ShapeFormer: A Transformer for Point Cloud Completion

Mukund Varma T[1], Kushan Raj[1], Dimple A Shajahan[1,2], M. Ramanathan[1]
[1]Indian Institute of Technology Madras, [2]TKM College of Engineering
{mukundvarmat, kushan5711}@gmail.com

Abstract

We present ShapeFormer, a pure transformer based architecture that efficiently predicts missing regions from partially complete input point clouds. Prior work for point cloud completion still produce samples of inferior visual quality, specifically near smooth regions, sharp corners, thin lines, etc. To solve these problems, we carefully design the encoder and decoder of ShapeFormer to - (1) encode the partial input point cloud using memory efficient Local Context Transformer, (2) predict missing regions from the overall shape representation using Folding Blocks, (3) guide the completion procedure using geometric cues present in the input partial shape using Skip Context Transformer, (4) and finally group points based on their semantic similarity into regions using the learnable Region Grouping layers. Our experiments demonstrate that ShapeFormer can accurately predict complete point clouds of high visual quality, and can achieve competitive results in the Completion3D benchmark and even outperform state-of-the-art methods in the Multi-View Partial Point Cloud benchmark (↓ 10% CD). We introduce Completion3D-C, a benchmark to evaluate robustness of various point cloud completion methods and ShapeFormer achieves best performance across various unseen transformations (↓ 14% CD on average). We also show that our method generalizes well to out-of-domain samples belonging to both seen and unseen categories. All results bring us one step closer to using transformers as a "universal modelling tool" for point clouds. Code will be made available after acceptance.

1. Introduction

Recent improvements in deep learning techniques along with abundant access to point cloud data [29, 5] has enabled great progress in 3D computer vision [17, 19]. However, raw point clouds captured by 3D scanners or depth cameras are often incomplete due to occlusions, light reflection, limited sensor resolution, etc [34]. Therefore, recovering complete point clouds from partial scans is a very important task. Point clouds are unstructured, unordered and a majority of the early methods transform 3D data to regular representations like images [21] and voxels [18]. However, these methods are limited by the number of views/resolution of voxels and require large amount of storage, compute. With the advent of PointNet [17], deep learning architectures are capable of directly operating on 3D coordinates and this has been extended for the point cloud completion task [34, 23]. The task of point cloud completion requires - 1. retention of the geometric properties present in the input partial point cloud, 2. predict missing portions based on the given input. Existing work derive a global shape representation which is then used to estimate the missing regions [34, 23]. However, the pooling operation leads to loss of information which cannot be recovered in the decoding stage. Follow-up works [14, 26, 15, 31] improve completion results by preserving geometric details using local features extracted from the input point cloud. However, they still tend to "average" unique shapes within a class and produce a common structure that can minimize loss against all the samples. More recent methods like - [12] predict the missing part of the point cloud instead of the whole object, [32, 16] use probabilistic modelling to learn partial-to-complete mapping and [30, 25] utilize transformers for better decoding quality. However, the completed point clouds still lack important geometric details especially around thin lines, sharp corners, etc (see Fig. 3). Another important challenge in point cloud completion is comparison with the ground truth. Existing similarity metrics include Chamfer Distance (CD), and Earth-Mover's Distance (EMD), each with their own advantages and disadvantages. CD is computationally efficient but fails to penalize regions with different density distribution as the ground truth point cloud while EMD is of $O(N^2)$ complexity and cannot be applied to dense point clouds due to memory bottleneck. Therefore, there is a strong requirement to introduce robust yet efficient metrics to evaluate the quality of point cloud completion.

Transformers have achieved great results across various tasks and domains [24, 9]. Attention in its core is a set operator - implying that it is invariant to permutation and cardinality of the input elements, which makes it ideal for point cloud representation. Recent works utilizing transformers for 3D vision have showcased great performance in classification and segmentation tasks [20, 35, 11] while there is very little work for shape completion. To this end, we propose ShapeFormer - a fully-attention encoder decoder model for point cloud shape completion. The encoder contains multiple Local Context Aggregation Transformers, followed by Fully Connected (FC) layers to generate a coarse complete point cloud. The coarse shape is then up-sampled to higher resolutions in multiple stages by - preserving geometric details of the partial shape via Skip Context Aggregation Transformers, predicting missing regions from the overall shape vector and refining the generated point cloud by encoding region-wise representations of semantically similar parts. Our method performs competitively with existing methods on the Completion3D [23] and even outperforms them on the Multi-View Partial Point Cloud [16] datasets. Next, we establish a detailed test bench - Completion3D-C (C stands for Corrupted) for robustness analysis of point cloud shape completion networks to various (unseen) synthetic, domain shifts. We further evaluate our method's generalization capacity to out-of-domain samples from both seen and unseen categories. Our key contributions can be summarized as follows:

