ASHF-Net: Adaptive Sampling and Hierarchical Folding Network for Robust Point Cloud Completion

Daoming Zong, Shiliang Sun*, Jing Zhao
School of Computer Science and Technology, East China Normal University, Shanghai, China
ecnuzdm@gmail.com, slsun@cs.ecnu.edu.cn, jzhao@cs.ecnu.edu.cn

Abstract

Estimating the complete 3D point cloud from an incomplete one lies at the core of many vision and robotics applications. Existing methods typically predict the complete point cloud based on the global shape representation extracted from the incomplete input. Although they could predict the overall shape of 3D objects, they are incapable of generating structure details of objects. Moreover, the partial input point sets obtained from range scans are often sparse, noisy and non-uniform, which largely hinder shape completion. In this paper, we propose an adaptive sampling and hierarchical folding network (ASHF-Net) for robust 3D point cloud completion. Our main contributions are two-fold. First, we propose a denoising auto-encoder with an adaptive sampling module, aiming at learning robust local region features that are insensitive to noise. Second, we propose a hierarchical folding decoder with the gated skip-attention and multi-resolution completion goal to effectively exploit the local structure details of partial inputs. We also design a KL regularization term to evenly distribute the generated points. Extensive experiments demonstrate that our method outperforms existing state-of-the-art methods on multiple 3D point cloud completion benchmarks.

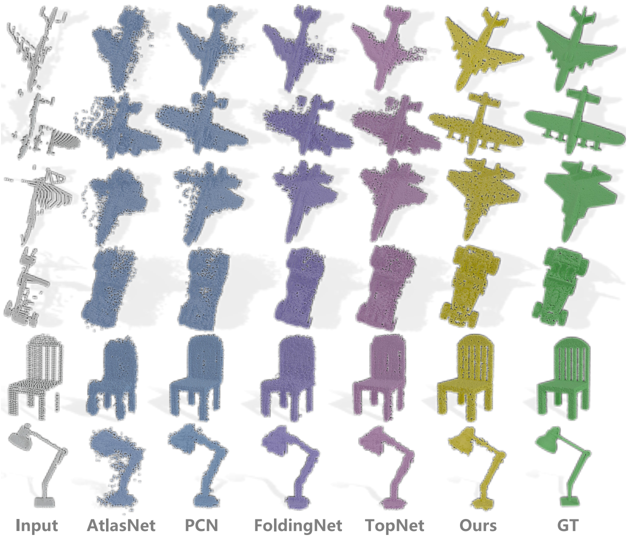

Figure 1: Our method has advantages in acquiring high-fidelity dense point clouds and avoiding uneven distribution, blurred details, or geometric structure loss.

Introduction

Point cloud, as a common data format for describing the 3D shape of an object, has achieved significant attention due to the rapid development of 3D acquisition technologies, and can be easily collected by 3D sensors and depth cameras. However, raw point clouds produced by those devices are usually highly sparse, noisy, and seriously incomplete due to limited sensor resolution and occlusion (Yuan et al. 2018; Wen et al. 2020), which hampers the downstream tasks, such as shape classification (Sarmad, Lee, and Kim 2019) and rendering. Consequently, recovering the complete point clouds from partial observations, namely point cloud completion, is important for subsequent 3D vision applications.

Unlike images or voxel grids where convolutional neural networks (CNNs) can be directly applied, point cloud processing poses great challenges to directly applying 2D and 3D convolutions due to its sparsity and disorder properties. Several methods (Li et al. 2016; Dai, Ruizhongtai Qi, and Nießner 2017; Han et al. 2017; Liu et al. 2019b) convert point clouds into 3D voxels and then apply 3D CNNs to them. However, the voxelization operation leads to an irreversible loss of geometric information and has a high computational cost. Therefore, most of the existing methods (Qi et al. 2017a,b) use the Multi-Layer Perceptrons (MLPs) to process point clouds directly. Generally, these methods first use sampling approaches to select central points from the incomplete input point clouds, and then utilize max-pooling to aggregate features across points in a global or hierarchical fashion. Finally, global point features are formed to model the complete shape. Despite the modest success of these methods, they have two main bottlenecks. First, raw point cloud data captured from these 3D sensors or reconstruction algorithms inevitably contain outliers or noise in real-world scenarios. However, a common issue in existing sampling approaches is that the sampled points are limited to a subset of the original point clouds, where little attention is paid to the biased effects of outliers and noise on point feature learning. Second, previous work on point completion

*Corresponding Author.

aims to predict the general/overall shape of a category but ignores the local structure details of a specific object (Huang et al. 2020), thus resulting in detailed geometrical structure loss (Yang et al. 2018). Concretely, they sometimes produce distorted results or even fail to preserve some of the actual structures which have been revealed in the partial inputs, as shown in Figure 1.

To overcome the noise problem, we propose a denoising auto-encoder to learn robust local region features directly from noisy-partial input point clouds. This is achieved by extending the recent adaptive sampling idea (Yan et al. 2020), which first re-weights the neighbors around the initial sampled points obtained by farthest point sampling (FPS) (Qi et al. 2017a), and then adaptively adjusts the coordinates of the sampled points beyond the entire point cloud. Such coordinate adjustment is conducive to fitting intrinsic geometric sub-manifolds and is able to eliminate the influence of noise or outliers. To preserve the structure details of the objects in partial inputs, we further propose a hierarchical folding decoder that combines a gated skip-attention module and a multi-resolution completion target. It completes the point cloud progressively, e.g., from skeleton to details, while preserving the structure details on local regions under all resolution levels. Specifically, the decoder has the same number of resolution levels as the number of layers in the encoder, where the gated skip-attention (GSA) module connects each level of the decoder to the corresponding layer of the encoder. We establish the GSA module to determine the relevance of attention results and queries by combining the attention mechanism with the attention gate, so that the decoder can selectively convey geometric information from the local regions of incomplete point clouds to generate complete point clouds. The multi-resolution supervision gradually increases the number of points that need to be predicted and guides the decoder progressively to predict the 3D shape of an object from skeleton to details, which enables the network to consistently generate the complete shape under all resolution levels. Further, we design a KL regularization term to enhance the uniformity of the output point cloud distribution. Our main contributions can be summarized as follows.

