Orthogonal Dictionary Guided Shape Completion Network for Point Cloud

Pingping Cai, Deja Scott, Xiaoguang Li, Song Wang

University of South Carolina, USA
{pcai,ds17,xl22}@email.sc.edu, songwang@cec.sc.edu

Abstract

Point cloud shape completion, which aims to reconstruct the missing regions of the incomplete point clouds with plausible shapes, is an ill-posed and challenging task that benefits many downstream 3D applications. Prior approaches achieve this goal by employing a two-stage completion framework, generating a coarse yet complete seed point cloud through an encoder-decoder network, followed by refinement and upsampling. However, the encoded features suffer from information loss of the missing portion, leading to an inability of the decoder to reconstruct seed points with detailed geometric clues. To tackle this issue, we propose a novel Orthogonal Dictionary Guided Shape Completion Network (ODGNet). The proposed ODGNet consists of a Seed Generation U-Net, which leverages multi-level feature extraction and concatenation to significantly enhance the representation capability of seed points, and Orthogonal Dictionaries that can learn shape priors from training samples and thus compensate for the information loss of the missing portions during inference. Our design is simple but to the point, extensive experiment results indicate that the proposed method can reconstruct point clouds with more details and outperform previous state-of-the-art counterparts. The implementation code is available at https://github.com/corecai163/ODGNet.

Introduction

Point cloud is an efficient data structure for representing 3D objects in the form of a set of point coordinates. Despite its advantages, raw point clouds collected by existing 3D sensors often suffer from sparsity and incompleteness (Geiger et al. 2013), which significantly hinders their usability in downstream applications like autonomous driving (Zeng et al. 2018; Li et al. 2021), object detection (Zhou and Tuzel 2018; Shi and Rajkumar 2020), and segmentation (Zhang et al. 2023; Zhao et al. 2022). Therefore, inferring and reconstructing the missing regions of the incomplete point cloud is an inevitable and essential task in 3D computer vision. However, this point cloud completion task is extremely challenging. The successful reconstruction of correct shapes in the missing portions relies on a combination of high-level semantic understanding of the target object and low-level spatial and geometric relationships of nearby points. Moreover, this completion task is regarded as an ill-posed inverse problem. In other words, a single incomplete input can correspond to multiple plausible outputs, further complicating the inference of possible geometric details for the missing portion.

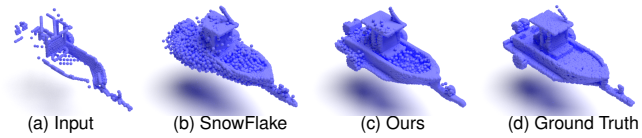

Figure 1: The point cloud completion results from different methods. We see that the previous method, SnowFlake (Xiang et al. 2021), cannot reconstruct the detailed shape for the missing portion, while our proposed method can infer a plausible shape.

Early traditional methods for solving this ill-posed problem relied on shape priors or hand-crafted geometric regularities (Kazhdan and Hoppe 2013; Lozes, Elmoataz, and Lézoray 2014; Hu, Fu, and Guo 2019; Pauly et al. 2008). However, these approaches have been overshadowed by deep learning-based methods. Previous state-of-the-art (SOTA) deep learning solutions follow the two-stage completion framework (Wenxiao Zhang 2020; Yan et al. 2022; Xiang et al. 2021, 2022; Pan et al. 2021; Yu et al. 2021; Zhou et al. 2022; Tang et al. 2022; Wang et al. 2022; Yu et al. 2023), where it first generates coarse but complete seed point clouds via an encoder-decoder network, and then employs an upsampling network to upsample and refine them. However, the encoded features derived from incomplete inputs represent only partial information and lack detailed geometric features for the missing parts. As a result, the seed points generated by the decoder may possess limited representation capability, which can potentially bottleneck the subsequent upsampling performance. Simply increasing the complexity of the upsampling network, as done by many previous works (e.g., SnowFlakeNet), might bring only limited benefit to the final performance, as illustrated in Figure 1, if the seed point clouds fail to adequately represent the underlying point-cloud shape.

Thus, in this paper, we present a "simple but straightforward" network, ODGNet, that mitigates the bottleneck observed in previous techniques and significantly improves

point cloud completion performance. Especially, the proposed ODGNet comprises two key components: the Seed Generation U-Net and the Dictionary Guidance Module. The Seed Generation U-Net effectively enhances the representation capability of generated seed points through multi-level feature extraction, concatenation, and the utilization of the local seed feature — a shape feature vector that captures the local geometry around each seed point. In parallel, the Dictionary Guidance Module plays a vital role by learning orthogonal shape priors from complete point clouds during supervised training and facilitating the recovery of better shapes during inference. Our key insight to mitigate shape information loss is the introduction of learnable shape dictionaries, enabling us to learn shape priors in the feature space. Furthermore, to ensure the shape dictionary captures distinguishable prior features effectively, we introduce additional orthogonal constraints to it. Lastly, we employ Upsample Transformers (Zhou et al. 2022) to upsample the seed points to the target resolution, further refining the completion results.

To verify the effectiveness of the proposed method, we evaluate it on three standard datasets: PCN (Yuan et al. 2018), ShapeNet-55/34 (Yu et al. 2021), and KITTI (Geiger et al. 2013). Experiment results show that the proposed method can recover detailed and plausible shapes for the missing portion. It can also achieve promising results and outperform previous SOTA counterparts easily. Our primary contributions can be summarized as follows:

1. We present a pioneering approach by introducing learnable shape priors to a deep learning architecture, effectively addressing the ill-posed completion task. This is achieved through the Dictionary Guidance Module, which compensates for missing geometric details
2. We design a simple yet forthright shape completion network built upon Seed Generation U-Net and the Dictionary Guidance Module to improve the representation ability of the seed points and upsampling performance.
3. We conduct comprehensive experiments on three datasets and the results confirm the effectiveness of the proposed algorithm by outperforming previous SOTA counterparts.

