SAUM: Symmetry-Aware Upsampling Module for Consistent Point Cloud Completion

Hyeontae Son and Young Min Kim

Department of ECE, Seoul National University
sonhyuntae@snu.ac.kr youngmin.kim@snu.ac.kr

Abstract. Point cloud completion estimates the complete shape given incomplete point cloud, which is a crucial task as the raw point cloud measurements suffer from missing data. Most of previous methods for point cloud completion share the encoder-decoder structure, where the encoder projects the raw point cloud into low-dimensional latent space and the decoder decodes the condensed latent information back into the list of points. While the low-dimensional projection extracts semantic features to guide the global completion of the missing data, the unique local geometric details observed from partial data are often lost. In this paper, we propose a shape completion framework that maintains both of the global context and the local characteristics. Our network is composed of two complementary prediction branches. One of the branches fills the unseen parts with the global context learned from the database model, which can be replaced by any of the conventional shape completion network. The other branch, which we refer as a Symmetry-Aware Upsampling Module (SAUM), conservatively maintains the geometric details given the observed partial data, clearly utilizing the symmetry for the shape completion. Experimental results show that the combination of the two prediction branches enables more plausible shape completion for point clouds than the state-of-the-art approaches.[1]

Keywords: Point cloud Completion, Point Upsampling, Symmetry, Two-Branch Network

1 Introduction

The real-world 3D measurements enable direct interaction with the physical environment and various applications, such as robotic grasping [1] and SLAM [2]. However, they rely on accurate 3D shapes, which require inferring the unknown geometry given partial measurements. Recent approaches learn a shape prior knowledge from the database of 3D shapes to complete the 3D shape. They train a neural network that generates the complete shape given a partial shape motivated by the success of CNN-based computer vision technology. A straight-forward method is to represent the 3D shape with a dense 3D grid of voxels [3,4,5] and use 3D CNNs which are robust to the irregularity of inputs [6]. However, the

[1] Code available on **https://github.com/countywest/SAUM**

3D grid representation requires memory cubic to the resolution, whereas the 3D shape actually occupies only a sparse set of the dense grid. The memory inefficiency results in a coarse grid resolution, and the completed shape suffers from the quantization effects that loses the fine-grained geometric features [7].

Point-based approaches, on the other hand, represent the 3D shape in terms of points sampled on the surface of the geometry. It is not only memory efficient but also can be directly applied to the real-world 3D measurements to compensate for the prevalent scarcity of the raw point cloud data. Generating point clouds has been suggested for various tasks such as 3D reconstruction from a single image [8,9], super-resolution of point clouds [10,11,12,13], representation learning [14,15,16,17], and shape completion [18,19,20,21,22,23,24,25,26]. Most of them follow the conventional encoder-decoder structure. Given a list of 3D point coordinates, the encoder compresses the high-dimensional data into a low-dimensional global feature vector, whereas the decoder converts the compressed feature vector back into the 3D point cloud representing the shape. Despite the wide range of possible architecture choices, the decoder is designed to regress the 3D point clouds solely from the encoded global feature vector, which inherently focuses on regenerating the global semantics of the input data. While the generated geometry is approximately similar to the inputs, often the fine details are lost or hallucinated.

Our initiative is to create a point cloud shape completion pipeline that preserves the local details which can be observed from the partial measurements. In the case of images, the low-level features are successfully preserved using the U-Net architecture [27] for tasks such as semantic segmentation. The encoder-decoder structure for images are given as symmetric layers; the layers of the encoder network progressively reduce the spatial resolution of the given image to create more abstract representation and the decoder layers gradually increase the resolution back to the original image size. The U-Net model uses so-called "skip connections" between the symmetric layers existing in the encoder and the decoder. As a result, the original feature maps in the early layers of encoders are directly connected to the late-stage decoder layers, which can help the high-frequency details pass to the output directly. However, the skip connections cannot be applied to the networks for point cloud completion; a point cloud is not a regular grid and the numbers of points in input and output are usually not the same.

Instead, we create two complementary branches where one creates the global semantics and the other compensates for existing local details. We refer to the second branch as a **Symmetry-Aware Upsampling Module (SAUM)**, applicable to all of the existing encoder-decoder structured shape completion methods. SAUM is designed to generate locally consistent point clouds by fully utilizing the high-frequency features, and is also able to find the symmetric points of the input point clouds without enforcing symmetry explicitly. The SAUM can be easily combined as a parallel branch to the existing encoder-decoder architecture and overcome the fundamental limitation. Our experiments show that SAUM qualitatively increases the shape fidelity especially on the narrow

structures with fine details, and quantitatively boosts the performance of various existing methods.

2 Related Works

Our framework uses the existing point cloud completion network as a backbone and attaches an additional branch to guide the locally consistent shape completion. We first review the existing point cloud completion, followed by point upsampling, which is the task that preserves the structure but enhance the quality by upsampling. We also compare our network with previous two-branch architecture dealing with 3D reconstruction.

Point Cloud Completion. Point cloud completion has gained recent attention as 3D acquisition becomes readily available with commodity depth sensors. There are two mainstreams for point cloud completion: supervised point completion [18,19,22,24,28,25,26] where the incomplete-complete pairs of point cloud data are given, and unsupervised point completion [20,21,23] without the explicit pairing of data.