- We propose a denoising auto-encoder for learning robust local region features from partial inputs, which can effectively eliminate the influence of noise or outliers.
- We propose a hierarchical folding decoder combined with a gated skip-attention and a multi-resolution completion target. The gated skip-attention fuses the information of local region features from the encoder into the decoder at different resolutions, enabling the network to infer the missing regions with more detailed geometry information from incomplete point clouds. The multi-resolution completion target guides the decoder to detail the point clouds at different resolutions.
- We propose a novel KL regularization term to prevent excessive overlap or extreme sparsity of the generated point clouds.

Related Work

Existing works on point cloud completion and reconstruction can be roughly divided into four categories based on their network architecture types, which are `MLP-based`, `Folding-based`, `GAN-based` and `Voxel-based`.

MLP-based Networks. As a pioneer, PointNet (Qi et al. 2017a) first models each point independently using pointwise MLPs and then aggregate a global feature by the max-pooling operation. Extending PointNet, PointNet++ (Qi et al. 2017b) further applies a hierarchical structure with kNN grouping to learn local region features with increasing contextual scales. Both methods simply aggregate regional information by the largest activation, which means that the geometric relationships among 3D points are not yet fully utilized. To alleviate the loss of shape details caused by MLPs, FoldingNet (Yang et al. 2018) introduces a new decoding operation dubbed Folding and deforms a 2D plane into a 3D shape, which favors continuous and smooth structures. PCN (Yuan et al. 2018) first proposes the learning-based architecture focusing on shape completion tasks. TopNet (Tchapmi et al. 2019) proposes a hierarchical tree-structure rooted network to generate point cloud without assuming any specific topology for input point sets. NSFA (Zhang, Yan, and Xiao 2020) proposes the local feature and residual features aggregation strategies to represent the known part and the missing part separately. In addition, NSFA also designs a refinement component to prevent the generated point cloud from non-uniform distribution and outliers. Nevertheless, most of these work suffers from the information loss of structure details, as they predict the whole point cloud only from a single global shape representation.

Folding-based Networks. Folding-based decoder, as a generic architecture first introduced by Yang et al. (2018), is provably able to reconstruct an arbitrary point cloud from a 2D grid, achieving low reconstruction errors even for objects with delicate structures. Folding-based methods (Yuan et al. 2018; Tchapmi et al. 2019; Liu et al. 2019a) usually sample 2D grids from a 2D plane with fixed size and then concatenate them with the global shape representation extracted by the point cloud feature encoder. AtlasNet (Groueix et al. 2018), MSN (Liu et al. 2019a) and SA-Net (Wen et al. 2020) recover the complete point cloud of an object by estimating a collection of parametric surface elements and learn mappings from 2D square to 3D surface elements. Despite their limited success, the fine details of an object are often missed. As can be seen in Figure 1, most existing folding-based methods, such as PCN, FoldingNet and TopNet, are incapable of producing the structure details of an object to some extent. One reason is that they only rely on a single global shape representation to predict the entire point cloud, whereas the rich local region information favorable for recovering detailed geometric structures are not fully utilized.

GAN-based Networks. L-GAN (Achlioptas et al. 2018) first introduces the deep generative model for the point cloud. Although L-GAN is capable of performing point cloud completion tasks, its architecture is not particularly tailored for the shape completion tasks, and hence the performance is not competitive. Among GAN-based works, one purpose to use Generative Adversarial Networks (GANs) is to avoid complex optimizations and speed up predictions. For example, RL-GAN-Net (Sarmad, Lee, and Kim 2019) introduces a reinforcement learning agent to control GAN network for real-time point cloud shape completion. Another line of GAN-based work is to use adversarial loss to enhance the prediction accuracy, such as PF-Net (Huang et al. 2020) and CRN (Wang, Ang Jr, and Lee 2020).

Voxel-based Networks. Early works (Dai, Ruizhongtai Qi, and Nießner 2017; Han et al. 2017) typically apply 3D convolutional neural networks (CNNs) build upon the volumetric representation of 3D point clouds. For instance, several works (Mao, Wang, and Li 2019; Hua, Tran, and Yeung 2018) develop CNNs operating on discrete 3D grids that are transformed from point clouds. PointCNN (Qi et al. 2017a) achieves permutation invariance through a χ-conv transformation. Hua, Tran, and Yeung (2018) define convolutional kernels on regular 3D grids, where the points are assigned with the same weights when falling into the same grid. Thomas et al. (2019) propose both rigid and deformable kernel point convolution operators for 3D point clouds using a set of learnable kernel points. However, converting point clouds into 3D volumes introduces a quantization effect that inevitably discards some details of an object (Wang and Lu 2019) and is difficult to capture high-resolution or fine-grained features. Recently, GRNet (Xie et al. 2020) introduces 3D grids as intermediate representations to regularize unordered point clouds. Despite its limited success in relieving the detail loss, it suffers from high computational cost that is cubic to the resolution of the 3D grids.

Approach

Problem Setup and Notations. Given a partial point set $\mathcal{S} \in \mathbb{R}^{N_p \times 3}$, our goal is to generate a complete point set $\mathcal{R} \in \mathbb{R}^{N_c \times 3}$, where N_p, N_c denote the number of the partial input points and complete output points, respectively.

Point Cloud Denoising Auto-encoder

Farthest point sampling (FPS) (Qi et al. 2017a) is the most widely used sampling method on shape completion tasks (Huang et al. 2020; Wen et al. 2020), as it has a relatively good coverage of the entire point set. However, one main issue in FPS is that its large sensitivity to the outliers and noise, which makes it highly unstable for dealing with real-world point cloud data. To endow our model with noise immunity, we propose to rectify outliers and reduce noise using the spatial and feature information of the input points.