Related Work

Based on the network architecture of previous point cloud completion methods, we can classify them into two categories: Voxelization-based and Point-based methods.

Voxelization Based Method

Voxelization-based methods attempt to migrate solutions from 2D completion tasks to 3D point clouds by voxelization and 3D convolutions (Dai, Qi, and Nießner 2017; Wu et al. 2015; Han et al. 2017; Xie et al. 2020). To begin, Wu et al. (2015) introduces the 3D occupancy grid, which designates each voxel as a probabilistic distribution of binary variables to represent 3D shapes and uses Convolutional Deep Belief Networks to hallucinate the missing regions. However, the resolution of the 3D voxel grid is limited because of the high computational cost, making it challenging to reveal fine local geometric details. To improve the representation capability of the 3D occupancy grid, 3D-EPN (Dai, Qi, and Nießner 2017) encodes the implicit distance field functions into the 3D voxels and leverages high-level semantic features from a classification network to guide the shape completion process. In addition, GRNet (Xie et al. 2020) proposed a novel gridding process to improve the representation ability of 3D grids. Although voxelization-based methods can take advantage of 3D convolution to regularize unordered point clouds, these methods suffer from extensive computational costs or information loss during voxelization.

Point Based Method

Recently, with advancements in the network architectures designed for the point cloud (Qi et al. 2017a,b; Zhao et al. 2021), point-based methods have evolved into mainstream solutions for point cloud completion tasks and have achieved promising progress (Tchapmi et al. 2019; Pan 2020; Xie et al. 2020; Yuan et al. 2018; Pan et al. 2021; Xiang et al. 2021; Wang, Ang, and Lee 2022; Yu et al. 2021; Liu et al. 2020; Wen et al. 2021, 2022; Wenxiao Zhang 2020; Yan et al. 2022). For example, TopNet (Tchapmi et al. 2019) introduced a one-stage framework by modeling the point cloud generation process as the growth of a rooted tree, where one parent feature is split to generate multiple child features. The generated point features, however, lack accurate shape information of missing parts and cannot be constrained explicitly. It was then surpassed by the two-stage framework (Yan et al. 2022; Xiang et al. 2021, 2022; Pan et al. 2021; Yu et al. 2021; Zhou et al. 2022).

The two-stage completion framework can achieve better performance due to its ability to impose more constraints on the coarse-to-fine point cloud generation process. PCN (Yuan et al. 2018) is one of the pioneering works for the two-stage point completion framework, wherein the first stage uses PointNet (Qi et al. 2017a) layers to extract the global feature vector and MLPs to produce a coarse point cloud. The second stage uses a folding-based upsampling block (MLPs) to generate a dense and complete point cloud. However, the simple MLPs cannot fully exploit and preserve intricate geometric shapes, which limits the overall performance of PCN. Thus, SnowFlake (Xiang et al. 2021, 2022) introduces a novel snowflake point deconvolution block to upsample the points in the feature space and achieves promising performance. Comparatively, FBNet (Yan et al. 2022) and SeedFormer (Zhou et al. 2022) also focus more on the upsampling stage by introducing the Feedback-Aware Completion block and Upsample Transformers to refine and upsample the low-quality point cloud, respectively. However, these two-stage shape completion methods overlook the importance of the seed generation stage and limit their upsampling performance.

Proposed Method

Overview

Given an incomplete and sparse point cloud $P \in \mathrm{R}^{N_p * 3}$ as input, our objective is to infer its missing shapes and pro-

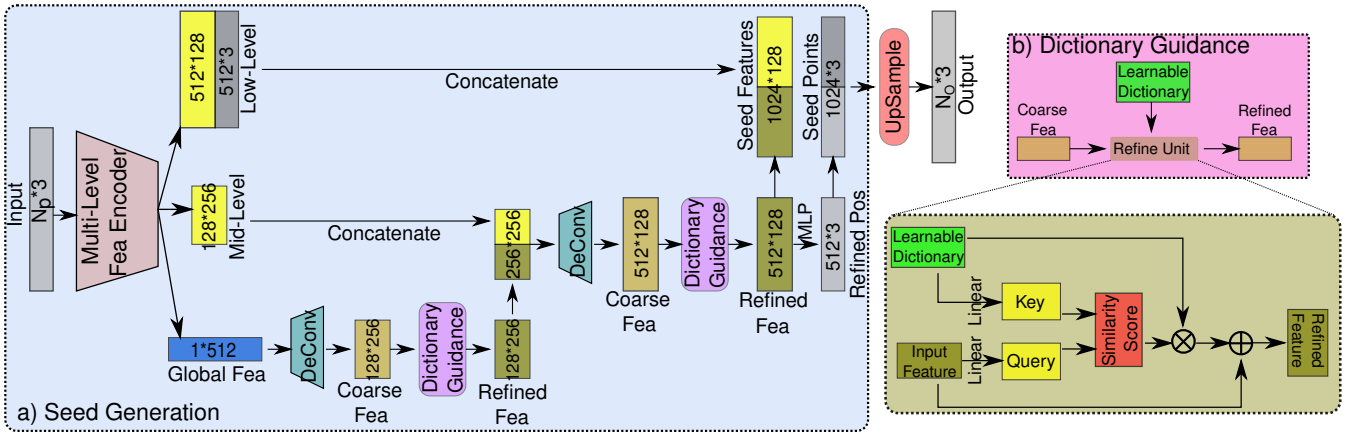

Figure 2: The overall architecture of the proposed network. a) The architecture for the Seed Generation U-Net. b) The detailed architecture for the Dictionary Guidance Module.

duce a complete and dense point cloud $O \in \mathrm{R}^{N_o*3}$. Following the two-stage point completion framework, we first design a seed generation U-Net to generate the coarse but complete seed points $S \in \mathrm{R}^{N_s*3}$ and then upsample them to the target resolution using Upsample Transformers (Zhou et al. 2022). Figure 2 shows the overall architecture design of the proposed ODGNet.