The supervised network is trained to minimize the distance between the generated shape and the ground truth complete shape where Chamfer distance or Earth Mover's distance is generally chosen as the metric to compare two point clouds. The basic framework is encoder-decoder structure motivated from autoencoder, where the initial list of 3D points are encoded to make a global feature vector (GFV) from which the decoder generates point cloud.

Unsupervised shape completion uses a generative network motivated by the l-GAN framework [14] for utilizing the learned prior knowledge of the complete shapes. In addition to the distance between the input and output shapes, the unsupervised shape completion is trained with additional losses such as discriminator loss for generating plausible shape and latent space loss for the semantic similarity [20,21,23]. Even with the assistance of a generative network, the basic framework is encoder-decoder structure as the supervised case.

While the previous works share the encoder structure originated from PointNet [29], different choices for decoder architecture have been introduced for shape completion; namely fully connected layers [14], tree-based [19,17], or 2D manifold-based [15,9] networks. Despite the progress in the decoder architecture, the encoder-decoder framework is fundamentally limited as the shape is generated from condensed global information without local guidance. As a result, the outputs are often blurry and inconsistent with the input shapes, instead of fully utilizing the clear geometric features given from the input. Our work attaches an additional network to the existing decoder architecture to complement the performance of existing shape completion and achieve the local consistency.

Point Upsampling. Point cloud upsampling increases the number of samples given the sparse set of points. While the task is not a shape completion, it creates high-resolution data capturing the local details, which resembles our

complementary goal. PU-Net [10] pioneered the deep-learning-based point cloud upsampling. They proposed the "feature expansion" method implemented by separated convolutions. Another key contribution in the paper is proposing the repulsion loss to distribute the points more uniformly to alleviate the tendency for the generated points to stick to each other. Wang et al.[12] and Li et al.[13] used feature expansion but with different methods, namely code assignment inspired by the 2D manifold-based decoders [15,9]. Wang et al.[24] proposed the cascaded networks using the point upsampling networks for the point completion task. One of the strong cues they introduced is the mirror operation assuming the reflection symmetry, successfully generating recurring local structures. Their network requires additional pre-trained auto-encoder to capture the shape prior. On the other hand, our shape completion implicitly learns intrinsic symmetry without any assumptions or additional networks. We adopted the feature expansion used in the PU-Net [10] to shape completion task, and the results show that it could find the local structure of the shapes without additional loss term.

Two-branch Network. Several works attempted to create complementary networks to deal with both local and global contexts of the point cloud shape. PSGN [8] first suggested the two-branch network generating point clouds, and suggested the different nature of the fully-connected networks and convolutional neural networks. Ours adopted the two-branch network for utilizing the local features by SAUM and the global semantic information by the existing decoders. Conceptually, our two-branch network emulates the skip-connection that observes details of original data [27] and global guidance [4]. Authors of [4] implemented global guidance by using the channel-wise concatenation of the global and local features and insisted it plays a key role in the consistent shape completion. The key difference between ours and their methods is the axis of the feature maps when concatenating them. Unlike the channel-wise feature concatenation in the skip-connection and the global guidance, ours uses point-wise concatenation of the generated outputs, which can be interpreted as the union of the point sets. The reason for the difference is the irregular nature of the point cloud representation which cannot be handled as a grid-like structure directly, whereas Han et al. [4] handle shapes in voxel grids and projected images.

3 Two-Branch Network

Given a set of incomplete point cloud P_{in}, our shape completion network creates a set of points P_{out} that is uniformly sampled on the overall shape surface. We present a two-branch network that is composed of the conventional encoder-decoder framework and the symmetry-aware upsampling module (SAUM), as shown in Fig. 1.

One of the challenges with neural network architecture using point cloud is that the structure is irregular with a varying number of inputs, N. The encoder aggregates the information from the varying number of points with maxpooling operation and find a global feature vector (GFV) with a fixed dimension. Then

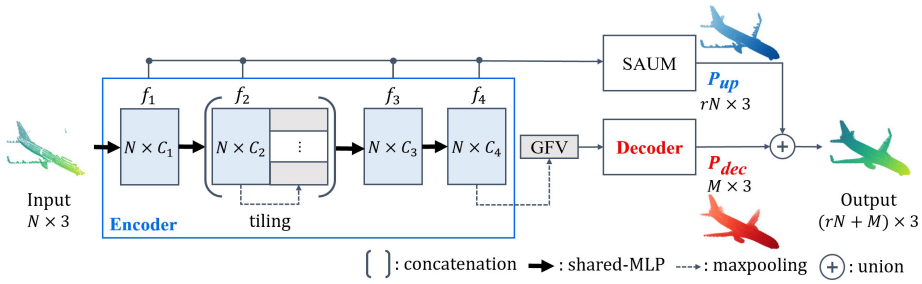

Fig. 1. The overall architecture of the two-branch network using SAUM. SAUM outputs rN points utilizing the multi-level local information. This module can be easily attached to the existing encoder-decoder architecture and the final output is a simple combination of outputs from two branches.

the decoder can use a fixed structure to regress to M points, P_{dec}. The additional SAUM branch upsamples the input P_{in} by the factor of r using multi-level features of the individual points resulting in P_{up} with rN points.

The final completed output is a set union of the two branches

$$P_{out} = P_{up} \cup P_{dec}, \tag{1}$$

where the P_{dec} is the output of the decoder comprised of M points and the P_{out} is the combination of our two prediction branches. As a result, the two-branch network will create $rN + M$ points.

