Deep Cascade Generation on Point Sets

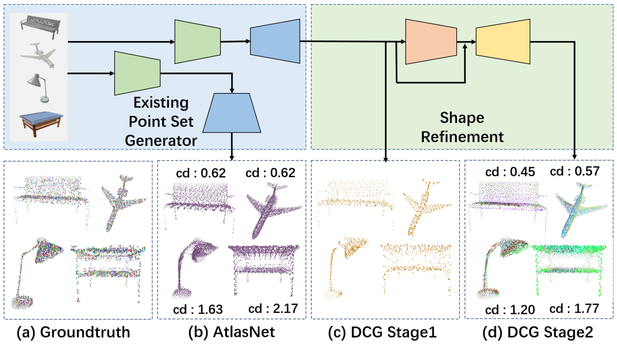
Kaiqi Wang, **Ke Chen**[*] and **Kui Jia**
South China University of Technology
mswkq@mail.scut.edu.cn, {chenk, kuijia}@scut.edu.cn

Abstract

This paper proposes a deep cascade network to generate 3D geometry of an object on a point cloud, consisting of a set of permutation-insensitive points. Such a surface representation is easy to learn from, but inhibits exploiting rich low-dimensional topological manifolds of the object shape due to lack of geometric connectivity. For benefiting from its simple structure yet utilizing rich neighborhood information across points, this paper proposes a two-stage cascade model on point sets. Specifically, our method adopts the state-of-the-art point set autoencoder to generate a sparsely coarse shape first, and then locally refines it by encoding neighborhood connectivity on a graph representation. An ensemble of sparse refined surface is designed to alleviate the suffering from local minima caused by modeling complex geometric manifolds. Moreover, our model develops a dynamically-weighted loss function for jointly penalizing the generation output of cascade levels at different training stages in a coarse-to-fine manner. Comparative evaluation on the publicly benchmarking ShapeNet dataset demonstrates superior performance of the proposed model to the state-of-the-art methods on both single-view shape reconstruction and shape autoencoding applications.

1 Introduction

3D geometry of an object is a vital property in a number of applications such as computer vision [Simon *et al.*, 2018; Bronstein *et al.*, 2017] and graphics [Kazhdan *et al.*, 2006], making representation learning to generate a high-resolution surface active and hot. In the era of deep learning since 2012, Euclidean convolution operation has gained significant progress on feature encoding for regularly sampled data such as images [He *et al.*, 2016; Huang *et al.*, 2017] or videos [Karpathy *et al.*, 2014]. In 3D computer vision, volumetric voxels are the first attempt for surface generation [Choy *et al.*, 2016], owing to direct application of 3D Euclidean convolutional operation on discretized regular grids. Although existing deep learning methods [Choy *et al.*, 2016; Girdhar *et al.*, 2016; Tatarchenko *et al.*, 2017; Tulsiani *et al.*, 2017; Yan *et al.*, 2016; Tatarchenko *et al.*, 2016] based on a voxel representation achieve competitive generation performance, they still suffer from inherent drawbacks of surface representation, i.e., voxel-wise information sparsity, which thus leads to expensive memory cost $O(h^3)$ proportional to cubic of voxels' dimension h.



(a) Groundtruth (b) AtlasNet (c) DCG Stage1 (d) DCG Stage2

Figure 1: Point surface generation from single images with the proposed deep cascade generation (DCG) network and the state-of-the-art AtlasNet. We adopt the AtlasNet as the autoencoder of the first cascade level in our DCG net. For a fair comparison, (b) the AtlasNet and (d) the DCG have the identical size of points in their final point cloud representation. The values of the Chamfer distance (cd) in (b) and (d) indicate the errors between point predictions and ground truth point clouds illustrated in (a). Point cloud samples are from the popular ShapeNet benchmark.