Seed Generation U-Net

Our primary focus is to enhance the representation capabilities of the seed point cloud. Given that downsampling operations in the encoder can lead to the loss of fine details, making it challenging to recover them during decoding, we draw inspiration from evolved 2D image processing techniques, such as U-Net (Ronneberger, Fischer, and Brox 2015). Especially, we adopt a similar approach by extracting multi-level features to preserve fine details at various resolutions and concatenate them with the decoder's output for better seed generation. Moreover, instead of merely generating coordinates of seed points, we introduce the concept of **seed features**, which are feature vectors representing rich local geometric details surrounding each seed point, to improve the representation capabilities. Figure 2a illustrates the overall architecture design of the Seed Generation U-Net.

Encoder The primary objective of the seed generation U-Net's encoder is to extract multi-level shape features from an incomplete point cloud P. To achieve this, we leverage the set abstraction module proposed in (Qi et al. 2017b), which facilitates gradual sub-sampling of points and extraction of local shape features at various resolutions.

In particular, for each level, we take the input point coordinates P_i and corresponding point features F_i, and model the output coordinates P_{i+1} and features F_{i+1} using the composition of two functions, expressed as follows:

$$P_{i+1}, F_{i+1} = \mathtt{PT}(\mathtt{SA}(P_i, F_i)) \tag{1}$$

Here, PT refers to the point transformer described in (Zhao et al. 2021), while SA represents the set abstraction module. Additionally, it's worth noting that $F_0 = P_0$ for the initial input. By stacking multiple set abstractions and point transformers together, we can extract point features and point coordinates at multiple levels, including the global shape feature, which represents the general shape information of the incomplete input.

Decoder The decoder module aims to generate the complete seed point cloud. Drawing inspiration from TopNet (Tchapmi et al. 2019), we adopt a similar approach to generate seed points in the feature space. This is accomplished by progressively splitting the input global shape code into multiple child features. Instead of employing multi-branch MLPs as done in TopNet, we utilize 1D deconvolution layers to generate these child features. Each child feature captures the local shape of the missing portion, and by stacking multiple 1D deconvolution layers with different kernel sizes and strides, we can effectively produce varying numbers of child features.

However, it is important to note that these child features only represent coarse shape information and may suffer from information loss. To overcome this limitation, we introduce the Dictionary Guidance Module, which plays a pivotal role in reconstructing detailed geometry and generating refined features from these coarse child features. Subsequently, the refined feature is concatenated with the multi-level features extracted from the encoder, resulting in complete shape features as illustrated in Figure 2a.

Finally, we employ an MLP layer to regress the refined point coordinates from the refined point features, completing the seed point cloud generation process.

Dictionary Guidance Module

Recall that another flaw in the seed generation process is the missing shape information from the incomplete input, which makes the point cloud completion task ill-posed and non-trivial. Presumably, without additional knowledge and guidance, the network can only generate ambiguous shapes for these missing portions.

To tackle this challenging problem, we introduce prior knowledge into the reconstruction process. The idea of our

solution is that we can learn the common shape features from the ground truth point clouds, *e.g*. features of airplanes in the training set, during the supervised training. Subsequently, during inference, the learned common features can be treated as strong priors to guide the shape completion, *e.g*. unseen airplanes in the testing set.

However, implementing this idea and seamlessly integrating it into the deep neural network is another challenge. Drawing inspiration from the Detection Transformer (DETR) (Carion et al. 2020), which leverages the learnable query (feature vector) to benefit object detection and bounding box generation, we take advantage of this learnable feature vector. We build a learnable dictionary that can learn the shape priors automatically from training samples. Figure 2b shows the network architecture of the Dictionary Guidance module. It contains a learnable dictionary and a Refine Unit to integrate shape priors into the coarse input features, ensuring the smooth guidance of the shape completion process.

Refine Unit With the input coarse point features $F \in \mathtt{R}^{N*C}$, our primary objective is to leverage additional shape information from the learnable dictionary $D \in \mathtt{R}^{N_d*C}$ and generate the refined point features $F' \in \mathtt{R}^{N*C}$. The implementation of our feature Refine Unit is designed to be straightforward and intuitive. Specifically, for the coarse features F, we first find their similar feature vectors in the learnable dictionary D. Then, we aggregate these similar feature vectors and integrate them with the coarse features, effectively compensating for any missing details.

Regarding our first step to calculate the similarity score $Sim \in \mathtt{R}^{N*N_d}$ between two feature tensors, we borrow the solution from the cross-attention mechanism:

$$Q = \phi(F); K = \psi(D); Sim = \sigma(\frac{QK^T}{\sqrt{d_k}}) \tag{2}$$

where ϕ and ψ are linear layers, σ is the Softmax function, d_k is dimension of K. Then we aggregate the related feature vectors in the dictionary using the predicted similarity scores, and the refined features can be obtained by:

$$F' = 0.5 * (MatMul(Sim, D) + F) \tag{3}$$

where $MatMul$ is the matrix multiplication and 0.5 is the coefficient to balance two components.