The two-branch network is jointly trained in a supervised setting, minimizing the reconstruction loss for the completed shape compared to the ground truth. The two most popular choices to evaluate the distance between two point sets are the Chamfer distance (CD) and the Earth Mover's distance. We adopted the Chamfer Distance as the reconstruction loss because it can be calculated even if the sizes of two point sets are different. Given two sets of 3D point cloud P_1 and P_2, the Chamfer Distance is defined as

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

We compare the raw output points P_{out} against the ground truth without further sampling, and train the entire network to minimize the reconstruction loss.

In the following, the network structures for the two branches are further explained.

3.1 Encoder-Decoder Network

Most of the previous approaches dealing with shape completion use the encoder-decoder framework, which we also utilize as one of the two branches to capture the global context from the partial data. The encoder follows the conventional

structure motivated by the PointNet architecture. Specifically, features are extracted from individual points with d consecutive shared MLP layers, and we used four shared MLPs as shown in Fig. 1. Starting from the N points with three coordinates each, each MLP transforms them into $C_1, \cdots, C_d$ dimensional features, respectively. We denote the extracted $N \times C_i$ feature from the i-th layer of the encoder as f_i. Then the information from N points at the final depth d layer is combined into a single C_d dimensional vector with max-pooling operation, to which we refer as GFV.

The encoder used in our experiments is the same as Yuan *et al.* [18], which uses tiling the intermediate global feature (Fig. 1). The encoder structure is shared across different choices of decoders, and from the GFV, the decoder then regresses a list of M three dimensional points. The fine details are diluted as the global information is compressed into a single GFV.

3.2 Symmetry-Aware Upsampling Module (SAUM)

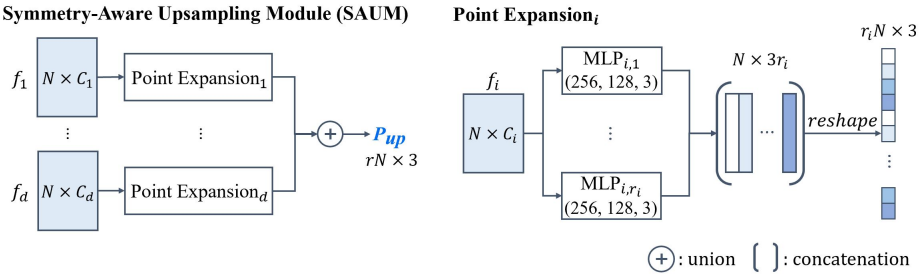

Fig. 2. SAUM architecture. The input is a list of encoder feature map $N \times C_1, \cdots, N \times C_d$, and the output is a set of rN points where $r = \sum_{i=1}^{d} r_i$ (left). SAUM is comprised of the d Point Expansion modules each of which maps the per-point features to upsampled points (right).

To complement the global context captured by the encoder-decoder network, we design a network that contains the multi-layer information of local context captured from intermediate layers of the encoder. SAUM is inspired by the U-Net architecture, which is widely used to transfer low-level features from the encoder to the decoder. The point cloud is not a regular grid structure and the direct connection between the encoder and decoder is not possible. Instead, we use the intermediate features from the encoder and create upsampling networks to find the underlying structure from various levels of abstraction.

Specifically, given the intermediate feature f_i from i-th layer, the $N \times C_i$ dimensional feature is mapped to r_iN points by our *point expansion* module (Fig. 2, left). Inspired by the feature expansion operator [10], the point expansion (written as PE) module for f_i is composed of r_i units of the sequential shared

MLP (Fig. 2, right):

$$PE_i(f_i) = RS([MLP_{i,1}(f_i), \cdots, MLP_{i,r_i}(f_i)]), \tag{3}$$

where

$$MLP_{i,j} = C^3_{i,j}(C^2_{i,j}(C^1_{i,j})). \tag{4}$$

$C^k_{i,j}(\cdot)$ is a single shared MLP block, which is implemented as a 1×1 convolution, meaning that k-th block for the j-th upsampling branch of the i-th feature maps. We designed the last convolution $C^3_{i,j}(\cdot)$ for the output to have 3-dimensional channels so that it can generate a point cloud with size $N \times 3$. Thus, point expansion consists of independent upsampling branches as a special case of the feature expansion [10]. In short, we effectively use three consecutive 1×1 convolution to convert the f_i into three dimensional points for r_i units. $[\cdot]$ operator means the channel-wise feature concatenation, so the output for f_i is a tensor of size $N \times 3r_i$. $RS(\cdot)$ is a reshape operation to convert a $N \times 3r_i$ tensor to a $r_iN \times 3$ point cloud. In the end, the $PE_i(f_i)$ consists of r_i point sets which are the expanded from the i-th layer.

The final output of SAUM is P_{up} which is an union of the each expanded point set $PE_i(f_i)$

$$P_{up} = \bigcup_{i=1}^{d} \{PE_i(f_i)\}. \tag{5}$$

where the union operator $\bigcup$ can be implemented as a point-wise concatenation. Thus, P_{up} is comprised of the upsampled points of each layer, whose size is $rN \times 3$ where $r = \sum_{i=1}^{d} r_i$.

4 Experimental Setting

In this section, we discuss the datasets and the implementation details to test the performance of our suggested shape completion network.