A point cloud providing on-surface details is a powerful parametric shape representation, which can alleviate 3D data occupancy sparsity in the rasterized representation. Nevertheless, Euclidean convolution based deep networks cannot be applied to point set generation in view of irregular structure of points, which encourages a number of deep algorithms to regress points' 3D position directly, e.g., point set generation (PSG) [Fan *et al.*, 2017] and AtlasNet [Groueix *et al.*, 2018]. These methods are designed in an encoder-decoder structure, reconstructing a collection of points from a latent feature vector encoded from input data, which achieve state-of-the-art generation performance and computational efficiency. In surface generation, a point cloud representa-

[*] Corresponding author

tion favors for its simplicity to learn from, with the price of missing points' neighborhood information, which preserves low-dimensional manifolds of shape. Point-wise correlation has been verified as an important property of shape in 3D recognition, e.g., DGCNN [Wang *et al.*, 2018b] and SO-Net [Li *et al.*, 2018]. Recently, an alternative parametric representation – a triangle mesh [Groueix *et al.*, 2018; Wang *et al.*, 2018a] can incorporate the underlying manifold structure of a surface, but suffers from irregular and complex combinatorial relation and thus is made challenging in the perspective of model learning.

A simple and flexible representative structure and *rich local neighborhood information* are both desired properties for shape reconstruction and autoencoding. For both advantages, we design a deep cascade model of two encoder-decoders, which aim to firstly generating a coarse surface and then locally refining 3D shape via feature encoding on its graph representation respectively. Specifically, the former one replicates the network structure as the state-of-the-art competitors (e.g., the AtlasNet [Groueix *et al.*, 2018] in our experiments), while the latter one concerns on point set reconstruction on feature encoding of local connectivity, which first constructs a k-NN graph on the generated surface in the first stage and discovers correlation between neighboring points via graph convolution as the DGCNN [Wang *et al.*, 2018b]. Simply put, our method adopts simple point clouds to represent object shape and designs a stack of autoencoders to mine point-wise dependency. Figure 1 illustrates the key difference between our direct competitor – the AtlasNet [Groueix *et al.*, 2018] and the proposed DCG network, with visualizing results of some testing examples in our experiments.

2 Related work

Learning to Generate 3D Surface. A number of algorithms have been proposed for generating 3D surface of object shape from single images [Choy *et al.*, 2016; Girdhar *et al.*, 2016; Yan *et al.*, 2016; Tatarchenko *et al.*, 2017; Tulsiani *et al.*, 2017], image sequences [Choy *et al.*, 2016; Kar *et al.*, 2017], point sets [Fan *et al.*, 2017; Groueix *et al.*, 2018], or depth images [Yang *et al.*, 2017], which can be categorized into two categories dependent on using volumetric voxels or non-Euclidean parametric surface representations. On one hand, with a volumetric shape representation, supervised deep learning based algorithms [Choy *et al.*, 2016; Girdhar *et al.*, 2016; Tatarchenko *et al.*, 2017; Tulsiani *et al.*, 2017; Yan *et al.*, 2016; Tatarchenko *et al.*, 2016] for 3D shape reconstruction have been developed based on 3D Euclidean convolutional encoding and decoding along regularized grids. These volumetric CNNs designed in an encoder-decoder structure focused on either extracting good latent vector via inter-modality feature fusion [Girdhar *et al.*, 2016] and view-wise correlation mining [Choy *et al.*, 2016], or alleviating the inherent occupancy sparsity problemby replacing voxels with computationally efficient alternatives such as octree [Tatarchenko *et al.*, 2017], RGB-D [Tatarchenko *et al.*, 2016] or multi-view images [Yan *et al.*, 2016]. On the other hand, a non-Euclidean parametric representation such as point clouds [Fan *et al.*, 2017; Groueix *et al.*, 2018] and meshes [Wang *et al.*, 2018a; Pan *et al.*, 2018; Tang *et al.*, 2019] can be considered a powerful alternative, which avoids occupancy sparsity in the volumetric shape representation but the problem of operating a convolution on non-Euclidean data arises accordingly. The first pioneering work to generate a point-based surface with a deep net is the Point Set Generation (PSG) Network [Fan *et al.*, 2017], which encodes a single image into a latent vector to regress points' positions directly. Because of missing local connection in the representation, generated points' positions in the PSG net have a large variation when direct recovery of the object surface, which encourages Pixel2mesh [Wang *et al.*, 2018a] to regularize 3D shape by favoring for losses to enforce local smooth manifold structure. Most relevant to our work is the AtlasNet [Groueix *et al.*, 2018], which generates a point cloud representation of a surface via learning a regression mapping between encoded feature vectors from input data and surface parameters (points' positions) of 3D shape. The key difference between our method and the AtlasNet lies in two folds. First, our method favors for hierarchical coarse-to-fine learning in a cascade structure, while the AtlasNet has one stage to generate a parametric surface based on patches. Second, the AtlasNet learns implicitly to incorporate local connection between points via learning a mapping between vectors encoding shape and points' positions to its surface parameters. Beyond implicit feature encoding as the AtlasNet, our method also explicitly adopts graph convolution on non-Euclidean data inspired by [Bronstein *et al.*, 2017; Wang *et al.*, 2018b]. Experimental evaluation in Sec. 4.2 verifies superior efficacy of the proposed Deep Cascade Generation (DCG) to other competitors.