We first apply FPS to obtain the relatively uniform points as original sampled points similar to (Yan et al. 2020). Let $\mathcal{S}_t \in \mathbb{R}^{N_t \times 3}$ be the sampled N_t points from N_p input points at the t-th layer, $p_k^t \in \mathcal{S}_t$ be a sampled point with its related point feature f_k^t from $\mathcal{F}^t \in \mathbb{R}^{N_t \times D_t}$ at the t-th layer. For the central point p_i^t, we first gather its K neighboring points, i.e., $\mathcal{N}_i^t = \{p_1^t, \dots, p_k^t, \dots, p_K^t\}$, via the KNN algorithm based on point-wise Euclidean distances. We then explicitly encode the relative point positions as follows:

$$r_k^t = \mathrm{MLP}\left(p_i^t \oplus p_k^t \oplus (p_i^t - p_k^t) \oplus ||p_i^t - p_k^t||\right), \qquad (1)$$

where p_i^t or p_k^t denotes the `x-y-z` positions of points, $\oplus$ denotes the concatenation operation, and $||\cdot||$ measures the Euclidean distance between the central point and its neighbors. After that, we concatenate the encoded relative point positions r_k^t with its related point features f_k^t to obtain augmented feature vectors $\hat{f}_k^t$, i.e., $\hat{f}_k^t = f_k^t \oplus r_k^t$ for $k = 1, \dots, K$. Now, we have a neighboring feature set $\hat{\mathcal{F}}_i^t = \{\hat{f}_1^t, \dots, \hat{f}_k^t, \dots, \hat{f}_K^t\}$ for the central point p_i^t with the local geometric information encoded in it. To adaptively capture the features of neighboring points at different proximity, we aggregate the neighboring features using the attention mechanism (Vaswani et al. 2017). The aggregation rule for central points is defined as

$$\tilde{f}_i^{t+1} = \phi_t \left(f_i^t \oplus \sum\nolimits_{k \in \mathcal{N}_i^t} \mathcal{A}_{i,k} \psi_t(\hat{f}_k^t) \right), \qquad (2)$$

where ψ_t is a layer-specific linear transformation and can be easily implemented by independent 1D convolution `Conv` : $\mathbb{R}^{D_t} \mapsto \mathbb{R}^{D_{t+1}}$. ϕ_t is a projection matrix for feature transformation. $\mathcal{A}_i = [\alpha_{i,k}]_{k=1}^K$ is the attention weight matrix which adaptively controls the contribution of a neighbor p_k to its central point p_i. The attention score $\alpha_{i,k}$ can be computed by various score functions. For simplicity, we use the dot-product attention (Vaswani et al. 2017) as follows:

$$\alpha_{i,k} = \mathrm{Softmax}(< W_q^t \hat{f}_i^t, W_k^t \hat{f}_k^t >). \qquad (3)$$

To eliminate the biased effect of noise and outliers, we also update the current coordinates of the central point p_i^t with its weighted average neighbor point coordinates, that is,

$$\tilde{p}_i^{t+1} = \mathcal{A}_i P_i,\ P_i = [\ p_k^t\ ]_{k=1}^K, \qquad (4)$$

where $P_i \in \mathbb{R}^{K \times 3}$ is the K neighbors' coordinate matrix corresponding to the sampled central point p_i^t. Note that the new coordinates of p_i^t are not confined to a subset of original point clouds, thereby enabling them more suitable for feature learning with intrinsic geometry and more robust to noise. We refer to such feature and position update rule as the adaptive sampling (AS) module.

Given a noisy and partial point set S, we train an encoder $E_\gamma^r : \mathcal{S} \mapsto \mathbb{X}_r$ with AS modules and a decoder $D_\psi^r : \mathbb{X}_r \mapsto \tilde{\mathcal{S}}$ with the reconstruction loss defined as follows:

$$\mathcal{L}^{\mathrm{EMD}}(\gamma, \psi) = \mathbb{E}_{\mathcal{S} \sim p_{\mathrm{partial}}} d(\mathcal{S}, D_\psi^r(E_\gamma^r(\mathcal{S}))), \qquad (5)$$

where $\mathcal{S} \sim p_{\mathrm{partial}}$ denotes point set drawn from the set of noisy and partial point sets, $d(\mathcal{S}_1, \mathcal{S}_2)$ is the Earth Mover's Distance (EMD) (Fan, Su, and Guibas 2017) between point sets $\mathcal{S}_1$ and $\mathcal{S}_2$, forcing the reconstructed output to have the same density distribution as the input, and (γ, ψ) are the learnable parameters of the encoder and decoder networks, respectively. For the architecture of encoder E_γ^r, we stack the AS module hierarchically which resembles the PointNet++ (Qi et al. 2017b), while for the decoder D_ψ^r, we transform the latent vector $\mathbb{X}_r$ by using three independent point-wise MLPs with ReLU activations to generate a reconstructed point set $\tilde{\mathcal{S}}$.