4.1 Datasets

PCN dataset. The PCN dataset [18] is composed of pairs of partial point cloud and corresponding complete point cloud derived from the eight categories of ShapeNet dataset [30]. Specifically, the eight categories are: airplane, cabinet, car, chair, lamp, sofa, table, and vessel and the number of the training models is 30974. The complete point clouds are created by uniformly sampling from the complete mesh, and the partial point clouds are generated by back-projecting 2.5D depth images into 3D to simulate the real-world sensor data. We used the same train/valid/test split provided by the original PCN dataset.

TopNet dataset. The TopNet dataset [19] is composed of 28974 training models

and 800 validation samples. The key difference to the PCN dataset is the number of points of ground truth: PCN dataset contains 16384 points whereas TopNet is comprised of 2048 points. Since TopNet dataset does not provide the ground truth for test data, we used the provided validation set for testing and picked 600 samples from the training data to use it as a validation set.

KITTI dataset. The KITTI dataset, provided by [18], consists of the real-world LIDAR scans of 2401 cars. Unlike the previous two datasets, the partial point clouds in KITTI have no ground truth. Therefore we trained the networks with the car category in the ShapeNet and tested the performance in KITTI.

4.2 Implementation Details

We tested the performance of shape completion with and without the additional SAUM branch given the conventional encoder-decoder network. In all experiments we used the PointNet-based encoder that makes the best performances in [18] and [19]. We experimented on the four decoders to represent existing shape completion methods as baseline models: **FCAE** [14], **AtlasNet** [9], **PCN** [18], and **TopNet** [19]. We used $N = 2048$ for input point cloud by random subsampling/oversampling the points and designed all of the decoders to generate $M = 16384$ points for PCN dataset and $M = 2048$ points for TopNet dataset.

Our two-branch network is implemented by attaching our module to the baseline architectures mentioned above and jointly trained in an end-to-end manner. We chose the upsampling ratio r to be eight and two for the PCN dataset and the TopNet dataset respectively. Specifically, we set the $r_i = 2$ for $i = 1, \cdots, 4$ for PCN dataset and the $r_1 = r_2 = 1$ for TopNet dataset. The convolutional layers of the feature expansion ($C^1_{i,j}, C^2_{i,j}, C^3_{i,j}$ in Eq.(4)) are (256, 128, 3) for all experiements. For training the PCN [18], we followed the existing two-staged generation but attached SAUM only to the final output.

We implement the networks using TensorFlow and trained them on Nvidia RTX 2080 Ti and Titan RTX. All of the models are trained using Adam optimizer [31] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for up to 300K steps with a batch size 32. The learning rate was initially chosen to be 10^{-4} and decayed by 0.7 every 50K steps. The best model was chosen based on the reconstruction loss calculated in the validation set.

4.3 Evaluation Metrics

For the evaluation of the consistent completion, we used three metrics: Chamfer Distance (CD), Earth Mover's Distance (EMD), and F-Score. We used the Eq.(2) for calculating the CD. While CD is a widely used metric to compare a pair of point sets, it is limited to represent the shape fidelity as points scattered without the correct geometric details can achieve a lower value in sum. The EMD is more sensitive metric to capture the detailed shape similarity, and is defined as:

$$EMD(P_1, P_2) = \min_{\phi: P_1 \to P_2} \frac{1}{|P_1|} \sum_{x \in P_1} \|x - \phi(x)\|_2, \tag{6}$$

where $\phi(x)$ represents the bijection from P_1 to P_2.

To demonstrate that our suggested network indeed better captures the fine structure, we adopt the F-Score suggested by recent works [32,28] as a supplementary metric. F-Score is motivated by the IoU metric in object detection and is defined as

$$\text{F-Score(d)} = \frac{2 \cdot \text{precision}(d) \cdot \text{recall}(d)}{\text{precision}(d) + \text{recall}(d)}, \tag{7}$$

with

$$\text{precision}(d) = \frac{1}{N_{P_{out}}} \sum_{p \in P_{out}} \mathcal{I}[\min_{p' \in P_{gt}} \|p - p'\| < d] \tag{8}$$

and

$$\text{recall}(d) = \frac{1}{N_{P_{gt}}} \sum_{p' \in P_{gt}} \mathcal{I}[\min_{p \in P_{out}} \|p - p'\| < d] \tag{9}$$

where $\mathcal{I}[\cdot]$ represents an indicator function. Conceptually, precision represents the portion of the correct prediction of the reconstructed point cloud, and recall represents the reconstructed portion from the ground truth shape. The F-Score is high when both precision and recall are high.

For the KITTI dataset, which has no ground truth shape, we cannot apply the previous metrics. Instead, we adopt the Fidelity [18], which is the average distance from each point in the input to its nearest neighbor in the output for the validation of the local consistency.