1. We propose a pure transformer network - ShapeFormer that achieves more expressive and universal point cloud representation to accurately complete missing regions in partial point clouds with high visual quality.
2. We carefully design the architecture of ShapeFormer to extract information rich feature representations of the partial input using multiple *Local Context Transformers* with increasing receptive fields at deeper layers, and generate a complete shape - by predicting missing regions from the overall shape vector, transfer information from the encoded partial input to the decoder using *Skip Context Transformers*, and associate points based on part-wise similarity using *Region Grouping* layers. By optimizing the standard Chamfer loss along with newly introduced *Routing, Part* losses our method learns to generate complete point clouds from partial inputs.
3. We empirically demonstrate that ShapeFormer performs on-par when compared to other methods in the Completion3D benchmark and even out-performs state-of-the-art methods in the Multi-View Partial Point Cloud dataset by as much as 10% (relative). Owning to the discrepancies in standard metrics like Chamfer Distance, and Earth-Mover's Distance we briefly describe a perceptual similarity based metric - which we term *Learned Point Cloud Distance*.
4. We introduce a benchmark dataset - Completion3D-C to evaluate robustness of methods on unseen synthetic transforms, out-of-domain samples belonging to seen and unseen categories and ShapeFormer improves performance by as much as 14% (relative) when compared to existing state-of-the-art.

2. Related Works

Advances in Transformers. Transformer [24] using self-attention mechanism can effectively capture long-range correlation and exchange information globally among the inputs. It has demonstrated a remarkable performance in natural language processing [8, 7, 3] and many cross-disciplinary applications [13, 33, 37]. Recent advances have also successfully extended Transformer to computer vision tasks. [9] was the first work to employ a pure transformer architecture (ViT) for image classification. The follow-up works extend ViT to various classic vision tasks, such as object detection [4, 39, 36, 22], and segmentation [6, 27], video processing [38, 2]. More recently, transformers have shown effectiveness in point cloud processing, specifically for the tasks of classification [20, 35, 11], retrieval [20], and segmentation [35].

Point Cloud Completion. Point Cloud completion aims to generate a complete point cloud given an incomplete input. One of the first works, PCN [34] first generates a coarse point cloud, then up-sampled using folding operations. Follow-up works like, TopNet [23] utilize a tree-structured decoder, SA-Net [28] introduces connections between the encoder and decoder to preserve the input structure, while PF-Net [12] fuses multi-scale inputs and argues that predicting only missing regions can avoid geometric distortions. More recent methods incorporate architectural changes [14, 26, 15, 31], modify training schemes [32, 16] and constantly attempt to improve performance.

Transformers Meet Point Cloud Completion. Following the success of transformers in point cloud representation, there exists few recent works which attempt to extend these methods for point cloud completion. [30] models the generation of complete point clouds as the snowflake-like growth of points in 3D space, while [25] capture both local, global context using self and cross attention operations. These few works indicate the feasibility of transformers to model complex tasks in point cloud processing like point cloud completion.

3. Method

Given a partial point cloud (P) with N points, our goal is to predict the complete shape (C) of the same resolution. To this end, we propose ShapeFormer which derives point-wise representations to encode the shape information and predicts the complete shape in a hierarchical manner. In the following sections, we first introduce the preliminary of attention and then provide an overview of PCT, followed by network architecture details.

3.1. Preliminary: Attention

Attention was first proposed for NLP [24], where the goal is to focus on a subset of important words. Consequently,

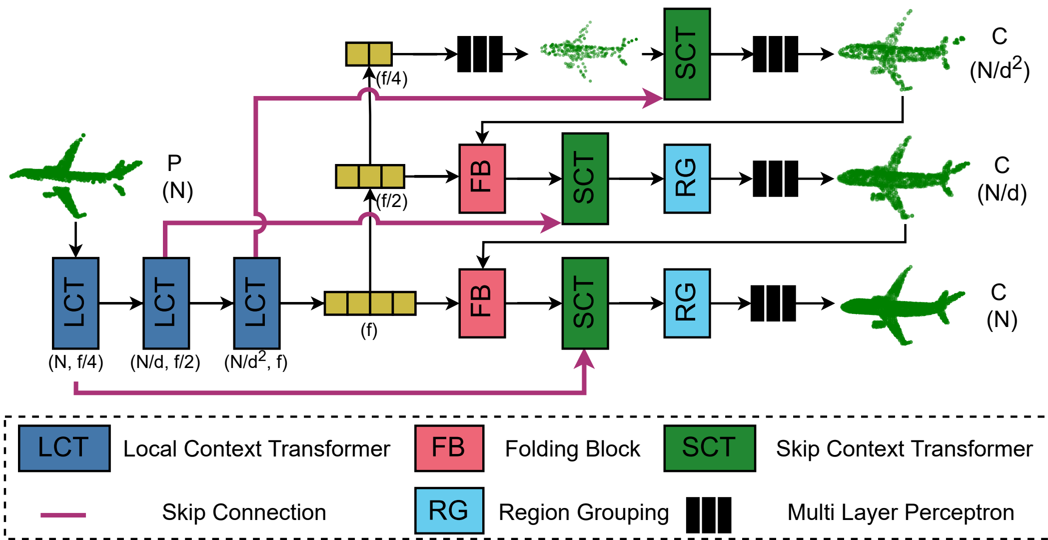


Figure 1: Overview of ShapeFormer: 1) encode partial input using LCT, 2) transfer encoder representations to decoder using SCT to guide generation process, 3) predict missing regions using FB, 4) refine predicted complete point cloud by grouping points into semantically similar regions using RG

relations between inputs are highlighted that can be used to capture context and higher-order dependencies. The attention matrix $A(\cdot)$ indicates a score between N_q Queries (Q) and N_k Keys (K), which indicates the highly correlated part that the input sequence needs to focus on.