Othogonal Constraint Furthermore, as previously mentioned, to guarantee the representation ability of the learnable dictionary, we hope that each prior feature in the dictionary can be discriminative to others. To accomplish this, we introduce Orthogonal Constraints to each learnable dictionary $D \in \mathtt{R}^{N_d*C}$ so that each prior feature is orthogonal to others. Mathematically, this can be defined as follows:

$$\hat{D} = Normalize(D); L_{oth} = ||\hat{D}\hat{D}^T - I||_2^2, \tag{4}$$

where $I \in \mathtt{R}^{N_d \times N_d}$ is the Identity Matrix, N_d is the number of learnable vectors in the dictionary, and $N_d \leq C$.

Loss Function

Similarly to prior two-stage completion pipelines, we use the Chamfer Distance (CD) as a loss function to explicitly guide the Seed Generation and Upsampling processes. In particular, the CD loss is defined as follows:

$$CD = \frac{1}{N_1}\sum_{o \in O}\min_{g \in GT}||o-g||_2^2 + \frac{1}{N_2}\sum_{g \in GT}\min_{o \in O}||o-g||_2^2 \tag{5}$$

where O is the predicted completed point cloud with the number of N_1 points and GT is the ground truth point cloud with the number of N_2 points. Note that there are two variants for CD which we denote as CD-L2 and CD-L1. Specifically, CD-L2 is equal to CD while CD-L1 takes the square root of the L2-Norm and is divided by 2. To sum up, the total loss function used in training is defined as follows:

$$L = CD_{seed} + \lambda CD_{upsample} + \beta \sum L_{oth} \tag{6}$$

where $CD_{upsample}$ is the coarse to fine upsampling loss for Upsample Transformers, $\sum L_{oth}$ is the Orthogonal Constraints for all learnable dictionaries. λ and β are the hyperparameter to balance different terms and are set to 1 for all experiments.

Experiments

Dataset and Evaluation Metric

PCN: The PCN dataset is first introduced by Yuan et al. (2018) and contains pairs of partial and complete point clouds from 30,974 models of 8 categories collected from the ShapeNet (Chang et al. 2015). To maintain consistency with previous methods (Yuan et al. 2018; Xie et al. 2020; Xiang et al. 2021), we adopt the same train/test splitting strategy, comprising 28,974 training samples, 800 validation samples, and 1,200 testing samples. Additionally, in anticipation of the varying number of points for the incomplete point clouds, we follow prior works by resampling them to a standardized size of 2,048 points.

ShapeNet-55/34: The ShapeNet-55/34 datasets, introduced in PoinTr (Yu et al. 2021), are also derived from ShapeNet (Chang et al. 2015). ShapeNet-55 consists of 55 categories and comprises 41,952 training shapes and 10,518 testing shapes. On the other hand, ShapeNet-34 contains 46,765 shapes from 34 categories for training, and the testing set consists of 5,705 shapes, divided into two parts: 3,400 shapes from 34 seen categories and 2,305 shapes from 21 unseen classes. Following the previous works, we evaluate the models on the point cloud data with different missing point ratios of 25%, 50%, and 75%, representing three difficulty levels of completion tasks: simple (S), moderate (M), and hard (H), respectively.

KITTI: Since the previous two datasets are synthetic data generated from CAD models or meshes, which might be different from real scanned point clouds, we also include the KITTI dataset (Geiger et al. 2013). Essentially, it is collected from an autonomous driving platform and is a challenging real-world computer vision benchmark. We also follow the previous method by extracting a sequence of Velodyne scans from the KITTI dataset and only focusing on points within the object bounding boxes labeled as cars. In total, it has 2483 partial point clouds and no ground truth.

Methods	Average	Plane	Cabinet	Car	Chair	Lamp	Couch	Table	Watercraft
FoldingNet(Yang et al. 2018)	14.31	9.49	15.80	12.61	15.55	16.41	15.97	13.65	14.99
TopNet(Tchapmi et al. 2019)	12.15	7.61	13.31	10.90	13.82	14.44	14.78	11.22	11.12
AtlasNet(Groueix et al. 2018)	10.85	6.37	11.94	10.10	12.06	12.37	12.99	10.33	10.61
PCN(Yuan et al. 2018)	9.64	5.50	22.70	10.63	8.70	11.0	11.34	11.68	8.59
GR-Net(Xie et al. 2020)	8.83	6.45	10.37	9.45	9.41	7.96	10.51	8.44	8.04
PMP(Wen et al. 2021)	8.73	5.65	11.24	9.64	9.51	6.95	10.83	8.72	7.25
PoinTr(Yu et al. 2021)	8.38	4.75	10.47	8.68	9.39	7.75	10.93	7.78	7.29
NSFA(Wenxiao Zhang 2020)	8.06	4.76	10.18	8.63	8.53	7.03	10.53	7.35	7.48
SnowFlake(Xiang et al. 2021)	7.19	4.24	9.27	8.20	7.75	5.96	9.25	6.45	6.37
FBNet(Yan et al. 2022)	6.94	3.99	9.05	7.90	7.38	5.82	8.85	6.35	6.18
ProxyFormer(Li et al. 2023)	6.77	4.01	9.01	7.88	7.11	5.35	8.77	6.03	5.98
AdaPoinTr(Yu et al. 2023)	6.53	3.68	8.82	7.47	6.85	5.47	8.35	5.80	5.76
SeedFormer(Zhou et al. 2022)	6.74	3.85	9.05	8.06	7.06	5.21	8.85	6.05	5.85
Ours	**6.50**	**3.77**	**8.77**	**7.56**	**6.84**	**5.09**	**8.47**	**5.84**	**5.66**

Table 1: Point cloud completion results on the PCN dataset compared to previous algorithms (CD-L1 $\times 10^{-3}$).

