.67**	**0.40**	1.53	**1.29**	**1.09**	**2.29**

Table 2: Quantitative results with various resolutions on the MVP dataset. For CD ($\times 10^4$), lower is better. For F1-score, higher is better.

Method	2048		4096		8192		16384	
	CD	F1	CD	F1	CD	F1	CD	F1
PCN [45]	9.77	0.320	7.96	0.458	6.99	0.563	6.02	0.638
TopNet [26]	10.11	0.308	8.20	0.440	7.00	0.533	6.36	0.601
MSN [16]	7.90	0.432	6.17	0.585	5.42	0.659	4.90	0.710
CRN [30]	7.25	0.434	5.83	0.569	4.90	0.680	4.30	0.740
VRCNet [18]	5.96	0.499	4.70	0.636	3.64	0.727	3.12	0.791
SnowflakeNet [37]	5.71	0.503	4.40	0.661	3.48	0.743	2.73	0.796
FBNet **(Ours)**	**5.06**	**0.532**	**3.88**	**0.671**	**2.99**	**0.766**	**2.29**	**0.822**

4   Experiments

Implementation and Evaluation Metrics. The number of time steps in our FBNet is set to 3 for low resolution (2K) task and for other higher resolution tasks (e.g., 4K, 8K, 16K) is set to 2. Full settings are detailed in the supplementary material. The model's performance is evaluated by Chamfer Distance (CD) and F1-score.

4.1   Evaluation on MVP Dataset.

The MVP dataset is a multi-view partial point cloud dataset. There are 26 incomplete shapes that are captured from 26 uniformly distributed views for each CAD model. Furthermore, MVP consists of 16 shape categories of partial and complete shapes for training and testing. MVP dataset provides various resolutions (including 2048, 4096, 8192, and 16384) of ground truth which can be used to precisely evaluate the completion methods at different resolutions.
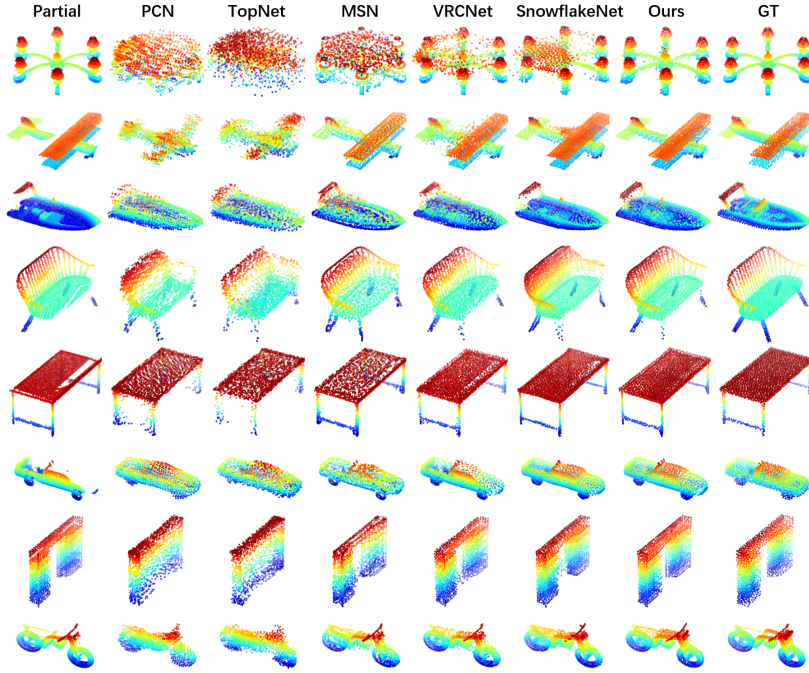
Fig. 6: Qualitative comparison of different methods on MVP dataset (2048 points). Our FBNet generates better complete shapes with fine-grained details compared to other methods.

To verify the effectiveness of our FBNet, we conduct a series of experiments on the MVP dataset. We train our FBNet and SnowflakeNet [37] on the MVP dataset and cite the results of other SOTA methods from [18]. The quantitative results on the high resolution (16384) completion task are shown in Table 1, our FBNet achieves the lowest CD distances in 14 categories. Compared with SnowflakeNet [37], FBNet reduces averaged CD error with a margin of 16%, which demonstrates the superior performance of our method.

The various resolutions completion results are shown in Table 2, our FB-Net outperforms all the other methods with the lowest reconstruction error and highest F1-Score. The visualization results of 2048 points completion are shown in Fig. 6, our FBNet can not only get the lowest reconstruction errors but also recover fine-grained details of the targets. Take the second row, for example, the proposed FBNet can precisely recover the missing fuselage of the plane. PCN [45], TopNet [26] and MSN [16] can only generate coarse plane shape and fail to maintain the original geometric information of partial input. VR-CNet [18] and SnowflakeNet [37] get better results compared to the previous method,however they also generates outliers during the completion processing.