Geometric Deep Learning on Point Sets. Recently, a number of geometric deep learning methods are designed on non-Euclidean data especially point clouds. As pioneering works, the PointNet [Qi *et al.*, 2017a] and the Pointnet++ [Qi *et al.*, 2017b] start the trend of implementing deep learning on unordered point sets. The permutation invariance of point clouds is encoded by point-wise manipulation and a symmetric function for accumulating features, but failing to exploit point-wise connectivity. Recent progress on geometric deep learning such as spectral networks [Bruna *et al.*, 2013; Defferrard *et al.*, 2016; Kipf and Welling, 2016] and dynamic graph CNN (DGCNN) [Wang *et al.*, 2018b] inspire us to encode additional local connection into a feature vector, which provides low-dimensional manifold information to regularize surface parameterization. Our motivation is verified in Tables 1, 2, and 3.

Contributions. The novelties of our method are as follows.

- We develop a novel deep cascade learning to progressively evolves from coarse to fine point clouds, which can explicitly encodes their neighborhood information to locally refine point-based shape.
- An ensemble of refined point sets to construct a dense surface avoids local minima caused by complex combinatorial irregularities when exploiting point-wise correlation and also reduces computational costs, compared to directly generating a dense surface.

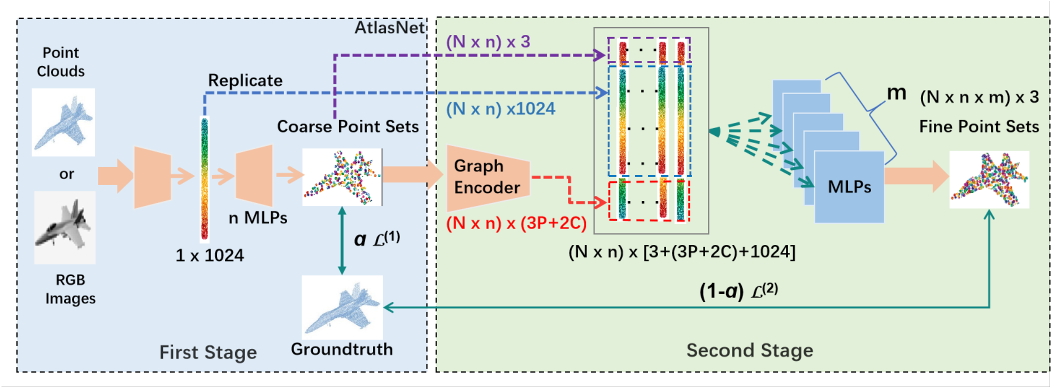


Figure 2: Pipeline of the proposed DCG Net consisting of two cascade stages – coarse shape generation and shape refinement. The former adopts the state-of-the-art AtlasNet, while the latter adopts graph convolution based encoding and an ensemble of decoders.

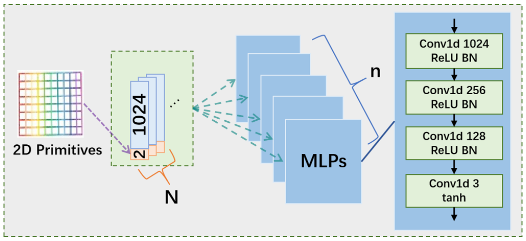


Figure 3: Network structure of the encoder-decoder in the first cascade stage following the state-of-the-art AtlasNet.