5 Results

Table 1. Quantitative results on the three datasets. The average CD multiplied by 10^3, EMD multiplied by 10^2, F-Score and, Fidelity are reported. The lower the CD, EMD, and Fidelity, the better. The higher the F-Score, the better. Better results are in bold.

dataset	PCN			TopNet			KITTI
metric	CD	EMD	F-Score	CD	EMD	F-Score	Fidelity
FCAE	9.799	17.128	0.651	22.036	14.731	0.597	0.03305
FCAE+SAUM	**8.668**	**9.015**	**0.745**	**20.295**	**8.573**	**0.654**	**0.01931**
AtlasNet	9.739	18.295	0.669	21.903	10.751	0.612	0.03505
AtlasNet+SAUM	**8.725**	**8.436**	**0.747**	**20.649**	**8.004**	**0.654**	**0.01820**
PCN	9.636	8.714	0.695	21.600	10.319	0.620	0.03382
PCN+SAUM	**8.900**	**6.631**	**0.741**	**20.400**	**7.847**	**0.665**	**0.01822**
TopNet	9.637	12.206	0.668	21.700	10.813	0.612	0.03595
TopNet+SAUM	**8.316**	**10.471**	**0.756**	**20.305**	**7.827**	**0.666**	**0.01684**

The quantitative results of SAUM-attached point cloud completion are shown in the Table 1 for PCN, TopNet, and KITTI datasets. Note that, for the fair

comparison, we used farthest point sampling to sample the equal number of points for the reconstruction and the provided ground truth (16384 for PCN and 2048 for TopNet) in all of the following quantitative evaluations(Table 1, 2, 3) and visualizations(Figure 3, 4, 6) because our network doesn't generate same number of points with the ground truth. For the distance threshold d defined in the F-Score, we chose the value roughly around the mean Chamfer Distance, $d = 10 \times 10^{-3}$ for the PCN dataset and $d = 20 \times 10^{-3}$ for the TopNet dataset respectively. The ground truth shape does not exist for the KITTI dataset, and we evaluated the Fidelity which measures how well the input is preserved [18].

The results indicate that the attachment of SAUM boosts the performance of all existing decoders for all metrics and datasets. The improvement of EMD and F-Score, which are known to be more informative evaluation measure [14,32], is significant with our module.

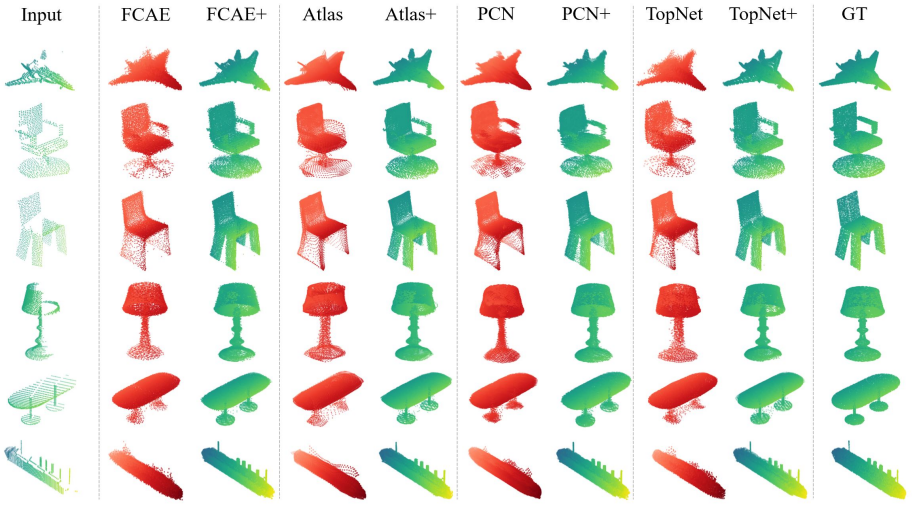

Fig. 3. Point cloud completion results of the various baselines and SAUM-attached models (post-fixed by +) on the PCN dataset.

Figure 3 visualizes shape completion results of various input categories. We can observe that the existing shape completion methods (shown in red) generate approximately similar shape, but fail to preserve fine details for all of the four implementations tested. For example, the airplane tails or decorative curves at the lamp are diluted, and the points are scattered around the empty space between armrests, table legs, and mast of the ship. When there exist narrow structures or holes, points are scattered around the region, filling the space which should have been empty or deleting fine details. This phenomenon is called as the blindness of CD metric, discussed in [14]. In contrast, the SAUM-attached models (shown in green) better catch detailed local information and suffer less from the

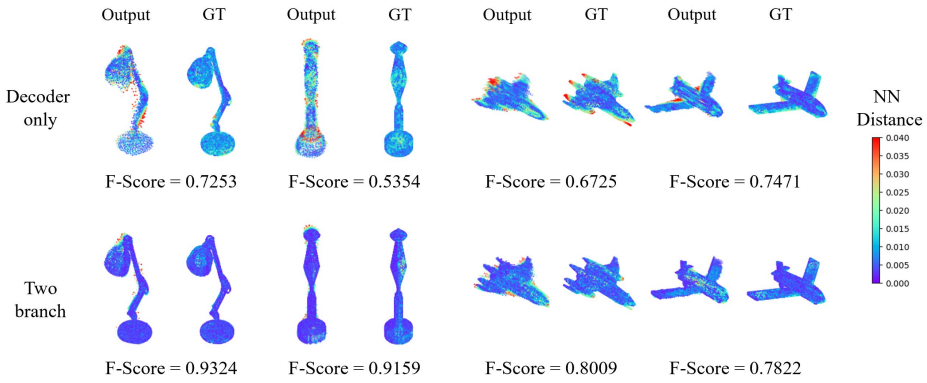

Fig. 4. Visualization of the output of neural networks and the ground truth. Each point is colored by the distance to the closest point in the other point cloud (nearest neighbor distance). The F-Score is high when both precision and recall are high, namely the ratio of blue points are high in the visualization for both of the output and the ground truth. TopNet was used for the decoder model for the lamps(left two columns) and PCN for the airplanes(right two columns).

problem of the blindness of CD metric, because SAUM increases the number of points capturing the local structure. Figure 4 compares the reconstruction results against the ground truth, and indicates that the SAUM-attached networks have greater shape fidelity (lower distance to the nearest neighbor (NN)) compared to the decoder-only networks. This suggests SAUM increases the precision and the recall, and in the end increases the F-Score.