$$A(Q, K) = \text{softmax}\left(\frac{Q \cdot K^T}{\gamma}\right) \tag{1}$$

where $\text{softmax}(\cdot)$ normalizes a matrix row-wise, and γ is called a temperature factor. To capture the relations between the input sequence, the Values V are weighted by the scores from Eqn. 1. Therefore, we have

$$\text{Attention}(Q, K, V) = A(Q, K) \cdot V, \tag{2}$$

The Transformer Attention, based on the Multi-Head Attention (MHA) operation is an extension of Eqn. 2. Rather than computing the attention once, the MHA operation computes it for H times (a.k.a. H-head attention). This helps the transformer jointly attend to different information derived from each head. The output from each of these heads are concatenated before projecting onto a final output dimension, followed by a residual connection with the input (Q) to the transformer. The overall operation can be summarized as:

$$\begin{aligned} \text{MHA} &= \text{concat}_{i=1}^{H}(\text{Attention}_i(Q, K, V)) \\ \text{Transformer} &= \text{Q} + \text{MLP}(\text{MHA}) \end{aligned} \tag{3}$$

3.2. Overview

As seen in Fig. 1, PCT takes a partial point cloud as input and derives point-wise representations by deriving context from its corresponding local neighbourhood. These point-wise representations are then pooled to derive an overall shape feature (F), then used to predict the complete shape in multiple stages hierarchically with increasing resolution. In each stage, we utilize three steps - (1) predict missing regions using the overall shape vector, (2) guide the completion process using skip connections to retain geometric details of the input partial point cloud, (3) concurrently refine the predicted missing and complete shapes by deriving region-wise vectors. Our network pipeline consists of the following stages:

Local Context Aggregation. Given an input partial point cloud, we first encode point-wise features by deriving context from the local neighbourhood of each point. This operation represented by the **Local Context Transformer** (LCT) in Fig. 1, learns relationships between each point and its K nearest neighbours. Similar to average pooling in standard convolution networks, we apply Farthest Point Sampling (FPS) by a factor d after each LCT block (except the last) to resemble increasing receptive fields at deeper layers.

Coarse-to-Fine Point Cloud Prediction. The point-wise representations derived from the final LCT block is mean-pooled to derive a global vector which represents the overall shape of the input partial point cloud. Inspired by [12], we predict the complete point cloud in multiple stages hierarchically in a coarse-to-fine manner. We first pass the derived shape vector through three MLP layers, each responsible to predict point clouds in different resolutions. The point cloud predicted in the top-most level act as center points to the next, where we predict displacement vectors (Δx, Δy,

Δz) to up-sample the complete shape to a higher resolution. This repeats, until a complete point cloud of the desired resolution is obtained (N in this case) and the overall structure is quite similar to a Feature Pyramid Network (FPN). The initial coarse point cloud is obtained from the global shape vector by passing it through an MLP layer, followed by a reshaping operation. This is further refined using the **Skip Context Transformer** (SCT), followed by an MLP layer before passing onto the next level.

Skip Context Aggregation. To ensure that the generated complete shape is coherent to the input partial point cloud, we utilize skip connections between the encoder and decoder at corresponding levels. This operation represented by **Skip Context Transformer** (SCT) in Fig. 1, learns relationships between the predicted complete shape and its closest similar neighbourhood in the partial input point cloud. This helps retain the local geometric details of the input partial shape in the predicted complete point cloud.

Global Upsampling. In the lower levels, we use a **Folding Block** [34] (FB) seen in Fig. 1 which utilizes the predicted complete shape from the previous level, global shape feature from the current level to derive displacement vectors. This operation learns to predict the missing regions from the global shape feature and up-sample the complete point cloud simultaneously.

Region Context Aggregation. A point cloud is characterized by multiple groups of points, together representing the complete shape. To effectively utilize these geometries and to refine the predicted point cloud, we derive region-wise features using the **Region Grouping** (RG) operation shown in Fig. 1. This also helps associate the missing points with the existing partial shape and induces more uniformity in the predicted complete point clouds.

3.3. Network Architectures

The key component of transformers is the attention block as we discussed in Sec. 3.1. Attention in its core is a set operator which makes it ideal for sequence modality tasks and hence we propose to use transformers as a key component in our method.

3.3.1 Local Context Transformer

The attention operation described in Sec. 3.1, computes relationships between every element in the input sequence which is computationally expensive in the context of point clouds (since N is at least > 1000) and extracts global information which is not necessarily useful for shape completion. The Local Context Transformer (shown in Fig. 2a), encodes the partial point cloud by aggregating information from each point's immediate neighbourhood. This is done by deriving the Q vector from the point-wise features, while the K, V vectors are derived from each point's K nearest neighbours. Unlike the standard dot product attention described in Eqn. 1, we compute the attention matrix using the subtraction relationship between Q, K vectors which does not collapse channel dimensions and enables it to be more expressive. The normalized attention matrix is then multiplied with the V vector and sum-pooled along the K axis to aggregate information from the neighbouring points. To encode spatial information, we project the distances between each center point and its neighbours to a higher dimension using an MLP layer, which is then added to the A matrix and V vectors. Each LCT block in Fig. 1 contains multiple attention layers, finally followed by a down-sampling operation (except the last block). To avoid loss of information in the down-sampling operation, we use FPS to derive the sampled point cloud, compute its nearest neighbours with respect to the previous resolution and aggregate the information using an MLP followed by a max-pooling operation.