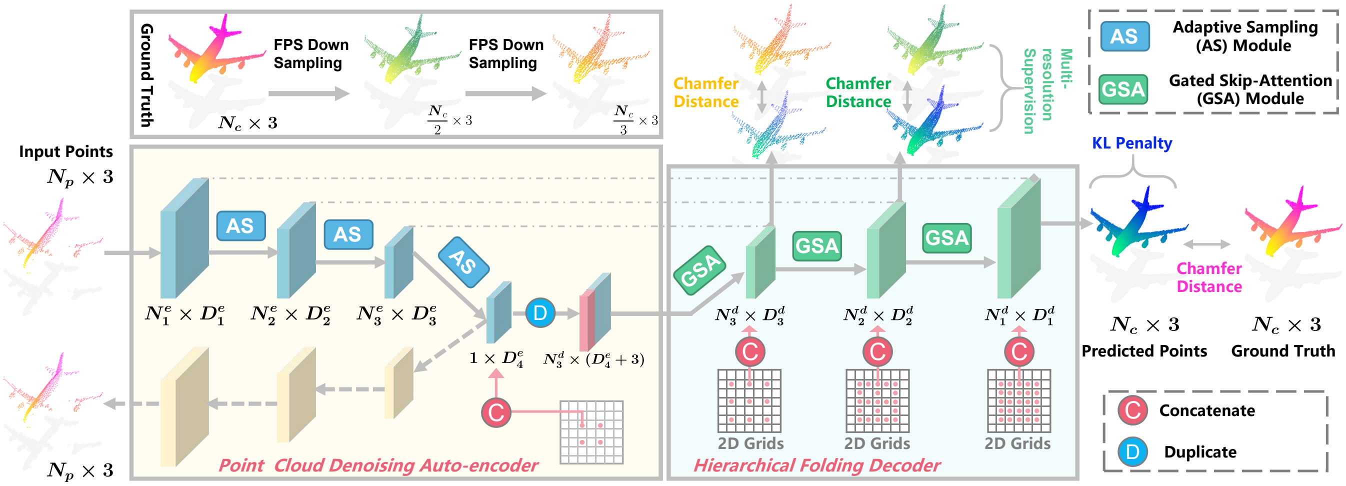

Figure 2: Overall architecture of ASHF-Net. ASHF-Net mainly consists of three modules: the denoising auto-encoder aims to extract robust local region features from the partial inputs; the hierarchical folding decoder aims to progressively reconstruct the complete point clouds; the gated skip-attention bridges the local region features in encoder and the point features in decoder.

Hierarchical Folding Decoder

As shown in Figure 2, our decoder hierarchically completes the point clouds at three resolution levels, corresponding to the three resolution levels of the encoder. Each resolution level of the decoder comprises a gated skip-attention module for conveying the local region features from the encoder, and an associated supervision target for guiding the point cloud completion from skeleton to details.

To begin with, denote by $\mathbb{X}_r \in \mathbb{R}^{1\times \mathtt{D}^{\mathtt{e}}_4}$ the global shape vector extracted from the denoising auto-encoder. We first duplicate $\mathbb{X}_r$ by $\mathtt{N}^{\mathtt{d}}_3$ times to form $\mathtt{N}^{\mathtt{d}}_3 \times \mathtt{D}^{\mathtt{e}}_4$ dimensional points features for the 3-th level of decoder. To mimic the morphing of a 2D square into a 3D surface, we propose to sample 2D grids with an increasing density from the 2D plane of fixed size. Specifically, for the $\mathtt{N}^{\mathtt{d}}_3$ point features in the 3-th layer of decoder, the $\mathtt{N}^{\mathtt{d}}_3$ 2D grids are evenly sampled from 46×46 2D plane and concatenated with the point features, as illustrated in Figure 2. Then, the point features with 2D grids are passed through MLPs and transformed into 3-dimensional latent codewords following FoldingNet (Yang et al. 2018). These 3-dimensional codewords are again concatenated with the point features in the 3-th level of decoder. After that, we obtain the point feature $P_i \in \mathbb{R}^{\mathtt{N}^{\mathtt{d}}_3\times(\mathtt{D}^{\mathtt{e}}_4+3)}$.

To effectively integrate the point features P_i generated by the decoder and the local region features E_i extracted from the encoder, we use the attention mechanism (Vaswani et al. 2017) to jointly exploit attentions over them. We first compute the query $Q_i \in \mathbb{R}^{\mathtt{N}^{\mathtt{d}}_3\times\mathtt{d}}$, key $K_i \in \mathbb{R}^{\mathtt{N}^{\mathtt{e}}_3\times\mathtt{d}}$ and value $V_i \in \mathbb{R}^{\mathtt{N}^{\mathtt{e}}_3\times\mathtt{d}}$ by three individual linear projections, i.e., $Q_i = W_qP_i, K_i = W_eE_i, V_i = W_vE_i$. A skip-attention module $f_{att}(Q_i, K_i, V_i)$ measures the similarity between Q_i and K_i and use the similarity scores to compute weighted average vectors over V_i, which is defined as

$$\hat{V}_i = f_{att}(Q_i, K_i, V_i) = \mathrm{Softmax}(Q_iK_i^T)V_i. \qquad (6)$$

However, not all region features under each level of resolutions will contribute equally to 3D shape inference and reconstruction. To avoid introducing redundant or misleading information and focus on the most relevant regions, we propose the gated skip-attention module to measure the attention results and the query Q_i, as depicted in Figure 3. The GSA module generates an information vector I_i and an attention gate G_i via two separated linear transformations both conditional on the attention results $\hat{V}_i$ and the query Q_i,

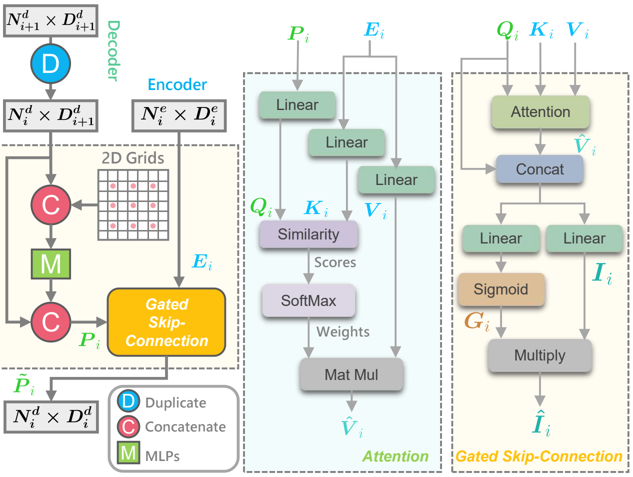

Figure 3: An illustration of the gated skip-attention module.

$$I_i = W^s_qQ_i^T + W^s_v\hat{V}_i^T + B^s, \qquad (7)$$

$$G_i = \sigma(W^g_qQ_i^T + W^g_v\hat{V}_i^T + B^g), \qquad (8)$$

where $W^s_q, W^s_v, W^g_q, W^g_v \in \mathbb{R}^{\mathtt{N}^{\mathtt{d}}_3\times\mathtt{d}}$ are trainable transformation matrices and $B^s, B^g \in \mathbb{R}^{\mathtt{N}_3\times\mathtt{d}}$ are bias matrices. σ denotes the sigmoid activation. We then add a filter by applying the attention gate to the information vector using element-