Figure 5 depicts the complementary nature of the two branches. We show the shape completion of the individual branches before the set union in addition to the final output compared against the ground truth. The upsampling of SAUM (blue) complements the global shape acquired from the decoder (red) and mainly preserves the observed details. It is interesting that our network is only trained with the reconstruction loss defined by CD, and we did not explicitly enforce the structural prior, but SAUM clearly utilizes the geometric structure of the input shape, such as reflective and rotational symmetry as shown in airplane wings or chair legs. Note that feature expansion tends to generate points located near the original points in the upsampling task, but when applied to the completion task, it can also generate symmetric points without additional loss term. However, SAUM is not sufficient to generate global semantics of the underlying shape from partial data. Conventional decoders are trained to regress to complete shapes based on the semantic prior which compensate for the shortcomings of SAUM. In our implementation, the joint training of SAUM and a decoder specifically utilizes the output of the decoder to predict the residual of SAUM. As a result, the two branches benefit from the complementary nature and the decoder relieves the burden of predicting complete shapes from SAUM.

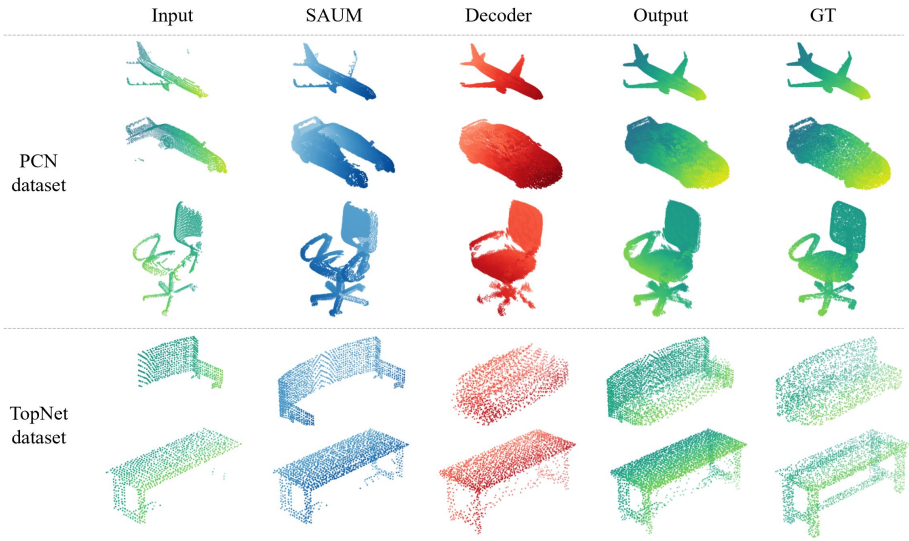

Fig. 5. Visualization of generated points from different branches, namely SAUM (blue) and decoder (red), and the union of them (green). PCN was used for the decoder.

Self-Consistency Test. Loss of detailed geometry with existing decoders is a prevalent phenomenon regardless of the amount of incompleteness. It is due to the last maxpooling layer[24] in the encoder limiting only a fixed number of critical points and therefore the expressive power of the GFV. Even with an input that has no missing region, existing methods suffer from the bottleneck phenomenon of the performance, namely output blurry shape as ever. On the other hand, SAUM whose architecture focuses the local features can preserve the input geometry.

Table 2. Self-consistency test on the PCN dataset. The average CD multiplied by 10^3, EMD multiplied by 10^2, F-Score for $d = 10 \times 10^{-3}$, and their improvement ratio compared to the cases of the partial input (Table 1) are reported. The lower the CD and EMD, the better and the higher the F-Score, the better. Better results are in bold.

Methods	CD	EMD	F-Score
FCAE	8.936 (8.81% ↓)	16.851 (1.62% ↓)	0.692 (6.30% ↑)
FCAE+SAUM	**6.610 (23.74% ↓)**	**3.523 (60.92% ↓)**	**0.869 (16.64% ↑)**
AtlasNet	8.801 (9.63% ↓)	17.457 (4.58% ↓)	0.718 (7.32% ↑)
AtlasNet+SAUM	**6.469 (25.86% ↓)**	**3.543 (58.00% ↓)**	**0.877 (17.40% ↑)**
PCN	8.743 (9.27% ↓)	8.231 (5.54% ↓)	0.730 (5.04% ↑)
PCN+SAUM	**6.483 (27.16% ↓)**	**3.717 (43.95% ↓)**	**0.868 (17.14% ↑)**
TopNet	8.754 (9.16% ↓)	11.653 (4.53% ↓)	0.709 (6.14% ↑)
TopNet+SAUM	**6.131 (26.27% ↓)**	**3.420 (67.34% ↓)**	**0.892 (17.99% ↑)**