Evaluation Metrics: To quantitatively evaluate the performance of different algorithms, we use three commonly adopted metrics: **CD-L1**, **CD-L2**, and **F1-Score@1%**. For the CD metric, the smaller value is better, while for the F1 score, the larger value is better. For the KITTI dataset, we use **Fidelity** and **Minimal Matching Distance (MMD)** since there is no ground truth. Specifically, Fidelity measures the average distance from each point in the input to its nearest neighbor in the output and MMD measures how much the output resembles a typical car by calculating the Chamfer Distance between the output and the car point cloud from ShapeNet that is closest to the output point cloud.

Evaluation on PCN Dataset

We evaluate the performance of the proposed network on the PCN dataset and compare it with previous methods. As the required output resolution for the PCN dataset is 16,384, we set the upsampling ratios of the upsampling module to $\{1, 4, 4\}$. Besides, we set the size of dictionaries to be equivalent to their input coarse feature dimension. To train the network from scratch, we set the total epochs to 400 with a batch size of 32 and use Adam as an optimization function with a learning rate of 0.0004 at the beginning and gradually decrease the learning rate by 0.8 for every 20 epochs. The training is carried out on two Nvidia V100 32G GPUs.

Table 1 presents the quantitative results of our proposed method, along with the reported outcomes from previous algorithms. Notably, our method achieves an outstanding average CD-L1 score of 6.50×10^{-3}, showcasing a significant improvement over the performance of prior methods. In particular, we demonstrate a remarkable advancement of 0.24×10^{-3} in comparison to the counterpart SeedFormer (Zhou et al. 2022). Figure 3 includes a visual representation of the PCN completion results. From the figure, it becomes evident that our proposed algorithm excels in preserving shape details for the missing parts, while minimizing the presence of noise points. In contrast, other algorithms may generate ambiguous shapes, often accompanied by a considerable number of outliers.

Method	CD-S	CD-M	CD-H	CD-Avg	F-Score
FoldingNet	2.67	2.66	4.05	3.12	0.082
PCN	1.94	1.96	4.08	2.66	0.133
TopNet	2.26	2.16	4.3	2.91	0.126
PFNet	3.83	3.87	7.97	5.22	0.339
GRNet	1.35	1.71	2.85	1.97	0.238
SnowFlake	0.7	1.06	1.96	1.24	0.398
PoinTr	0.58	0.88	1.79	1.09	0.464
ProxyFormer	0.49	0.75	1.55	0.93	0.483
AdaPoinTr	0.49	0.69	1.24	0.81	0.503
SeedFormer	0.5	0.77	1.49	0.92	0.472
Ours	**0.47**	**0.70**	**1.32**	**0.83**	0.437

Table 2: The quantitative results of different methods on the ShapeNet-55 benchmark dataset (CD-L2 $\times 10^{-3}$).

Evaluation on ShapeNet-55/34 Dataset

To showcase the robust generalization capability of our proposed method, we performed additional experiments on the ShapeNet-55/34 dataset. As this dataset requires an output resolution of 8,192, we adjusted the upsampling ratios of the upsampling module to $\{1, 2, 4\}$. We used the same optimization settings as in the ShapeNet-PCN dataset to train our network from scratch, but we gradually decrease the learning rate by half for every 50 epochs.

In Tables 2 and 3, we present the performance of our proposed method in comparison to previous algorithms. Impressively, our method achieves an average CD-L2 score of 0.83×10^{-3} on the ShapeNet-55 dataset, demonstrating a remarkable advancement over the performance of previous counterparts. Furthermore, even when dealing with the most challenging ShapeNet-34 seen and ShapeNet-21 unseen dataset, our method continues to outperform previous counterparts. Note that limited by space, detailed results can be found in the Supplementary.

Evaluation on KITTI Dataset

Finally, we examine the robustness of the proposed algorithm on the KITTI dataset. As the KITTI dataset con-

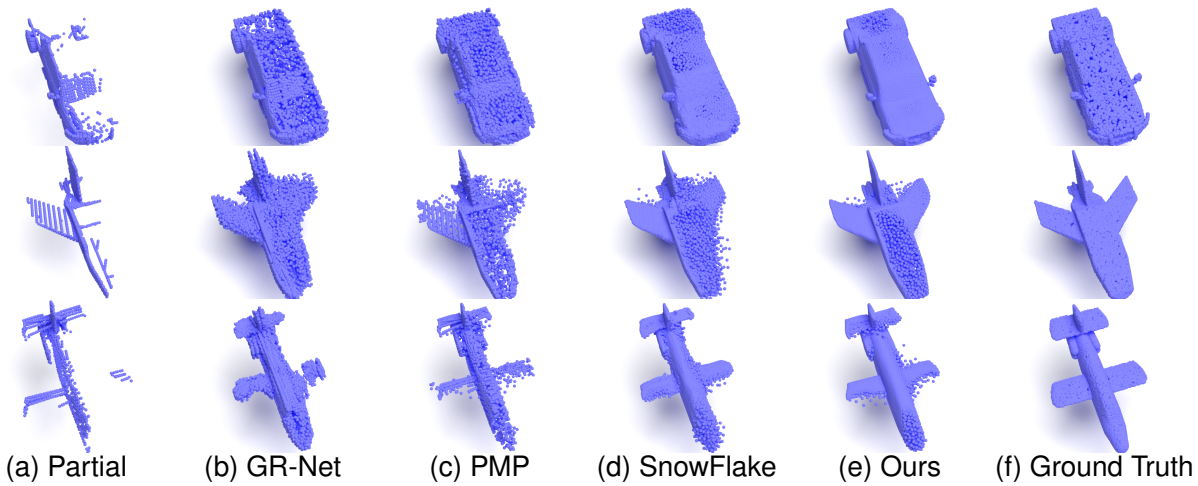

Figure 3: The completion results of various methods on the PCN dataset. Notably, our method can reconstruct the missing details, *e.g*. rearview mirror, better than others. Please zoom in for more details.