Table 3: Quantitative results on PCN dataset (16384) in terms of per-point L1 Chamfer distance $\times 10^3$). Our FBNet gets the lowest reconstruction error in all 8 categories.

| Methods	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft	Avg.
PCN [45]	5.50	22.70	10.63	8.70	11.00	11.34	11.68	8.59	9.64
TopNet [26]	7.61	13.31	10.90	13.82	14.44	14.78	11.22	11.12	12.15
CRN [30]	4.79	9.97	8.31	9.49	8.94	10.69	7.81	8.05	8.51
GRNet [41]	6.45	10.37	9.45	9.41	7.96	10.51	8.44	8.04	8.83
PMPNet [35]	5.65	11.24	9.64	9.51	6.95	10.83	8.72	7.25	8.73
NSFA [48]	4.76	10.18	8.63	8.53	7.03	10.53	7.35	7.48	8.06
VRCNet [18]	4.78	9.96	8.52	9.14	7.42	10.82	7.24	7.49	8.17
PoinTr [44]	4.75	10.47	8.68	9.39	7.75	10.93	7.78	7.29	8.38
VE-PCN [29]	4.80	9.85	9.26	8.90	8.68	9.83	7.30	7.93	8.32
SnowflakeNet [37]	4.29	9.16	8.08	7.89	6.07	9.23	6.55	6.40	7.21
FBNet (Ours)	**3.99**	**9.05**	**7.90**	**7.38**	**5.82**	**8.85**	**6.35**	**6.18**	**6.94**
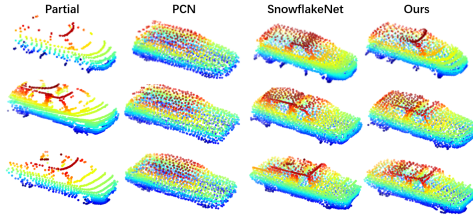
Fig. 7: Qualitative results on KITTI Dataset

Table 4: Quantitative results on KITTI dataset in terms of FD ($\times 10^4$) and MMD ($\times 10^2$). Lower is better.

Methods	FD↓	MMD↓
PCN [45]	11.12	**2.50**
SnowflakeNet [37]	2.08	2.81
FBNet (Ours)	**0.52**	2.97

4.2   Evaluation on PCN Dataset.

We also evaluate our FBNet with other completion methods on the PCN dataset [45]. The PCN dataset is derived from ShapeNet dataset [1], which covers 30974 CAD models from 8 categories. The resolutions of partial input and ground truth are 2048 and 16384, respectively. The quantitative results are represented in Table 3. The proposed FBNet achieves the lowest CD errors in all 8 categories, which shows the robust generalization capability of our method across categories.

4.3   Evaluation on KITTI Dataset.

We further evaluate our method on KITTI [6] dataset, which includes 2401 real-scanned partial cars captured by a LiDAR. We directly test the models trained on the MVP dataset (2048 points) without any finetuning or retraining operations. As the KITTI dataset does not have ground truth shapes, we can not directly calculate reconstruction error of different methods. The PCN [45] proposed the

Table 5: Ablation study of feedback mechanism on MVP dataset (2048 points).

Time Steps T	Feedback	CD	F1
1		5.41	0.527
2		5.36	0.526
3		5.33	0.513
2	✓	5.19	0.529
3	✓	**5.06**	**0.532**

Fidelity Distance (FD) and the Mini Match Distance (MMD) to evaluate the model's performance. We also use these two metrics in our experiment. The quantitative results in terms of FD and MMD metrics are reported in Tabel 4. Our FBNet achieves lower FD compared with SnowflakeNet [37] and PCN [45]. In addition, we visualize the reconstructed results, as shown in Fig. 7. We can see that FBNet can not only generate the general shape of the car but also preserve the observed fine-grained details.

5   Ablation Study

5.1   Feedback Refinement Mechanism

Our ablation experiments are conducted on the MVP dataset (2048 points). We set different time steps and enable/disable the feedback connections to evaluate the effectiveness of the feedback refinement mechanism. The experiment results are shown in Table 5. When number of time steps T is set to 1, our FBNet becomes a feedforward network without unfolding operation across time steps. The FBNet with 2 and 3 time steps are reported at the bottom of Table 5. Our FBNet with feedback mechanism achieves superior performance compared with the feedforward version of FBNet ($T = 1$). The FBNet without feedback connections is also evaluated shown in the second and third rows of Table 5. Without the feedback connections, the performance only gets slightly improved as the feature refinement is invalid across time steps. The gain comes from the updated input of the first FBAC block across time steps.