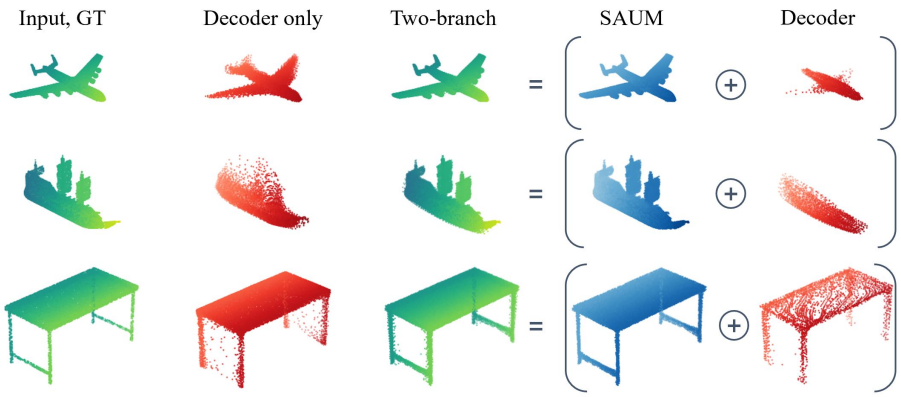

Fig. 6. Results of the self-consistency test on the PCN dataset. Note that the input is not partial shape but complete shape. TopNet was used for the decoder model for these visualizations.

Motivated by the observation, we propose a self-consistency test to evaluate the consistency of the networks by using the ground truth shape as the input and comparing it with the output. We experimented with the pre-trained networks of the existing methods and SAUM-attached models on the PCN dataset. Fig. 6 shows that the existing decoders cannot preserve the input geometry especially for the thin parts [18] while our SAUM not only keeps the fine details but also mitigates the decoder's hedging. Quantitative results on self-consistency test are reported in Table 2 with the same metric as shape completion: CD, EMD and F-Score. For each metric, we also reported the improvement ratio compared to the results of the original shape completion task(Table 1). The results show that the improvement ratios of SAUM-attached models are much greater than those of the decoder-only models. The augmentation of SAUM can go beyond the inherent limitation of the representation power for the conventional encoder-decoder architecture, help to resolve the bottleneck problem, and lead to consistent shape completion.

Ablation study. In our implementation, the base-line encoder uses encoder having four layers ($d = 4$) as suggested by previous works, and it is augmented with SAUM that is composed of the point expansion modules upsampling the points by the integer multiples of N given the intermediate features f_i ($i = 1, \cdots, d$). We change the attachment of the two-branch network and test with different upsampling ratio r and examine the quality of completed shape as an ablation study. The results using the PCN dataset are shown in Table 3. We first tested the performance of SAUM only without the encoder-decoder, and the performance is worse than most of the encoder-decoder baseline. Therefore, SAUM cannot generate plausible shape by itself, suggesting that the balance of SAUM and the existing decoders is important for consistent point cloud completion. As before, the attachment of SAUM enhances the quality of shape completion

Table 3. Ablation study on the PCN dataset. The average CD multiplied by 10^3, EMD multiplied by 10^2 and F-Score for $d = 10 \times 10^{-3}$ are reported. The lower the CD and EMD, the better and the higher the F-Score, the better. The best performance results are in bold.

Methods	CD	EMD	F-Score
SAUM (×4) only	10.928	33.164	0.589
SAUM (×8) only	**9.832**	**23.081**	**0.658**
FCAE	9.799	17.128	0.651
FCAE+SAUM (×4)	8.816	14.383	0.721
FCAE+SAUM (×8)	**8.668**	**9.015**	**0.745**
AtlasNet	9.739	18.295	0.669
AtlasNet+SAUM (×4)	8.756	13.477	0.732
AtlasNet+SAUM (×8)	**8.725**	**8.436**	**0.747**
PCN	9.636	8.714	0.695
PCN+SAUM (×4)	**8.898**	7.366	0.737
PCN+SAUM (×8)	8.900	**6.631**	**0.741**
TopNet	9.637	12.206	0.668
TopNet+SAUM (×4)	8.785	11.376	0.731
TopNet+SAUM (×8)	**8.316**	**10.471**	**0.756**

when applied jointly for all of the baseline networks. We also compare the effect of different upsampling ratio with $r = 4$ and $r = 8$. SAUM can complement the local consistency of the existing methods, improving the consistency with increasing r. The performance gain is more significant for EMD than CD, and we argue the EMD better captures the shape fidelity.

6 Conclusion

We propose SAUM, the symmetry-aware upsampling module that can be augmented to existing shape completion methods. Conventional shape completion methods are comprised of a decoder structure that generates unseen points based on semantic information condensed within a global feature vector and often fails to preserve local information. SAUM, on the other hand, utilizes the fine structure from the partial observation in addition to the latent symmetric structure. We suggest a two-branch architecture where the baseline decoder is attached with SAUM, which greatly improves the performance of the baseline. The two branches are complementary and in union generate globally consistent and completed shape, while maintaining the observed local structures.

Acknowledgements. This work was supported by the New Faculty Startup Fund from Seoul National University, KIST institutional program [Project No. 2E29450] and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2020R1C1C1008195).