	Seen ShapeNet-34					Unseen ShapeNet-21				
Method	CD-S	CD-M	CD-H	CD-Avg	F-Score	CD-S	CD-M	CD-H	CD-Avg	F-Score
FoldingNet	1.86	1.81	3.38	2.35	0.139	2.76	2.74	5.36	3.62	0.095
PCN	1.87	1.81	2.97	2.22	0.154	3.17	3.08	5.29	3.85	0.101
TopNet	1.77	1.61	3.54	2.31	0.171	2.62	2.43	5.44	3.5	0.121
PFNet	3.16	3.19	7.71	4.68	0.347	5.29	5.87	13.33	8.16	0.322
GRNet	1.26	1.39	2.57	1.74	0.251	1.85	2.25	4.87	2.99	0.216
PoinTr	0.76	1.05	1.88	1.23	0.421	1.04	1.67	3.44	2.05	0.384
SnowFlake	0.6	0.86	1.5	0.99	0.422	0.88	1.46	2.92	1.75	0.388
ProxyFormer	0.44	0.67	1.33	0.81	0.466	0.60	1.13	2.54	1.42	0.415
AdaPoinTr	0.48	0.63	1.07	0.73	0.469	0.61	0.96	2.11	1.23	0.416
SeedFormer	0.48	0.7	1.3	0.83	0.452	0.61	1.07	2.35	1.34	0.402
Ours	**0.44**	**0.64**	**1.14**	**0.75**	0.451	**0.59**	**1.01**	**2.26**	**1.29**	0.415

Table 3: Shape completion results on Seen ShapeNet-34 test set and Unseen ShapeNet-21 test set (CD-L2 $\times 10^{-3}$).

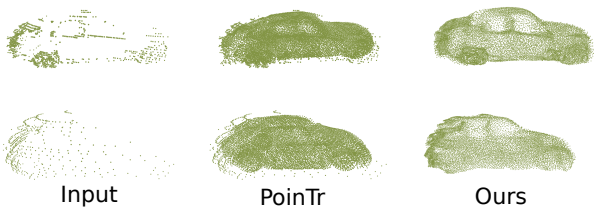

Figure 4: The visual comparison of different methods on the KITTI dataset. Our results are cleaner than PoinTr (Yu et al. 2021).

Method	FD ($\times 10^{-3}$)	MMD ($\times 10^{-3}$)
PCN	2.235	1.366
TopNet	5.354	0.636
GR-Net	0.816	0.568
PoinTr	0.00	0.526
ProxyFormer	0.00	0.508
AdaPoinTr	0.237	0.392
SeedFormer	0.151	0.516
Ours	1.28	**0.349**

Table 4: The evaluation results on the KITTI dataset.

tains only real-collected Lidar point clouds, we do not have ground-truth point clouds for training. Instead, we train our model on the PCN car dataset and test it on the KITTI dataset. Correspondingly, we use Fidelity and MMD to measure the performance. Please note that, since there is no ground truth, these metrics are not accurate measurements for the quality of generated point clouds. Table 4 shows the quantitative completion results, and Figure 4 shows some visual examples. We see that the previous method PoinTr (Yu et al. 2021) tends to generate outlier points, while our method can generate cleaner point clouds.

Ablation Study

Effectiveness of Seed Generation U-Net To prove that the proposed seed generation method can bring clean and significant performance improvements to the entire point cloud completion system, our first and foremost ablation

Seed Generation	Upsampling	CD-L1 ($\times 10^{-3}$)
FoldingNet	Folding	14.31
Ours	Folding	**7.59**
SnowFlake	PSCU	7.04
Ours	PSCU	**6.80**
SeedFormer	UpTrans	6.74
Ours	UpTrans	**6.50**

Table 5: Abaltion study on different seed generation methods and upsampling methods on the PCN dataset. PSCU means the Parametric Surface Constrained Upsampler (Cai et al. 2023). UpTrans means the Upsample Transformer in (Zhou et al. 2022). Visualization of generated seeds can be found in the Supplementary.

Dictionary Guidance	W/O	With	With
Orthogonal Constraints		W/O	With
CD-L1 ($\times 10^{-3}$)	6.62	6.55	6.50

Table 6: Ablation study on the Dictionary Guidance Module on the PCN dataset.

study is to determine the effectiveness of the proposed ODGNet in generating better seed points compared to other seed generation methods. We integrate the proposed Seed Generation U-Net into various upsampling methods and Table 5 shows the improvements. Remarkably, the use of our Seed Generation U-Net with the UpTrans demonstrates significant and direct improvement over the SeedFormer's seed generation method (Zhou et al. 2022), where the CD-L1 decreases significantly from 6.74×10^{-3} to 6.50×10^{-3}, showcasing a relative enhancement of 3.7%. Similar observations hold true for the remaining sections of the table. This insightful ablation study provides compelling evidence that the seed points generated by our method effectively preserve more intricate shape information, which in turn greatly benefits the upsampling modules and contributes to the overall performance improvement.