5.2   Input initialization of FBAC

As shown in Fig. 2, to synthesize the input of the present FBAC block, we aggregate partial and output of the previous block to a new point cloud and then downsample it via the FPS algorithm. Through this initialization operation, we refine the previous outputs with the original geometric information from partial shapes. We conduct ablation studies to evaluate the influence of different input initialization strategies. For the first FBAC block, we design two initialization ways: aggregation of HGNet and partial input P_c+P_{part} across all steps, aggregation of feedback points and partial input $P_{fb}+P_{part}$ at $t > 0$ step. For other

Table 6: The comparison of different input initialization strategies.

Strategy	First Block		Others Block			CD	F1
	P_c+P_{part}	$P_{fb}+P_{part}$	P_{ff}	$P_{ff}+P_{part}$	$P_{ff}+P_{fb}$		
A	✓		✓			5.44	0.520
B	✓			✓		5.28	0.528
C		✓	✓			5.31	0.508
D		✓			✓	19.93	0.233
E(Ours)		✓		✓		**5.06**	**0.532**

blocks, we design three strategies: (1) Feedforward points P_{ff} only. (2)Aggregation of feedforward points and partial input $P_{ff}+P_{part}$. (3) Aggregation of feedforward points and feedback points $P_{ff}+P_{fb}$. The experimental results are reported in Table 6, which proves the effectiveness of our initialization strategy.

5.3 Adaptive Graph Pooling

Table 7: The comparison of different pooling methods used in HGNet on the MVP dataset (2048 points).

Pooling Methods	CD	F1
Graph Pooling [50]	5.48	0.510
Point Pooling [21]	5.23	0.521
AdaptGP (Ours)	**5.06**	**0.532**

We study the effectiveness of the proposed AdaptGP method used in HGNet. The HGNet equipped with AdaptGP gets lowest CD error and best F1-score with a large margin compared with point pooling [21], regular graph pooling [50], as reported in Table 7. The AdaptGP pooling used in HGNet can improve the average performance of our FBNet.

6 Conclusion

In this paper, we propose a novel feedback network for point cloud completion, named FBNet. By introducing the feedback connection in FBAC blocks, FBNet can learn more representative and informative low-level features with the help of rerouted high-level information. As the result, the FBNet gradually refines the completion results across time steps and finally gets impressive complete shapes. Exhaustive experiments on several datasets indicate that our FBNet achieves superior performance compared to state-of-the-art methods.

References

1. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
2. Chen, C., Li, H.: Robust representation learning with feedback for single image deraining. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 7738–7747 (2021)
3. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 764–773 (2017)
4. Feng, M., Lu, H., Ding, E.: Attentive feedback network for boundary-aware salient object detection. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1623–1632 (2019)
5. Gadelha, M., Wang, R., Maji, S.: Multiresolution tree networks for 3d point cloud processing. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 103–118 (2018)
6. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013)
7. Guo, M.H., Cai, J.X., Liu, Z.N., Mu, T.J., Martin, R.R., Hu, S.M.: Pct: Point cloud transformer. Computational Visual Media **7**(2), 187–199 (2021)
8. Huang, T., Zou, H., Cui, J., Yang, X., Wang, M., Zhao, X., Zhang, J., Yuan, Y., Xu, Y., Liu, Y.: Rfnet: Recurrent forward network for dense point cloud completion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12508–12517 (2021)
9. Huang, Z., Yu, Y., Xu, J., Ni, F., Le, X.: Pf-net: Point fractal network for 3d point cloud completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7662–7670 (2020)
10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
11. Li, B., Ouyang, W., Sheng, L., Zeng, X., Wang, X.: Gs3d: An efficient 3d object detection framework for autonomous driving. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1019–1028 (2019)
12. Li, J., Chen, B.M., Lee, G.H.: So-net: Self-organizing network for point cloud analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 9397–9406 (2018)
13. Li, Q., Li, Z., Lu, L., Jeon, G., Liu, K., Yang, X.: Gated multiple feedback network for image super-resolution. In: BMVC (2019)
14. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: Pointcnn: Convolution on x-transformed points. Advances in neural information processing systems **31**, 820–830 (2018)
15. Li, Z., Yang, J., Liu, Z., Yang, X., Jeon, G., Wu, W.: Feedback network for image super-resolution. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 3862–3871 (2019)
16. Liu, M., Sheng, L., Yang, S., Shao, J., Hu, S.M.: Morphing and sampling network for dense point cloud completion. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 11596–11603 (2020)
17. Liu, Y., Fan, B., Xiang, S., Pan, C.: Relation-shape convolutional neural network for point cloud analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8895–8904 (2019)