3.3.2 Skip Context Transformer

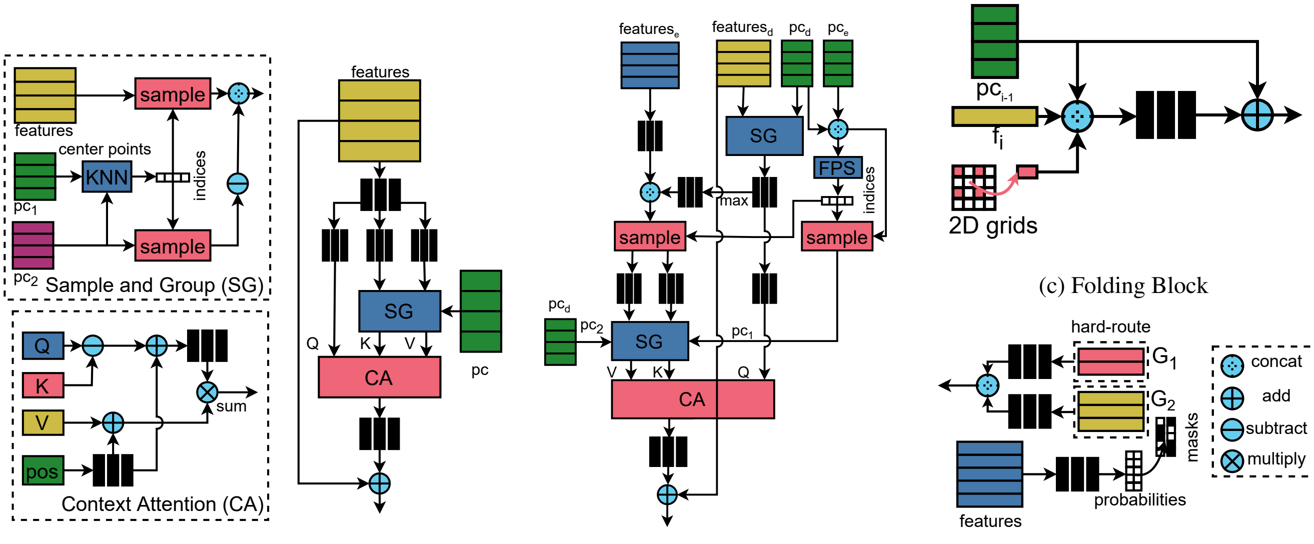
The Skip Context Transformer (shown in Fig. 2b), aggregates partial input shape representation from the encoder onto the predicted completed shape. To do so, we reuse the attention operation described in Sec. 3.3.1 and modify the inputs to derive Q, K, V vectors. The Q vector is derived using the features extracted from the predicted complete shape, while the K, V vectors are derived from each point's K nearest neighbours in the encoder partial shape. Since the predicted complete shape contains points that are not present in the encoder, the representations from the encoder and decoder (at the same level) are concatenated, and the FPS operation is performed to select a suitable set of point-wise representations covering the entire shape. These representations are used to derive the neighbourhood for each point in the complete shape. Simply put, this operation learns relationships between the predicted complete shape and the partial input which provide geometric cues to guide the completion process.

3.3.3 Folding Block

The Folding Block [34] (shown in Fig. 2c), globally up-samples the predicted complete point cloud using the overall shape representation and a base point cloud of lower resolution. In our network architecture, we repeat the predicted complete point cloud from the previous level to match the required final resolution, and concatenate 2D grids sampled from a 2D plane of fixed size. This representation is then concatenated with the global shape feature from the current level and transformed to displacement vectors using an MLP layer. Considering the points from the previous resolution as centers, we add these displacement vectors to generate an up-sampled point cloud.

3.3.4 Region Grouping

The Region Grouping Block (shown in Fig. 2d), leverages geometric cues present in various parts of the point cloud to refine the predicted complete shape. Unlike fixed grouping strategies, we employ a learnable MLP layer to predict



(a) Local Context Transformer (b) Skip Context Transformer (c) Folding Block (d) Region Grouping

Figure 2: Detailed Network Architectures of various components in ShapeFormer.

probability values for each point which determine the best group. The points are then hard-routed to these groups (i.e a point can belong to only one group and no residue of it is passed onto the others), followed by group-specific feature transformation learned using an MLP layer. Therefore, this operation helps associate various regions of the point cloud (i.e points belonging to the partial input and missing regions) based on their semantic similarity and further enhance them. The outputs across groups are concatenated to obtain the complete point-wise features which are transformed to 3D coordinates using an MLP layer. It is essential that the model explores all available groups before converging to the best group for a point. Hence at times during training, the points are routed to the group corresponding to their second-highest probability. Further to ensure that all points are not routed to a single group, we penalize the model based on the sum of squares of fraction of points routed to each group. To provide an intuition, let us assume N_1 and N_2 be the fraction of points routed to any two groups. If both these fraction of points are routed to a single group, then the value of routing loss will be higher as $(N_1 + N_2)^2 \geq N_1^2 + N_2^2$. While the loss might favour equal number of points being routed to each group, the model is still free to route any number of points and that this penalty just avoids it from getting biased to a selective number of groups.