Dictionary Guidance Module Furthermore, we investigate the importance of the Dictionary Guidance Module, which aims to compensate for missing detailed shape information. To achieve this, we remove the Dictionary Guidance modules and substitute them with MLPs so that they have similar parameter amounts. As depicted in Table 6, the optimal performance is attained when the Dictionary Guidance module and orthogonal constraints are applied. Especially, we observe a notable performance drop when the Dictionary Guidance module is removed, where the CD-L1 increases from 6.50×10^{-3} to 6.62×10^{-3}, presenting a substantial gap of 0.12×10^{-3}, which strongly validate the effectiveness of the proposed Dictionary Guidance module in enhancing the completion system's overall performance.

Analysis of the Learnable Dictionary In the previous ablation study, we illustrated the importance of the Dictionary Guidance module. Nevertheless, there still exists some curiosity about the meaning of the shape vectors in the dictionary proposed in this paper.

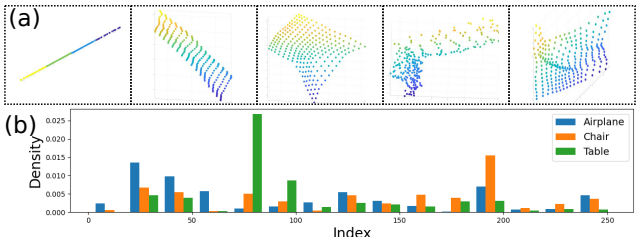

Figure 5: a) Visualization of reconstructed shapes from vectors in the learned dictionary. b) The density of the index of the maximum similarity scores in the learned dictionaries for 3 example classes.

To obtain a deep insight into this learnable dictionary, we incorporate the proposed seed generation backbone with a Folding-based upsampler (Yang et al. 2018). The Folding-based upsampler leverages seed features and predefined 2D grids to generate 3D points, enabling visualization of the shape features. After we trained this network on the PCN dataset, during the inference stage, we feed the shape vectors from the dictionary along with predefined 2D grids (16x16) into the Folding-based upsampler to showcase the shapes of learned vectors. The results are illustrated in Figure 5a. Since these shape vectors learn high-level shape features from training samples, they are not designed to represent real object parts, like desk corners or car wheels. Instead, we observe that each shape vector represents a distinct shape such as lines, planes, and curves, carrying a strong geometric meaning, which can be regarded as priors and fundamental building blocks for reconstructing missing components in the testing stage.

Furthermore, since the learned dictionary contains the shape priors extracted from the training samples, Intuitively, given different categories of point clouds, *e.g.* , airplane and table, each category of point cloud should utilize different shape priors to compensate for the missing details as they have distinct geometries. To verify this, we record the index of the maximum similarity scores in the Refine Unit and plot their density distributions. Figure 5b shows that the distribution of three categories in the PCN dataset has distinct patterns, which indicates that our method can automatically select the best combinations to reconstruct more details for the missing portion and achieve better performance.

Conclusion

In this paper, we propose the ODGNet, a simple but effective point cloud completion network, that aims to mitigate the bottlenecks of the two-stage framework and especially focuses on the first stage. It incorporates the newly designed learnable shape dictionaries to recover the fine-detailed shape information for the missing portions and multi-level feature extraction and concatenation to improve the representation ability of seed points. Without ornamentation, the experiment results show that our algorithm can efficiently reconstruct the missing portion with rich details and outperform previous state-of-the-art counterparts.

Acknowledgements

We sincerely thank the Senior Program Committee members and reviewers for their comments and contributions to the community. This work was supported, in part, by NEH PR-284350-22. The GPU used in this work was provided by the NSF MRI-2018966.

References

Cai, P.; Wu, Z.; Wu, X.; and Wang, S. 2023. Parametric Surface Constrained Upsampler Network for Point Cloud. In *AAAI conference on artificial intelligence*.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*.

Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; Xiao, J.; Yi, L.; and Yu, F. 2015. ShapeNet: An Information-Rich 3D Model Repository. arXiv:1512.03012.

Dai, A.; Qi, C. R.; and Nießner, M. 2017. Shape Completion using 3D-Encoder-Predictor CNNs and Shape Synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets Robotics: The KITTI Dataset. *International Journal of Robotics Research (IJRR)*, 32(11): 1231–1237.

Groueix, T.; Fisher, M.; Kim, V. G.; Russell, B.; and Aubry, M. 2018. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Han, X.; Li, Z.; Huang, H.; Kalogerakis, E.; and Yu, Y. 2017. High-resolution shape completion using deep neural networks for global structure and local geometry inference. In *IEEE International Conference on Computer Vision (ICCV)*.

Hu, W.; Fu, Z.; and Guo, Z. 2019. Local frequency interpretation and non-local self-similarity on graph for point cloud inpainting. *IEEE Transactions on Image Processing*, 28(8).

Kazhdan, M.; and Hoppe, H. 2013. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3).

Li, S.; Gao, P.; Tan, X.; and Wei, M. 2023. ProxyFormer: Proxy Alignment Assisted Point Cloud Completion With Missing Part Sensitive Transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9466–9475.

Li, Y.; Ma, L.; Zhong, Z.; Liu, F.; Chapman, M. A.; Cao, D.; and Li, J. 2021. Deep Learning for LiDAR Point Clouds in Autonomous Driving: A Review. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8): 3412–3432.