- A novel loss is designed to dynamically adjusting weights between losses on coarse and fine point clouds, which in principle enforces network optimization from losses on the output of cascade levels at different training stages.
- Our method significantly outperforms the state-of-the-art PSG [Fan *et al.*, 2017] and Atlasnet [Groueix *et al.*, 2018] on the public Shapenet benchmark on both single-view shape reconstruction and autoencoding tasks.

Source codes of our DCG method are available[1].

3 Methodology

We first formulate the surface generation problem on a parametric representation, i.e. a point cloud, into learning a mapping function $F(\cdot)$ from the input data $\mathcal{X}$ to the ground-truth surface $\mathcal{S}$. On point sets, the ground truth surface $\mathcal{S}$ can be approximated by a point cloud representation $\mathcal{P}^*$. The object function of point set generation can thus be written as

$$\min \quad \mathcal{L}(F(\mathcal{X}) - \mathcal{P}^*)$$

where $\mathcal{L}(\cdot)$ is the loss function. In the existing methods [Fan *et al.*, 2017; Groueix *et al.*, 2018] for generating a point-based shape, an encoder-decoder structure is popular. In details, $F(\cdot)$ can be decomposed into an encoder $E(\cdot)$ and a decoder $D(\cdot)$, i.e., $F(\mathcal{X}) = D(E(\mathcal{X}))$. Intuitively, the encoder $E(\cdot)$ encodes input data into a latent vector θ which is then decoded into 3D geometry to approximate $\mathcal{P}^*$.

In this section, we present the deep cascade generation (DCG) network on point sets, which consists of end-to-end trainable autoencoders. For generality, we define a cascade network with L stages, the l-th of which generates a point cloud representation $\mathcal{P}_l,\ l = 1, 2, \ldots, L$. In the first cascade level, $\mathcal{P}_1$ can be generated via the following equation:

$$\mathcal{P}_1 = F_1(\mathcal{X}),$$

while, in the remaining cascade levels ($l \geqslant 2$) for point set reconstruction,

$$\mathcal{P}_l = D_l([\mathcal{P}_{l-1}, \mathcal{H}_l(\mathcal{P}_{l-1}), \theta_1])$$

where D_l is a stack of decoders at stage l for generating 3d points; $\mathcal{P}_{l-1}$ denotes point sets to be refined (the purple arrow in Figure 2); $\mathcal{H}_l$ is a shape encoder on graph representation of $\mathcal{P}_{l-1}$ at the l-th stage to discover local correlation (the red arrow in Figure 2); θ_1 is the latent vector encoded in the first stage to represent a global feature on generating a coarse surface (the blue arrow in Figure 2).

For simplifying the network structure, we evaluate our deep cascade generation network with two cascade stages, one for generating a sparsely coarse shape (highlighted in a blue rectangle) and the other for shape refinement (highlighted in a green rectangle). We adopt the state-of-the-art AtlasNet [Groueix *et al.*, 2018] as the autoencoder in the first stage (see Sec. 3.1) and design the cascade structure and other factors favorable for coarse-to-fine point set generation. Specifically, there are three key components in our DCG net.

- Graph construction and feature encoding in the shape refinement stage to incorporate local connection of points (Sec. 3.2).
- An ensemble of refined parametric point sets to avoid local minima (Sec. 3.3).
- A dynamic loss function for enforcing the training procedure in a coarse-to-fine fashion (Sec. 3.4).

3.1 AtlasNet based Coarse Surface Generation

We adopt recent AtlasNet [Groueix *et al.*, 2018] to generate the coarse surface to be refined owing to its strong performance and efficiency. Figure 3 illustrates the deep structure

[1] https://wkqscut.github.io/DCGNet/.