3.4. Loss Functions

We train our network with an objective to minimize the Chamfer Distance [10] between the predicted complete point cloud and its corresponding ground truth at each resolution. This ensures that the predicted shape is consistent with the ground truth even at the least resolution and ensures that the later modules can learn to incorporate better details and refine the overall shape. However, Chamfer distance applied globally only ensures consistency in the overall shape and fails to penalize local geometric errors. Therefore, using the masks obtained via the Region Grouping block, we segment our regions from the predicted shape and the corresponding ground truth and apply chamfer distance locally. This further ensures that the model predicts fine details with greater accuracy. Our total loss L can be calculated as:

$$L = \lambda_{\text{chamfer}} \sum_{i=1}^{3} L_{\text{chamfer}}(C_i, GT_i) +$$

$$\lambda_{\text{part}} \sum_{i=1}^{G} L_{\text{chamfer}}(C_i \cdot M_i, GT_i \cdot M_i) + \lambda_{\text{reg}} \sum_{i=1}^{G} (N_i/N)^2 \tag{4}$$

where, C_i, GT_i represents the predicted, ground truth complete point clouds sampled at different resolutions, M_i, N_i represent the mask, number of points routed to each group respectively and corresponding weights for each loss indicated by λ.

4. Experiments

We conduct several experiments to compare Shape Completion Transformer against several state-of-the-art (SOTA) methods for shape completion. We first provide quantitative, qualitative results on benchmark datasets, followed by detailed robustness tests.

4.1. Implementation Details

We use the following hyper-parameters to train Point-Former for the shape completion tasks across different experiments. Our model consists of three LCT blocks in the encoder and similarly three levels in the decoder, each deriving point-wise representations of feature sizes 64, 128,

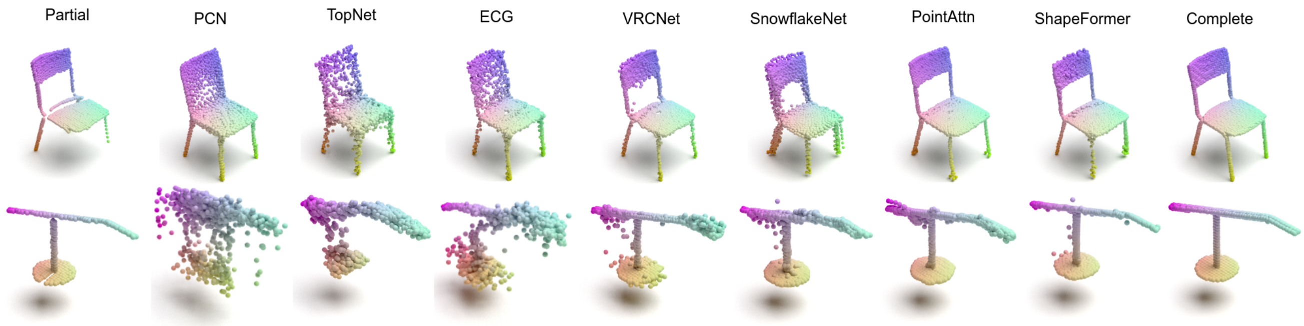


Figure 3: Qualitative results on samples from the Completion3D dataset. ShapeFormer can predict detailed structures (back of chair) and capture thin lines (lamp stand) more accurately than other methods.

256 respectively. Each transformer block - LCT, SCT contains two attention layers each with four heads. We choose d (down-sample factor) = 2, K (number of neighbours) = 64, G (number of groups) = 4. In the Folding blocks, we sample grids with higher density as the resolution gets bigger i.e at the later layers. We train the overall network for 100 epochs to minimize the loss, given in Eqn. 4 (with λ_{chamfer} = 1000, λ_{part} = 100, λ_{reg} = 1) using the Adam optimizer with a batch size of 32, initial learning rate 0.0001, which is cosine decayed while training.

4.2. Results

Metrics. To measure the performance of our model we use two widely adopted metrics: Chamfer Distance(CD) and Earth-Mover's Distance(EMD) [10]. We report the averages of each metric across all point clouds for a given dataset.

Method	CD ($\times 10^{-4}$) ↓
PCN	18.22
TopNet	14.25
SA-Net	11.22
GRNet	10.64
PMP-Net	9.23
VRCNet	8.12
SnowflakeNet	7.60
PointAttn	6.63
ShapeFormer	10.09

Table 1: Results on the Completion3D test set.

Results on Completion3D Dataset. The Completion3D dataset [23] contains 30,958 models belonging to 8 categories, where each point cloud contains 2,048 points. The dataset is split into 28,974 train, 800 validation and 1,184 test point clouds and we follow the exact same splits to ensure fair comparison with other methods. Table. 1 presents the CD on the test set obtained by submitting predictions to the evaluation server and we can clearly see that our model performs competitively when compared to SOTA methods.