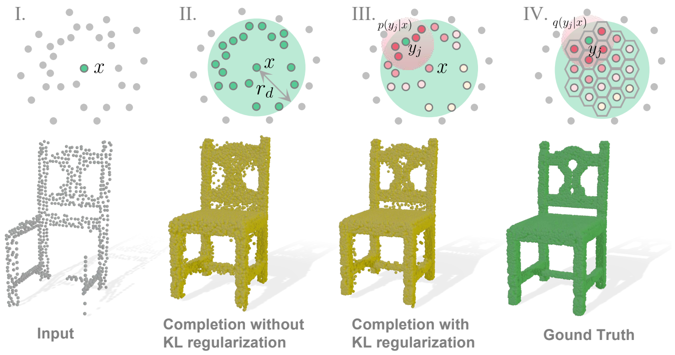

Figure 4: An illustration of the KL penalty term and the shape completion effect of adding KL penalty.

wise multiplication and obtain the attended information:

$$\hat{I}_i = G_i \odot I_i, \;\; \hat{P}_i = P_i + \hat{I}_i, \tag{9}$$

where $\odot$ denotes element-wise multiplication, and the final point feature representation $\tilde{P}_i \in \mathbb{R}^{\mathtt{N}_3^\mathtt{d} \times \mathtt{D}_3^\mathtt{d}}$ is obtained by a residual connection (He et al. 2016).

The point features $\tilde{P}_i$ in each level of the decoder accounts for predicting point clouds at different resolutions. We first use FPS to sample two different scale point sets from the original complete point clouds as the supervisors. As shown in Figure 2, two point clouds $\mathcal{R}^3 \in \mathbb{R}^{\frac{\mathtt{N_c}}{3} \times 3}$ and $\mathcal{R}^2 \in \mathbb{R}^{\frac{\mathtt{N_c}}{2} \times 3}$ are sampled from the ground truth point cloud $\mathcal{R} \in \mathbb{R}^{\mathtt{N_c} \times 3}$. We directly feed the point feature $\tilde{P}_i$ at the i-th level of decoder into MLPs to generate $\mathtt{N_c/i}$ points, which can be represented as $\mathcal{R}_p^i = \mathcal{RS}(\mathrm{MLPs}(\tilde{P}_i))$, where $\mathcal{RS}(\cdot)$ is a reshape operation. We choose the Chamfer Distance (CD) (Fan, Su, and Guibas 2017) as the completion loss for its efficiency. Note that our decoder predicts three point clouds at different resolutions, the total completion loss $\mathcal{L}_{com}$ hence consists of three terms and is computed by

$$\mathcal{L}_{com} = \mathcal{L}_{\mathrm{CD}}(\mathcal{R}_p, \mathcal{R}) + \alpha \mathcal{L}_{\mathrm{CD}}(\mathcal{R}_p^2, \mathcal{R}^2) + \beta \mathcal{L}_{\mathrm{CD}}(\mathcal{R}_p^3, \mathcal{R}^3) \tag{10}$$

where α and β are hyperparameters fixed to 0.5 and 0.3 respectively for our experiments.

To make the final generated point cloud $\mathcal{R}_p \in \mathbb{R}^{N_c \times 3}$ more evenly distributed, we propose a regularization term to penalize points if points are located too close to its neighboring points or too sparse distributed. Specifically, we first use FPS to pick M seed points in $\mathcal{R}_p$ and use a ball query of radius r_d to crop a point subset (as shown in Figure 4) from $\mathcal{R}_p$ at each seed. Then we compute the probability density function (pdf) of the neighboring points conditional on the seed point in a neighborhood, which is formulated as:

$$p(y_j|x) \approx \frac{1}{|\mathcal{N}(x)|\sigma^3} \sum_{k \in \mathcal{N}(x)} \{ \prod_{d=1}^{3} h(\frac{y_{j,d} - y_{k,d}}{\sigma}) \}, \tag{11}$$

where σ is the bandwidth which determines the smoothing of the resulting sample density function (we use $\sigma = .25 r_d$). h is the density estimation kernel, a non-negative function whose integral equals 1 (we use a Gaussian), and d is one the three dimensions of $\mathbb{R}^3$. The pdf of a point y_j with respect to a given point x is always relative to all other samples in the receptive field. Notice that $p(y_j|x)$ is high where the sampled points are dense and low where they are sparse. Ideally, the sampled points should be uniformly distributed inside a small neighborhood, i.e., $q(y_j|x) = 1/|\mathcal{N}(x)|$. We hence minimize the Kullback-Leibler (KL) divergence from $p(y_j|x)$ to $q(y_j|x)$ as a regularization term, that is,

$$\mathbb{KL}(p||q) = \sum_{i=1}^{M} \sum_{j=1}^{\mathcal{N}_x} p(y_j|x) \log \frac{p(y_j|x)}{q(y_j|x)}. \tag{12}$$

Experiments

Datasets and Implementation Details

ShapeNet. The ShapeNet dataset (Wu et al. 2015) for point cloud completion is derived from PCN (Yuan et al. 2018), which consists of 30,974 3D models from 8 categories. The ground truth point clouds containing 16,384 points are uniformly sampled on the mesh surfaces and the partial point input with 2048 points is generated by back-projecting 2.5D depth images into 3D. For a fair comparison, we use the same train/val/test splits as PCN (Yuan et al. 2018).

KITTI. The KITTI dataset (Geiger et al. 2013) is collected from a sequence of real-world Velodyne LiDAR scans, also derived from PCN (Yuan et al. 2018). For each frame, the car objects are extracted according to the 3D bounding boxes, which results in 2,401 partial point clouds. Note that the partial point clouds in KITTI are highly sparse and have no complete point clouds as ground truth. Besides, the point number of the incomplete car has a large range of variation. To obtain a fixed number of input points, we randomly select 2,048 points by randomly dropping or replicating points.