Liu, M.; Sheng, L.; Yang, S.; Shao, J.; and Hu, S.-M. 2020. Morphing and sampling network for dense point cloud completion. In *AAAI conference on artificial intelligence*.

Lozes, F.; Elmoataz, A.; and Lézoray, O. 2014. Partial difference operators on weighted graphs for image processing on surfaces and point clouds. *IEEE Transactions on Image Processing*, 23(9).

Pan, L. 2020. ECG: Edge-aware point cloud completion with graph convolution. *IEEE Robotics and Automation Letters*, 5(3).

Pan, L.; Chen, X.; Cai, Z.; Zhang, J.; Zhao, H.; Yi, S.; and Liu, Z. 2021. Variational Relational Point Completion Network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Pauly, M.; Mitra, N. J.; Wallner, J.; Pottmann, H.; and Guibas, L. J. 2008. Discovering structural regularity in 3D geometry. In *ACM SIGGRAPH*. ACM.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*.

Shi, W.; and Rajkumar, R. 2020. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tang, J.; Gong, Z.; Yi, R.; Xie, Y.; and Ma, L. 2022. LAKeNet: Topology-Aware Point Cloud Completion by Localizing Aligned Keypoints. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1726–1735.

Tchapmi, L. P.; Kosaraju, V.; Rezatofighi, H.; Reid, I.; and Savarese, S. 2019. TopNet: Structural Point Cloud Decoder. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, X.; Ang, M. H.; and Lee, G. 2022. Cascaded Refinement Network for Point Cloud Completion with Self-supervision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11): 8139–8150.

Wang, Y.; Tan, D. J.; Navab, N.; and Tombari, F. 2022. Learning Local Displacements for Point Cloud Completion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1568–1577.

Wen, X.; Xiang, P.; Han, Z.; Cao, Y.-P.; Wan, P.; Zheng, W.; and Liu, Y.-S. 2021. PMP-Net: Point cloud completion by learning multi-step point moving paths. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wen, X.; Xiang, P.; Han, Z.; Cao, Y.-P.; Wan, P.; Zheng, W.; and Liu, Y.-S. 2022. PMP-Net++: Point cloud completion by transformer-enhanced multi-step point moving paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1): 852–867.

Wenxiao Zhang, C. X., Qingan Yan. 2020. Detail Preserved Point Cloud Completion via Separated Feature Aggregation. In *European Conference on Computer Vision (ECCV)*.

Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3d shapenets: A deep representation for volumetric shapes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xiang, P.; Wen, X.; Liu, Y.-S.; Cao, Y.-P.; Wan, P.; Zheng, W.; and Han, Z. 2021. SnowflakeNet: Point Cloud Completion by Snowflake Point Deconvolution with Skip-Transformer. In *IEEE International Conference on Computer Vision (ICCV)*.

Xiang, P.; Wen, X.; Liu, Y.-S.; Cao, Y.-P.; Wan, P.; Zheng, W.; and Han, Z. 2022. Snowflake Point Deconvolution for Point Cloud Completion and Generation with Skip-Transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–18.

Xie, H.; Yao, H.; Zhou, S.; Mao, J.; Zhang, S.; and Sun, W. 2020. GRNet: Gridding Residual Network for Dense Point Cloud Completion. In *European Conference on Computer Vision (ECCV)*.

Yan, X.; Yan, H.; Wang, J.; Du, H.; Wu, Z.; Xie, D.; Pu, S.; and Lu, L. 2022. FBNet: Feedback Network For Point Cloud Completion. In *European Conference on Computer Vision (ECCV)*.

Yang, Y.; Feng, C.; Shen, Y.; and Tian, D. 2018. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yu, X.; Rao, Y.; Wang, Z.; Liu, Z.; Lu, J.; and Zhou, J. 2021. Pointr: Diverse point cloud completion with geometry-aware transformers. In *IEEE International Conference on Computer Vision (ICCV)*, 12498–12507.

Yu, X.; Rao, Y.; Wang, Z.; Lu, J.; and Zhou, J. 2023. AdaPoinTr: Diverse Point Cloud Completion with Adaptive Geometry-Aware Transformers. arXiv:2301.04545.

Yuan, W.; Khot, T.; Held, D.; Mertz, C.; and Hebert, M. 2018. PCN: Point Completion Network. In *International Conference on 3D Vision (3DV)*.

Zeng, Y.; Hu, Y.; Liu, S.; Ye, J.; Han, Y.; Li, X.; and Sun, N. 2018. RT3D: Real-Time 3-D Vehicle Detection in LiDAR Point Cloud for Autonomous Driving. *IEEE Robotics and Automation Letters*, 3(4): 3434–3440.

Zhang, C.; Wu, Z.; Wu, X.; Zhao, Z.; and Wang, S. 2023. Few-Shot 3D Point Cloud Semantic Segmentation via Stratified Class-Specific Attention Based Transformer Network. In *AAAI conference on artificial intelligence*.

Zhao, H.; Jiang, L.; Jia, J.; Torr, P. H.; and Koltun, V. 2021. Point transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhao, Z.; Wu, Z.; Wu, X.; Zhang, C.; and Wang, S. 2022. Crossmodal few-shot 3d point cloud semantic segmentation. In *Proceedings of the 30th ACM International Conference on Multimedia*, 4760–4768.

Zhou, H.; Cao, Y.; Chu, W.; Zhu, J.; Lu, T.; Tai, Y.; and Wang, C. 2022. SeedFormer: Patch Seeds based Point Cloud Completion with Upsample Transformer. In *European Conference on Computer Vision (ECCV)*.

Zhou, Y.; and Tuzel, O. 2018. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.