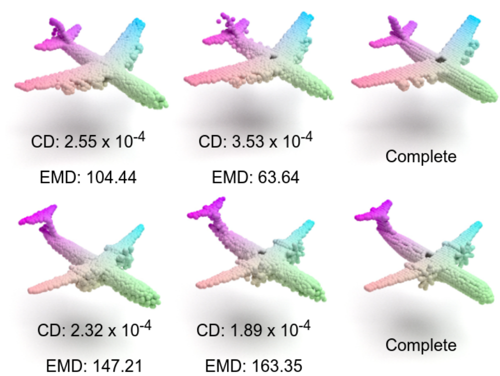


Figure 4: Disparity in the CD and EMD metrics with its correlation to visual quality.

As indicated by previous works [14], CD fails to penalize density variations, presence of noise, absence of fine geometric details, etc. Infact upon manual inspection, we identify several samples where ShapeFormer predicts complete point clouds of greater visual quality even when compared to methods like VRCNet, SnowflakeNet which have an overall lower chamfer distance (visualized in Fig. 3). EMD while being computationally more expensive than CD, is more locally discriminative and hence we compare the same on the Completion3D validation set (due to lack of access to GT point clouds in the test set). Surprisingly, we find that the performance of other baselines drop significantly on the validation set when compared to the test set while ShapeFormer outperforms all methods by CD and performs as well as PointAttn by EMD. Please note that we do not train the model on the validation set just like other methods. We attribute this difference in performance to various tricks employed by other baselines to achieve best performance on the test leaderboard (Eg: training the model to predict scaled down versions of the original point cloud as seen in SnowFlakeNet [1]). While most methods regard EMD as a better metric, we find several discrepancies between CD, EMD and its correlation to visual quality. For example, in Row-1 from Fig. 4, we can see that while EMD of the second point cloud is better than the first, it is relatively noisy and does not accurately construct the engine components while the vice versa is true in Row-2. To this end, we train a PointNet classifier on complete point clouds from the Completion3D train set and compute the L2-distance between the extracted features for the predicted, GT complete point clouds using the trained classifier. We term this metric **Learned PointCloud Distance** (LPS) and a smaller value indicates a higher semantic similarity between two point clouds. Table. 2 discusses these results and once again ShapeFormer outperforms most baselines and performs competitively when compared to PointAttn. A detailed analysis of the described metric does not fall within the scope of this paper and hence we leave this for future work.

Method	CD ($\times 10^{-4}$) $\downarrow$	EMD $\downarrow$	LPS ($\times 10^{-4}$) $\downarrow$
PCN	17.34	101.1	0.068
TopNet	22.16	105.72	0.116
ECG	19.52	129.65	0.082
VRCNet	15.57	115.96	0.064
SnowflakeNet	19.39	100.43	0.072
PointAttn	14.69	96.19	0.052
ShapeFormer	12.77	98.57	0.063

Table 2: Results on the Completion3D validation set.

Method	CD ($\times 10^{-4}$) $\downarrow$
PCN	9.77
TopNet	10.11
MSN	7.90
CRN	7.25
ECG	6.64
VRCNet	5.96
ShapeFormer	5.38

Table 3: Results on the MVP test set

Results on Multi-View Partial Point Cloud Dataset. Due to the smaller size of Completion3D dataset, we utilize the much larger Multi-View Partial Point Cloud (MVP) dataset [16] which contains over 100,000 models belonging to 16 categories. We choose the 2048 point resolution data subset for all our experiments due to computational limitations. Table. 3 discusses these results and ShapeFormer clearly outperforms existing SOTA methods by as much as 10% (relative). Fig. 8 provides qualitative results on samples from the MVP dataset and our method is able to produce accurate reconstructions of various regions. These results further indicate that our method showcases greater performance improvements when trained on much larger data which is consistent to similar observations in vision transformers [9].

Robustness results on Completion3D-C Benchmark. The datasets discussed above contain point clouds which are clean, noise-free and pose-normalized while in practice, one would expect a model to perform well on point cloud data that are transformed in several ways, unseen during training. To this end, we introduce a benchmark by extending the Completion3D dataset, called **Completion3D-C** (C for corrupted) and test various methods on their generalization capacity to unseen corruptions, domains and classes. As visualized in Fig. 5, the different corruptions in the benchmark include:

- **Noise**: Add random noise sampled from a normal distribution to each point.
- **Sparse**: Randomly set $\approx$ 88% of the 2048 points of each object to zero i.e., each object has only 256 valid points in this set.
- **Rotate**: Apply a random rotation on an arbitrary axis (z) to each object.
- **Perturb**: Rotate the object by a small magnitude along any axis.
- **Translation**: Each object is translated by a displacement vector sampled from a distribution.
- **Out-Of-Domain**: Identify samples belonging to the same categories from out-of-domain MVP dataset which are semantically farthest to the point clouds present in Completion3D. Semantic closeness is determined using extracted feature vector from a trained PointNet classifier.
- **Novel Category**: Identify samples belonging to unseen categories from MVP dataset (Eg: bed, bench, guitar, bus, etc).