Our model is implemented with PyTorch on 8 NVIDIA RTX 2080Ti GPUs. We first train the auto-encoder by optimizing Eq. 5. Then we train the decoder with a small learning rate (1e-5) using the Adam optimizer and save the best model if the validation loss does not decrease for 10 epochs. The batch size is set to 36. For the KL regularization term, we set the number of seed points M to 100 and select the ball query radius r_d from set $\{10, 8, 6, 4, 2\}$. We compute Eq. 12 for each r_d and then sum up the results as the regularization.

Comparison with State-of-the-Arts

Results on ShapeNet. All competing methods except for AtlasNet can easily predict the coordinates of 16384 points with 2048 points as input. In particular, we sample 16,384 points from the primitive surface elements generated by TopNet. We adjust the number of nodes and the size of feature embedding to make TopNet generate 16384 points. To generate 16384 points, we increase the number of mapping MLPs to 32 in MSN. Quantitative results in Table 1 indicate that ASHF-Net outperforms all competitive methods in terms of the Chamfer Distance. Figure 5 shows the qualitative results for point cloud completion on the test set of ShapeNet. We can observe that, AtlasNet, PCN, FoldingNet, and TopNet miss some structure details and only recover the sketch of object shapes, which may be caused by using only

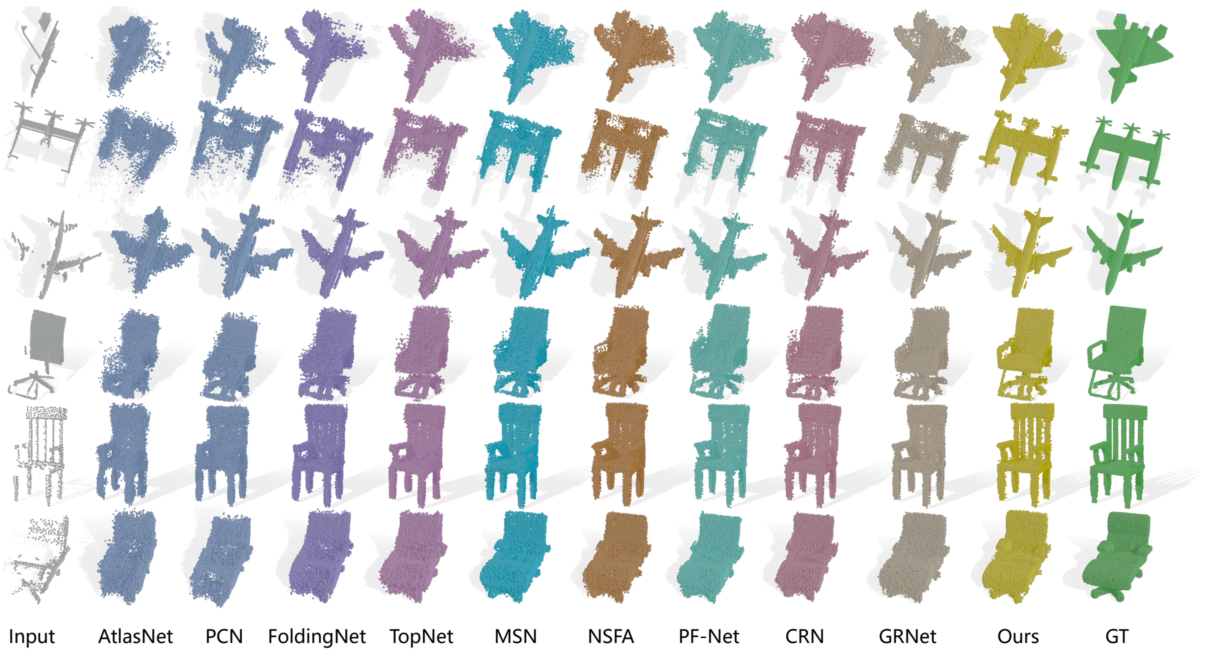

Figure 5: Visualization of point cloud completion using different methods. Each output point cloud contains 16384 points.

one single global shape vector for point cloud completion. Besides being competitive in preserving the geometric details of some recent models such as MSN, GRNet, NSFA, PF-Net and CRN, our model also performs best in detail completion. As shown in Figure 5, although they recognize the partial input as a chair, they cannot recover its details. In contrast, our model can predict more realistic structures and generate continuous and smooth details with more fidelity. Moreover, our model employs the KL divergence to ensure the points distribute uniformly, while most of these methods are more likely to overfill points in some regions. It can also be observed that our method can preserve the known structures, while most of the competitive methods distort or even neglect the structures revealed in the input.

Results on KITTI. Since there are no complete ground truth point clouds for KITTI, we use Fidelity, Minimal Matching Distance (MMD), Consistency and Uniformity (Xie et al. 2020) as evaluation metrics. Following the settings in (Xie et al. 2020; Yuan et al. 2018), we fine-tune all comparison methods on cars of the ShapeNet, except PCN and GRNet, which are evaluated directly using their released output or pre-trained model weights. Models trained specifically on cars are capable of learning the prior knowledge of the object class. Table 2 lists the completion results for cars in the LiDAR scans from the KITTI dataset. Thanks to the hierarchical folding decoder, the prediction of our model presents strong spatial continuity with a high level of restoration and the input is well preserved. The optimal Uniformity indicates that point clouds generated by the ASHF-Net are more evenly distributed than other methods, benefited from the KL-divergence penalty. In addition, the best Consistency confirms that the proposed method generates more reasonable shape completion with fewer genus-wise distortions.