References

1. Varley, J., DeChant, C., Richardson, A., Ruales, J., Allen, P.K.: Shape completion enabled robotic grasping. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017. (2017) 2442–2447
2. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J.J.: Past, Present, and Future of Simultaneous Localization and Mapping: Toward the robust-perception age. IEEE Trans. Robotics **32** (2016) 1309–1332
3. Dai, A., Qi, C.R., Nießner, M.: Shape completion using 3d-encoder-predictor cnns and shape synthesis. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. (2017) 6545–6554
4. Han, X., Li, Z., Huang, H., Kalogerakis, E., Yu, Y.: High-Resolution Shape Completion Using Deep Neural Networks for Global Structure and Local Geometry Inference. In: IEEE International Conference on Computer Vision, ICCV. (2017) 85–93
5. Stutz, D., Geiger, A.: Learning 3d Shape Completion From Laser Scan Data With Weak Supervision. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. (2018) 1955–1964
6. Mao, J., Wang, X., Li, H.: Interpolated Convolutional Networks for 3D Point Cloud Understanding. In: The IEEE International Conference on Computer Vision (ICCV). (2019)
7. Wang, Z., Lu, F.: VoxSegNet: Volumetric CNNs for Semantic Part Segmentation of 3D Shapes. IEEE transactions on visualization and computer graphics (2019)
8. Fan, H., Su, H., Guibas, L.J.: A Point Set Generation Network for 3D Object Reconstruction from a Single Image. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR. (2017) 2463–2471
9. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: A Papier-Mâché Approach to Learning 3D Surface Generation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018)
10. Yu, L., Li, X., Fu, C., Cohen-Or, D., Heng, P.: PU-Net: Point Cloud Upsampling Network. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR. (2018) 2790–2799
11. Yu, L., Li, X., Fu, C., Cohen-Or, D., Heng, P.: EC-Net: An Edge-Aware Point Set Consolidation Network. In: Computer Vision - ECCV 2018 - 15th European Conference, Proceedings, Part VII. (2018) 398–414
12. Wang, Y., Wu, S., Huang, H., Cohen-Or, D., Sorkine-Hornung, O.: Patch-Based Progressive 3D Point Set Upsampling. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR. (2019) 5958–5967
13. Li, R., Li, X., Fu, C.W., Cohen-Or, D., Heng, P.A.: PU-GAN: A Point Cloud Upsampling Adversarial Network. In: The IEEE International Conference on Computer Vision (ICCV). (2019)
14. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.J.: Learning Representations and Generative Models for 3D Point Clouds. In: Proceedings of the 35th International Conference on Machine Learning, ICML. (2018) 40–49
15. Yang, Y., Feng, C., Shen, Y., Tian, D.: FoldingNet: Point Cloud Auto-Encoder via Deep Grid Deformation. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR. (2018) 206–215

16. Yang, G., Huang, X., Hao, Z., Liu, M.Y., Belongie, S., Hariharan, B.: PointFlow: 3d Point Cloud Generation With Continuous Normalizing Flows. In: The IEEE International Conference on Computer Vision (ICCV). (2019)
17. Shu, D.W., Park, S.W., Kwon, J.: 3D Point Cloud Generative Adversarial Network Based on Tree Structured Graph Convolutions. In: The IEEE International Conference on Computer Vision (ICCV). (2019)
18. Yuan, W., Khot, T., Held, D., Mertz, C., Hebert, M.: PCN: Point Completion Network. In: Proceedings of 2018 International Conference on 3D Vision (3DV). (2018)
19. Tchapmi, L.P., Kosaraju, V., Rezatofighi, H., Reid, I., Savarese, S.: TopNet: Structural Point Cloud Decoder. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019)
20. Gurumurthy, S., Agrawal, S.: High Fidelity Semantic Shape Completion for Point Clouds Using Latent Optimization. In: IEEE Winter Conference on Applications of Computer Vision, WACV. (2019) 1099–1108
21. Sarmad, M., Lee, H.J., Kim, Y.M.: RL-GAN-Net: A Reinforcement Learning Agent Controlled GAN Network for Real-Time Point Cloud Shape Completion. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019)
22. Liu, M., Sheng, L., Yang, S., Shao, J., Hu, S.: Morphing and Sampling Network for Dense Point Cloud Completion. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, AAAI Press (2020) 11596–11603
23. Chen, X., Chen, B., Mitra, N.J.: Unpaired Point Cloud Completion on Real Scans using Adversarial Training. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net (2020)
24. Wang, X., , M.H.A.J., Lee, G.H.: Cascaded Refinement Network for Point Cloud Completion. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (2020)
25. Huang, Z., Yu, Y., Xu, J., Ni, F., Le, X.: PF-Net: Point Fractal Network for 3D Point Cloud Completion. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (2020)
26. Wen, X., Li, T., Han, Z., Liu, Y.S.: Point Cloud Completion by Skip-Attention Network With Hierarchical Folding. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (2020)
27. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Medical Image Computing and Computer-Assisted Intervention - MICCAI. (2015) 234–241
28. Xie, H., Yao, H., Zhou, S., Mao, J., Zhang, S., Sun, W.: GRNet: Gridding Residual Network for Dense Point Cloud Completion. CoRR **abs/2006.03761** (2020)
29. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR. (2017) 77–85
30. Chang, A.X., Funkhouser, T.A., Guibas, L.J., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: Shapenet: An information-rich 3d model repository. CoRR **abs/1512.03012** (2015)
31. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. In: 3rd International Conference on Learning Representations, ICLR. (2015)

32. Tatarchenko*, M., Richter*, S.R., Ranftl, R., Li, Z., Koltun, V., Brox, T.: What Do Single-view 3D Reconstruction Networks Learn? (2019)