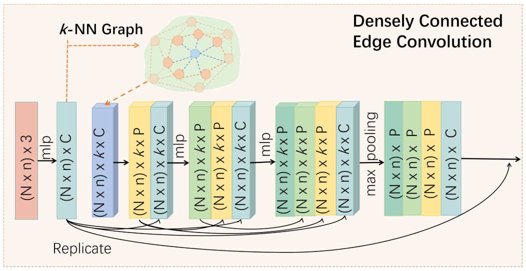


Figure 4: Net structure of a densely-connected graph encoder.

of the AtlasNet, which contains an encoder and n (e.g., five in our experiments) multi-layer perceptron (MLP) decoders, each of which with four fully connected layers aims to predict a parametric surface patch locally. We follow the settings in [Groueix *et al.*, 2018] for point set generation and shape autoencoding, i.e., ResNet-18 [He *et al.*, 2016] and PointNet [Qi *et al.*, 2017a] for feature encoding on images or point clouds respectively. Specifically, the ResNet-18 contains four residual blocks followed by one fully-connected layer, and each block consists of five 2D convolution layers, while the PointNet has four layers with three 1D convolution layers and one fully-connected layer. Inspired by the FoldingNet [Yang *et al.*, 2018], we use tiled N-dimensional 2D fixed grid points as 2D primitives during reconstruction rather than 2D points via uniformly random sampling, which together with the latent vector encoded by either ResNet-18 or PointNet are fed into the decoder as an input. The output dimension of hidden layers in each MLP based decoder is fixed to [1024, 512, 256, 3] followed by ReLU non-linearity operation. Finally, the output of the AtlasNet is a collection of $N \times n$ 3D points to represent a coarse surface of 3D object shape.

3.2 Graph Convolutional Encoding

Our motivation to exploit point-wise connectivity in a point cloud representation can be achieved via designing novel densely-connected MLPs (see Figure 4) on the predicted point set from the early cascade stage. We first extract a C-dimensional ($C = 24$) feature for each 3D point via one MLP layer, which contains one 1D convolution layer. A k nearest neighbor (k-NN) graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ in $\mathbb{R}^C$ containing $N \times n$ vertices $\mathcal{V}$ = $\{v_1,\ldots,v_{N\times n}\}$ and $\mathcal{E} \subseteq \mathcal{V}$ x $\mathcal{V}$ is constructed from an unstructured point feature set. We employ the edge convolution [Wang *et al.*, 2018b] on such a k-NN graph. If there exists an edge e_{ij} connecting a vertex v_i and its neighbor vertex v_j, we get an edge feature g_{ij} by applying a nonlinear function $h\{\cdot,\cdot\}$ with learnable parameters Θ on vertex v_i and edge e_{ij}. As a result, each vertex having k nearest neighbors will generate a P-dimensional feature as follows:

$$v'_i = \sum_{j\in\mathcal{N}(i)} h_{\mathbf{\Theta}}(v_i \,\|\, v_j - v_i) \in \mathbb{R}^P,$$

where $h_{\mathbf{\Theta}}$ denotes a MLP mapping and $\mathcal{N}(i)$ is a set of local neighbors' indexes around vertex v_i. Inspired by the densely connected networks [Huang *et al.*, 2017], the output of the graph convolution (the blue block in Figure 4) is fed into three 2D MLPs having 2D convolution layers with growth rate P = 12, whose layers are densely connected as Figure 4 shows. The output layer of such an encoder is a graph max pooling layer to take the maximum among the k vertex neighbors.

3.3 An Ensemble of Point Decoders

As shown in Figure 2, we employ a stack of decoders for a densely fine point-based surface, encouraged by the PointNet++ [Qi *et al.*, 2017b] for 3D shape analysis in a hierarchical learning fashion. Specifically, given a coarse surface $\mathcal{P}_{l-1}$ as an input, the surface output of the autoencoder at cascade level l is $\cup D_l^m$, where m is the size of point generators based on multi-layer perceptrons. We use the same network structure of the MLP in the AtlasNet, i.e. four 1D convolution layers with [1024, 512, 256, 3] hidden neurons respectively. Moreover, We apply residual skip-connections between two adjacent cascade levels, which ensures that the positions of coarser points can be propagated and updated through the entire network and incorporated for fine surface generation. Evidently, the size of points in such an ensemble learning manner is linearly proportional to the size m of stacked decoders, and thus evolves more dense surface with cascade levels l increases. For a fair evaluation in our experiments, we employ the identical shape representation at the final cascade level as comparative methods, which reduces the size of MLP in the coarse shape generation from twenty-five in [Groueix *et al.*, 2018] to five in our experiments. Consequently, learning parameters of decoders in original AtlasNet are reduced significantly (58.9% as shown in Table 1). Experiment results in Sec. 3.3 verify consistently a moderate improvement on generation performance by an ensemble of refined point sets.