18. Pan, L., Chen, X., Cai, Z., Zhang, J., Zhao, H., Yi, S., Liu, Z.: Variational relational point completion network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8524–8533 (2021)
19. Qi, C., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3d object detection in point clouds. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 9276–9285 (2019)
20. 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)
21. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. arXiv preprint arXiv:1706.02413 (2017)
22. 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)
23. Ran, H., Liu, J., Wang, C.: Surface representation for point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18942–18952 (2022)
24. Ran, H., Zhuo, W., Liu, J., Lu, L.: Learning inner-group relations on point clouds (2021)
25. Sam, D.B., Babu, R.V.: Top-down feedback for crowd counting convolutional neural network. AAAI (2018)
26. Tchapmi, L.P., Kosaraju, V., Rezatofighi, H., Reid, I., Savarese, S.: Topnet: Structural point cloud decoder. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 383–392 (2019)
27. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: Flexible and deformable convolution for point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6411–6420 (2019)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
29. Wang, X., , M.H.A.J., Lee, G.H.: Voxel-based network for shape completion by leveraging edge generation. In: ICCV) (2021)
30. Wang, X., Ang Jr, M.H., Lee, G.H.: Cascaded refinement network for point cloud completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 790–799 (2020)
31. Wang, Y., Tan, D.J., Navab, N., Tombari, F.: Softpoolnet: Shape descriptor for point cloud completion and classification. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16. pp. 70–85. Springer (2020)
32. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. Acm Transactions On Graphics (tog) **38**(5), 1–12 (2019)
33. Wen, X., Han, Z., Cao, Y.P., Wan, P., Zheng, W., Liu, Y.S.: Cycle4completion: Unpaired point cloud completion using cycle transformation with missing region coding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13080–13089 (2021)
34. Wen, X., Li, T., Han, Z., Liu, Y.S.: Point cloud completion by skip-attention network with hierarchical folding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1939–1948 (2020)

35. Wen, X., Xiang, P., Han, Z., Cao, Y.P., Wan, P., Zheng, W., Liu, Y.S.: Pmp-net: Point cloud completion by learning multi-step point moving paths. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7443–7452 (2021)

36. Wu, W., Qi, Z., Fuxin, L.: Pointconv: Deep convolutional networks on 3d point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9621–9630 (2019)

37. Xiang, P., Wen, X., Liu, Y.S., Cao, Y.P., Wan, P., Zheng, W., Han, Z.: Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5499–5509 (2021)

38. Xiang, T., Zhang, C., Song, Y., Yu, J., Cai, W.: Walk in the cloud: Learning curves for point clouds shape analysis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 915–924 (2021)

39. Xie, H., Yao, H., Zhou, S., Mao, J., Zhang, S., Sun, W.: Grnet: gridding residual network for dense point cloud completion. In: European Conference on Computer Vision. pp. 365–381. Springer (2020)

40. Xu, M., Ding, R., Zhao, H., Qi, X.: Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3173–3182 (2021)

41. Xu, Q., Sun, X., Wu, C.Y., Wang, P., Neumann, U.: Grid-gcn for fast and scalable point cloud learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5661–5670 (2020)

42. Xu, Y., Fan, T., Xu, M., Zeng, L., Qiao, Y.: Spidercnn: Deep learning on point sets with parameterized convolutional filters. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 87–102 (2018)

43. Yan, X., Zheng, C., Li, Z., Wang, S., Cui, S.: Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5589–5598 (2020)

44. Yu, X., Rao, Y., Wang, Z., Liu, Z., Lu, J., Zhou, J.: Pointr: Diverse point cloud completion with geometry-aware transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12498–12507 (2021)

45. 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)

46. Zeng, Y., Lin, Z.L., Yang, J., Zhang, J., Shechtman, E., Lu, H.: High-resolution image inpainting with iterative confidence feedback and guided upsampling. In: ECCV (2020)

47. Zhang, W., Yan, Q., Xiao, C.: Detail preserved point cloud completion via separated feature aggregation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16. pp. 512–528. Springer (2020)

48. Zhao, H., Jia, J., Koltun, V.: Exploring self-attention for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10076–10085 (2020)

49. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16259–16268 (2021)

50. Zhou, H., Feng, Y., Fang, M., Wei, M., Qin, J., Lu, T.: Adaptive graph convolution for point cloud analysis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4965–4974 (2021)