Please note that we also transform the corresponding ground truths in the cases of Rotate, Translate, Perturb to ensure consistency with the modified partial input. We train a model only on the original dataset, and evaluate the same on the above mentioned unseen transformations. Table. 4 presents the CD between the predicted and ground truth complete point clouds and ShapeFormer outperforms existing methods by as much as 14% (relative). Specifically we see that our method is robust to geometric transformations like rotate, translate, perturb when compared to other methods. This further justifies the usefulness of our approach for the task of point cloud completion.

4.3. Discussion

Robustness to Real Point Cloud Scans. We further evaluate our method on real point cloud scans captured using lidars. Specifically, we select common categories present in the S3DIS [1] and benchmark MVP dataset (chair, table, sofa) and segment out the incomplete, corrupt point cloud instances from large-scenes. We then evaluate a model trained on the MVP dataset on these unseen scans. Due to absence of ground truth, we compare the accuracy of a PointNet classifier trained only on clean instances from the same categories before and after completion i.e an accurately completed point cloud should be classified correctly. Table. 5 compares the accuracy of the classifier on the corrupt input partial point clouds as-is, and completed point clouds using VRCNet, and ShapeFormer. Interestingly, we see that the accuracy of the classifier drops on the VRCNet-predicted complete point clouds when compared to the original partial input due to generation of noise and other artifacts while ShapeFormer helps improve performance by 1%.

[1] similar discussions can be found here

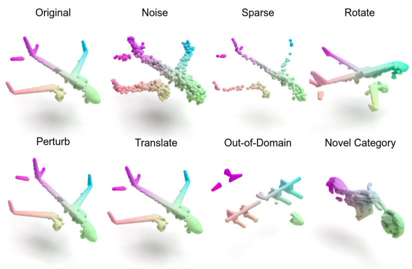


Figure 5: Samples from the Completion 3D-C benchmark.

Method	Original	Noise	Sparse	Rotate	Perturb	Translate	Out-Of-Domain	Novel Category	Average ($\times 10^{-4}$) $\downarrow$
PCN	17.34	19.34	28.64	52.93	21.7	43.85	10.76	14.52	26.13
TopNet	22.16	23.73	34.01	63.8	26.15	53.5	14.13	16.77	31.78
ECG	19.52	20.47	31.99	60.3	23.96	43.29	12.79	15.82	28.51
VRCNet	15.57	16.23	27.95	48.82	20.53	41.63	8.58	12.12	23.93
SnowflakeNet	19.39	24.93	30.67	38.93	20.53	33.81	7.06	8.74	23.12
PointAttn	14.69	16.64	25.77	47.7	17.58	36.31	6.59	8.35	21.70
ShapeFormer	12.77	23.09	28.2	28.2	14.06	23.99	8.57	11.31	18.77

Table 4: Results on the Completion3D-C benchmark for various unseen transformations.

Method	Accuracy (in %)
Input partial	9.87
VRCNet complete	8.91
ShapeFormer complete	10.01

Table 5: Point Cloud Classification accuracy on reconstructed point clouds from OOD Dataset.

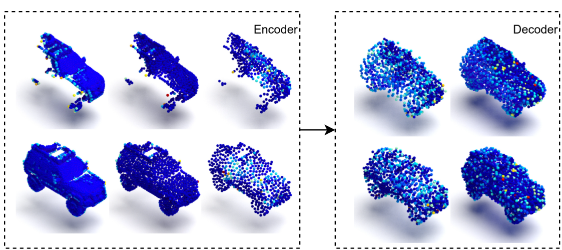


Figure 6: Visualization of learned attention maps in the encoder and decoder.

Visualizing Attention Maps. The learned attention maps compute relationships between the every point and its immediate neighbourhood. We visualize these attention maps by computing the most important point within this local neighbourhood and assign importance values based on the attention score. By adding these individual importance values for each point in the entire point cloud, we derive an average attention map describing the regions to which the model pays attention to. Fig. 6 visualizes the average attention across the entire point cloud and at different levels. We can see that the encoder initially pays uniform attention throughout the entire partial point cloud (higher near the incomplete regions) and more specific parts in the later layers. Similarly, the decoder pays attention to points throughout the shape at the lower resolution and once the predicted complete point cloud has sufficient details incorporated from the partial shape, it focuses more on the missing regions (last layer).

Visualizing Grouped Regions. In Fig. 7, we visualize the groups assigned to each point in the point cloud during the Region Grouping operation (where each color denotes a seperate group). Our model learns to group points into semantically similar regions, Eg: gun handle, gun barrel, airplane body, airplane wings, etc. Interestingly in the case of the airplane, our model routes symmetrically opposite but part-wise similar points to the same group. Therefore, beyond spatial similarity our model learns to group points based on geometric cues.

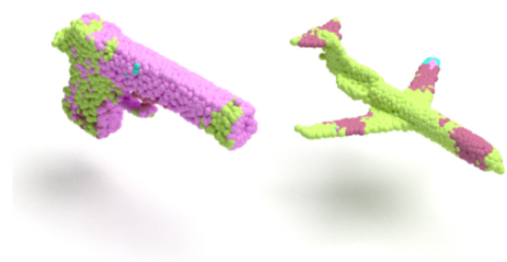


Figure 7: Figure represents the colour-coded groups created by ShapeFormer.