3.4 A Dynamic Loss for Network Optimization

We strive for optimizing predicted shape $\mathcal{P}_L$ from the final cascade stage by minimizing the objective function as:

$$\min \; \mathcal{L}(\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_L, \mathcal{P}^*)$$

which can be decomposed into loss functions on predicted surface $\mathcal{P}_l$ at level l and the ground truth point cloud $\mathcal{P}^*$ as

$$\min \; \sum_{l}^{L} w_l \mathcal{L}^{(l)}(\mathcal{P}_l, \mathcal{P}^*)$$

where w_l is the weight for the l-th loss $\mathcal{L}^{(l)}$. Such a weighted loss connected with point predictions of the hidden and output layers are popular in recent deep learning methods [Yuan *et al.*, 2018; Huang *et al.*, 2017], therefore we design a dynamically weighting strategy to enforce coarse-to-fine network optimization with training time evolving, which shares similar concept as exponential decay on adjusting learning rate. Specifically, in our two-stage cascade model ($L = 2$), the weights w_1 and w_2 for their corresponding losses are as

$$w_1 = \alpha = e^{-\lambda k} \in (0, 1]; \quad w_2 = 1 - \alpha;$$

where k is the current number of iterations during training, λ is the decay rate of w_1, and α is the trade-off parameter between losses. In our experiments, we adopt the Chamfer distance (cd) for $\mathcal{L}^{(l)}, l = 1, 2$ with more details given in Sec. 4.1.

Methods	CD ↓	HD ↓	F1 ↑	Dec. Params
PSG	4.83	2.20	48.30	–
AtlasNet	4.64	2.03	47.51	4.29
DCG wo/Graph (ours)	4.26	**1.85**	60.31	1.72
DCG (ours)	**4.09**	1.88	**60.56**	1.76

Table 1: Comparative evaluation on single-view reconstruction with 2500 predicted points. The Chamfer distance (CD) is in units of 10^3. The Hausdorff distance (HD) is in units of 10. For F1-score (F1), we use a threshold $\tau = 1e - 3$. Parameter size of decoders (Dec. Params) is in units of 10^7.

With such an exponential-decay weighted loss, coarse-to-fine network training can be achieved. During training, the dynamic loss has higher weights for generating a sparsely coarse surface at the early stage in view of reducing difficulties of direct mapping to dense surface. Moreover, a good coarse shape as an initial state can further make shape refinement simpler. Increasing weights for the fine surface in the following cascade stage leads to progressively generating shape details when training procedure evolves. In Table 4, we report experimental results to compare the proposed dynamic loss with other settings of the loss function, which verify our motivation to dynamically adjusting weights between losses.

3.5 Implementation Details

For simplicity, we use an identical MLP containing four fully-connected layers with channels 1024, 512, 256 and 3 respectively. All layers except the final one have a composite block of consecutive operations includes convolution, batch normalization, and ReLU non-linearity, while the $\tanh$ is then applied to the final layer (refer to network visualization in Figure 3). We then present details about end-to-end network optimization. As shown in Figure 2, our DCG net training takes input data (images for single-view shape reconstruction and point clouds for shape autoencoding) and ground truth point cloud representations for model training. In single-view shape reconstruction, each training image with size 137×137 is randomly cropped to size 127×127 for data augmentation and then resized to 224×224 before feeding into the feature encoder at the cascade level 1. All the point clouds sampled from CAD models are normalized into a unit sphere. We used the ADAM to train the model for a total of 420 epochs with an initial learning rate of 0.001 and batch size 32. For step decay on the learning rate, it is dropped by a factor of 0.1 after 300 and 400 epochs.