Methods	Plane	Cabinet	Car	Chair	Lamp	Sofa	Table	Vessel
AtlasNet	1.753	5.101	3.237	5.226	6.342	5.990	4.359	4.177
PCN	1.400	4.450	2.445	4.838	6.238	5.129	3.569	4.062
FoldingNet	3.151	7.943	4.976	9.225	9.234	8.895	6.691	7.325
TopNet	2.152	5.623	3.513	6.346	7.502	6.949	4.784	4.359
MSN	1.543	7.249	4.711	4.539	6.479	5.894	3.797	3.853
NSFA	1.751	5.310	3.429	5.012	4.729	6.413	4.000	3.555
PF-Net	1.551	4.430	3.116	3.962	4.213	5.874	3.347	3.887
CRN	1.455	4.212	2.969	3.238	5.160	5.013	3.988	3.962
GRNet	1.531	3.620	2.752	2.945	2.649	3.613	2.552	2.122
ASHF-Net	**1.398**	**3.490**	**2.322**	**2.815**	**2.519**	**3.483**	**2.422**	**1.992**

Table 1: Quantitative comparisons on the ShapeNet dataset for shape completion (on 16,384 points) w.r.t. the Chamfer Distance ($\times 10^{-4}$). Lower is better.

Ablation Study

We empirically examine the effectiveness of principal components in ASHF-Net via an ablation study, including the adaptive sampling (AS) module, the gated skip-attention module, the multi-resolution supervision and the KL regularization. The results on ShapeNet in terms of Chamfer Distance (CD) are shown in Table 3. Overall, we can see that each of the four new components consistently boosts the performance of ASHF-Net. We first construct an encoder resembling PointNet++ (Qi et al. 2017b) and a folding-based decoder similar to FoldingNet (Yang et al. 2018) as the baseline (`BS`). The insertion of the AS module over the baseline leads to lower CD (`w/AS`), indicating that local region features extracted by the AS module are more robust than the feature learning combined with FPS and max-poling in

Methods	FD	MMD	Consistency	Uniformity		
	($\times10^{-3}$)	($\times10^{-3}$)	($\times10^{-3}$)	0.4%	0.8%	1.2%
AtlasNet	1.759	2.108	0.700	1.146	0.874	0.686
PCN	2.235	1.366	1.557	3.662	7.710	10.823
FoldingNet	7.467	**0.537**	1.053	1.245	1.262	1.063
TopNet	5.354	0.636	0.568	1.353	1.219	0.950
MSN	**0.434**	2.259	1.951	0.822	0.523	0.383
NSFA	1.281	0.891	0.491	0.992	0.767	0.552
GRNet	0.816	0.568	0.313	0.632	0.489	0.352
PF-Net	1.137	0.792	0.436	0.881	0.682	0.491
CRN	1.023	0.872	0.431	0.870	0.673	0.485
ASHF-Net	0.773	0.541	**0.298**	**0.602**	**0.466**	**0.335**

Table 2: Results of point cloud completion on the KITTI dataset w.r.t. Fidelity Distance (FD), Minimal Matching Distance (MMD), Consistency, and Uniformity computed on 16,384 points. Lower is better.

Evaluation	Category	Ablation Versions				
		BS	w/ AS	w/ GSA	w/ $\mathcal{R}_p^{2,3}$	w/ u
CD ($\times10^{-4}$)	Plane	2.169	1.915	1.672	1.499	1.398
	Cabinet	5.626	5.543	3.608	3.577	3.490
	Car	3.516	3.302	2.701	2.672	2.322
	Chair	6.361	5.313	2.993	2.842	2.815
	Lamp	7.517	7.395	2.699	2.553	2.519
	Sofa	5.126	4.889	3.575	3.505	3.483
	Table	6.956	5.593	2.517	2.449	2.422
	Vessel	4.791	3.738	2.159	2.026	1.922
	Average	5.276	4.711	2.741	2.640	2.555

Table 3: Quantitative comparisons of the ablation study.

PointNet++. Then, the addition of the GSA module further improves the performance (`w/GSA`), validating the effectiveness of the GSA module in communicating the local region features extracted by the encoder with the point features generated by the decoder. Also, the introduction of the multi-resolution supervision results in performance gains (`w/R`). This suggests that integrating multi-resolution supervision is favorable for point cloud completion. Finally, the insertion of the KL regularization, hence the full version of the model, consistently achieves the lowest scores (`w/u`) in terms of CD, implying that ASHF-Net is able to generate a more uniformly distributed point cloud with high fidelity.

Robustness Test

Most existing point cloud completion methods can achieve good performance on clean synthetic datasets because they have stable distribution and do not contain any noise, such as PointNet++. However, in addition to high sparsity, raw point clouds data acquired by 3D sensor or reconstruction algorithms inevitably contain outliers or noise (Li et al. 2019; Yan et al. 2020). To further verify the generalization and robustness of the proposed model, we replace a certain number of randomly picked points with random noise on the test set of ShapeNet, as shown in Figure 6 (the red points represent random noise points). Notably, we also test the influence of noise on the completion effect of ASHF-Net's two baselines, namely PointNet* and PointNet++*, whose encoders are PointNet and PointNet++ respectively but the decoders are the same as our model. Figure 6 shows the completion results on the `plane` class in terms of CD varying by the number of replaced noisy points. We can see that as the noise increases, ASHF-Net consistently outperforms its competitors without significant performance degradation. Besides, we use t-SNE tool (Maaten and Hinton 2008) to visualize the latent representations extracted from our encoder and the PointNet++ encoder. It can be seen that after adding noise, ASHF-Net can learn more discriminative representations than PointNet++, indicating a stronger noise immunity.