5. Conclusion

We propose a pure-transformer based approach for point cloud completion - ShapeFormer which efficiently generates complete point clouds by encoding neighbourhood contextual information, and guides the decoding process using skip connections. By learning to group points into semantically similar regions, our method is able to refine the predicted complete point clouds and further optimize chamfer distance at a part level. Our model predicts complete point clouds with fine geometric details, smooth distributions and even outperforms existing state-of-the-art methods in benchmark datasets. We highlight several disparities in existing evaluation metrics and briefly describe a perceptual similarity based metric - Learned PointCloud Distance. Further, we introduce a new robustness benchmark for point cloud completion - Completion3D-C to evaluate methods on unseen synthetic and domain transforms. Our model successfully generalizes to out-of-domain data belonging to seen and unseen categories.

References

[1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1534–1543, 2016. 7

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[3] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[5] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1

[6] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[7] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019. 2

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*, 2018. 2

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 1, 2, 7

[10] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 5, 6

[11] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, 2021. 1, 2

[12] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le. Pf-net: Point fractal network for 3d point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7662–7670, 2020. 1, 2, 3

[13] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 2021. 2

[14] Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and Shi-Min Hu. Morphing and sampling network for dense point cloud completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11596–11603, 2020. 1, 2, 6

[15] Liang Pan. Ecg: Edge-aware point cloud completion with graph convolution. *IEEE Robotics and Automation Letters*, 5(3):4392–4398, 2020. 1, 2

[16] Liang Pan, Xinyi Chen, Zhongang Cai, Junzhe Zhang, Haiyu Zhao, Shuai Yi, and Ziwei Liu. Variational relational point completion network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8524–8533, 2021. 1, 2, 7

[17] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1

[18] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2016. 1

[19] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 1

[20] Dimple A. Shajahan, Mukund Varma T., and Ramanathan Muthuganapathy. Point transformer for shape classification and retrieval of urban roof point clouds. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022. 1, 2

[21] Jong-Chyi Su, Matheus Gadelha, Rui Wang, and Subhransu Maji. A deeper look at 3d shape classifiers. In *Second Workshop on 3D Reconstruction Meets Semantics, ECCV*, 2018. 1

[22] Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris Kitani. Rethinking transformer-based set prediction for object detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2020. 2

[23] Lyne P Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 383–392, 2019. 1, 2, 6

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1, 2

[25] Jun Wang, Ying Cui, Dongyan Guo, Junxia Li, Qingshan Liu, and Chunhua Shen. Pointattn: You only need attention for

point cloud completion. *arXiv preprint arXiv:2203.08485*, 2022. 1, 2

[26] Xiaogang Wang, Marcelo H Ang Jr, and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 790–799, 2020. 1, 2

[27] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[28] Xin Wen, Tianyang Li, Zhizhong Han, and Yu-Shen Liu. Point cloud completion by skip-attention network with hierarchical folding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[29] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 1

[30] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5499–5509, 2021. 1, 2

[31] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. Grnet: Gridding residual network for dense point cloud completion. In *European Conference on Computer Vision*, pages 365–381. Springer, 2020. 1, 2

[32] Xia Yaqi, Xia Yan, Li Wei, Song Rui, Cao Kailang, and Stilla Uwe. Asfm-net: Asymmetrical siamese feature matching network for point completion. In *Proceedings of the 29th ACM International Conference on Multimedia*, 2021. 1, 2

[33] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34, 2021. 2

[34] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 International Conference on 3D Vision (3DV)*, pages 728–737. IEEE, 2018. 1, 2, 4

[35] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021. 1, 2

[36] Minghang Zheng, Peng Gao, Xiaogang Wang, Hongsheng Li, and Hao Dong. End-to-end object detection with adaptive clustering transformer. In *British Machine Vision Conference (BMVC)*, 2021. 2

[37] Wenqing Zheng, Qiangqiang Guo, Hao Yang, Peihao Wang, and Zhangyang Wang. Delayed propagation transformer: A universal computation engine towards practical control in cyber-physical systems. *Advances in Neural Information Processing Systems*, 34, 2021. 2

[38] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[39] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations (ICLR)*, 2021. 2

A. Appendix

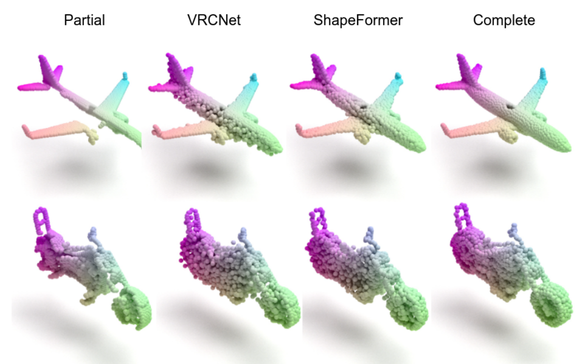


Figure 8: Qualitative results on samples from the MVP dataset. ShapeFormer predicts complete point clouds with higher uniformity and detail.