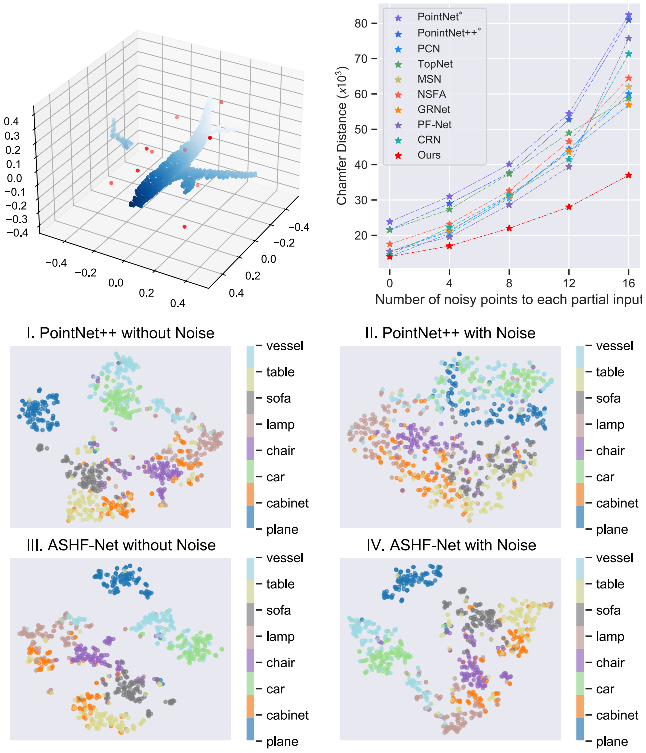

Figure 6: Illustration of point clouds with some points replaced by random noise and performance comparisons of different models against noise. (I & II). Visualization of the latent representations extracted by the PointNet++ encoder with clean inputs and noisy inputs. (III & IV). Visualization of the latent representations extracted by the ASHF-Net encoder with clean inputs and noisy inputs.

Conclusion

We have presented a novel point cloud completion framework, ASHF-Net, that: (a) is robust to noise or outliers in raw input point clouds; (b) progressively generates the point clouds at different resolution levels; (c) guarantees the even distribution of the generated point clouds. Experiments on multiple point cloud completion tasks as well as noise tests have verified the robustness and effectiveness of the model.

Acknowledgments

This work is supported by the NSFC Projects 62076096 and 62006078, Shanghai Municipal Project 20511100900, and Chenguang Program (No. 19CG25) by Shanghai Education Development Foundation and Municipal Education Commission. Prof. Shiliang Sun is the corresponding author of this paper.

References

Achlioptas, P.; Diamanti, O.; Mitliagkas, I.; and Guibas, L. 2018. Learning representations and generative models for 3d point clouds. In *ICML*, 40–49.

Dai, A.; Ruizhongtai Qi, C.; and Nießner, M. 2017. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *CVPR*, 5868–5877.

Fan, H.; Su, H.; and Guibas, L. J. 2017. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, 605–613.

Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets robotics: The kitti dataset. Intern. *Journal of Robotics Research (IJRR)* 1: 6.

Groueix, T.; Fisher, M.; Kim, V. G.; Russell, B. C.; and Aubry, M. 2018. A papier-mâché approach to learning 3d surface generation. In *CVPR*, 216–224.

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 *ICCV*, 85–93.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Hua, B.-S.; Tran, M.-K.; and Yeung, S.-K. 2018. Pointwise convolutional neural networks. In *CVPR*, 984–993.

Huang, Z.; Yu, Y.; Xu, J.; Ni, F.; and Le, X. 2020. PF-Net: Point Fractal Network for 3D Point Cloud Completion. In *CVPR*, 7662–7670.

Li, D.; Shao, T.; Wu, H.; and Zhou, K. 2016. Shape completion from a single rgbd image. *IEEE transactions on visualization and computer graphics* 23(7): 1809–1822.

Li, R.; Li, X.; Fu, C.-W.; Cohen-Or, D.; and Heng, P.-A. 2019. Pu-gan: a point cloud upsampling adversarial network. In *ICCV*, 7203–7212.

Liu, M.; Sheng, L.; Yang, S.; Shao, J.; and Hu, S.-M. 2019a. Morphing and sampling network for dense point cloud completion. *arXiv preprint arXiv:1912.00280* .

Liu, Z.; Tang, H.; Lin, Y.; and Han, S. 2019b. Point-Voxel CNN for efficient 3D deep learning. In *NIPS*, 965–975.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9(Nov): 2579–2605.

Mao, J.; Wang, X.; and Li, H. 2019. Interpolated convolutional networks for 3d point cloud understanding. In *ICCV*, 1578–1587.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 652–660.

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 *NIPS*, 5099–5108.

Sarmad, M.; Lee, H. J.; and Kim, Y. M. 2019. Rl-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion. In *CVPR*, 5898–5907.

Tchapmi, L. P.; Kosaraju, V.; Rezatofighi, H.; Reid, I.; and Savarese, S. 2019. Topnet: Structural point cloud decoder. In *CVPR*, 383–392.

Thomas, H.; Qi, C. R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; and Guibas, L. J. 2019. Kpconv: Flexible and deformable convolution for point clouds. In *ICCV*, 6411–6420.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*, 5998–6008.

Wang, X.; Ang Jr, M. H.; and Lee, G. H. 2020. Cascaded Refinement Network for Point Cloud Completion. In *CVPR*, 790–799.

Wang, Z.; and Lu, F. 2019. VoxSegNet: Volumetric CNNs for semantic part segmentation of 3D shapes. *IEEE transactions on visualization and computer graphics* .

Wen, X.; Li, T.; Han, Z.; and Liu, Y.-S. 2020. Point cloud completion by skip-attention network with hierarchical folding. In *CVPR*, 1939–1948.

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 *CVPR*, 1912–1920.

Xie, H.; Yao, H.; Zhou, S.; Mao, J.; Zhang, S.; and Sun, W. 2020. GRNet: Gridding Residual Network for Dense Point Cloud Completion. *arXiv preprint arXiv:2006.03761* .

Yan, X.; Zheng, C.; Li, Z.; Wang, S.; and Cui, S. 2020. PointASNL: Robust Point Clouds Processing using Nonlocal Neural Networks with Adaptive Sampling. In *CVPR*, 5589–5598.

Yang, Y.; Feng, C.; Shen, Y.; and Tian, D. 2018. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *CVPR*, 206–215.

Yuan, W.; Khot, T.; Held, D.; Mertz, C.; and Hebert, M. 2018. Pcn: Point completion network. In *2018 International Conference on 3D Vision (3DV)*, 728–737. IEEE.

Zhang, W.; Yan, Q.; and Xiao, C. 2020. Detail Preserved Point Cloud Completion via Separated Feature Aggregation. *arXiv preprint arXiv:2007.02374* .