4 Experiments

4.1 Settings

Dataset. We conduct experiments on the popular ShapeNet Core dataset (v2) [Chang *et al.*, 2015], which has been widely adopted in 3D shape reconstruction [Choy *et al.*, 2016; Fan *et al.*, 2017; Groueix *et al.*, 2018] and autoencoding [Yang *et al.*, 2018]. It contains 39689 CAD models belonging to 13 categories, which range from 1K to 10K samples. 30000 points are uniformly sampled from CAD models as the ground truth point cloud representation, but we randomly select 2500 points to supervise network training to avoid over-fitting. Moreover, in single-view reconstruction, 24 images from different viewing angles are rendered for each CAD model. In Figure 1, examples from the ShapeNet are visualized. We follow the settings in [Choy *et al.*, 2016; Groueix *et al.*, 2018], i.e., 31746 models for training and the remaining 7943 for testing.

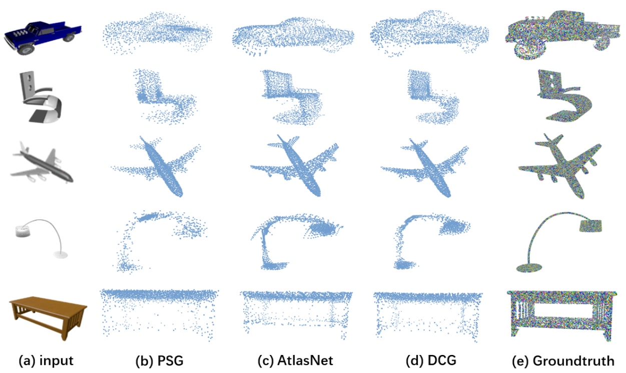


Figure 5: Single-view shape reconstruction comparison from an input image. (a) to generate a point cloud representation, (b) by PSG, (c) by AtlasNet, and (d) by our DCG.

Comparative Methods. We compare our method with two state-of-the-art methods, i.e., the PSG [Fan *et al.*, 2017] and the AtlasNet [Groueix *et al.*, 2018]. We utilize the two branches version of the PSG net to regress a total of 2500 points, which generates 768 points with deconvolution and 1732 points via two fully connected layers in the other branch. For the AtlasNet, we follow the settings in [Groueix *et al.*, 2018] and report results provided by the authors online[2], which are better than those in original AtlasNet paper [Groueix *et al.*, 2018]. Moreover, the AtlasNet generates 2500 points for its surface representation on 25 patches, each of which includes 100 points.

Evaluation Metrics. We evaluate the quality of predicted point clouds $\mathcal{P}$ to compare with the ground truth point clouds $\mathcal{P}^*$ by measuring the Chamfer distance (CD) [Fan *et al.*, 2017] and the Hausdorff distance (HD) [Tang *et al.*, 2009] respectively. Specifically, the Chamfer distance measures average matching distance of points in one set to the nearest points in the other set, while the Hausdorff distance for the maximum deviation between two sets. Moreover, F1 score (or termed as F-measure) introduced in [Wang *et al.*, 2018a] is also employed for the harmonic average of the precision and recall at a given threshold $\tau = 1e - 3$ on point sets.

4.2 Results

Comparison with State-of-the-Art. Experiment results to compare the proposed DCG net with the state-of-the-art PSG [Fan *et al.*, 2017] and AtlasNet [Groueix *et al.*, 2018] in Tables 1, 2, and 3 for single-view shape reconstruction, dense points inference, and shape autoencoding respectively. Our

[2] https://github.com/ThibaultGROUEIX/AtlasNet

Methods	pla.	ben.	cab.	car	cha.	mon.	lam.	spe.	fir.	cou.	tab.	cel.	wat.	mean
AtlasNet	1.93	3.00	3.56	3.07	4.07	4.57	11.41	7.81	1.93	4.04	3.68	2.94	3.37	4.01
DCG wo/Graph (ours)	**1.57**	2.35	3.05	2.73	3.37	**4.20**	9.93	**6.72**	1.40	**3.10**	3.22	2.21	2.74	3.38
DCG (ours)	1.62	**2.30**	**2.96**	**2.71**	**3.18**	4.45	**9.30**	7.28	**1.33**	3.24	**3.00**	**2.07**	**2.45**	**3.28**

Table 2: Single-view dense points inference using the Chamfer distance in units of 10^3, with 30000 predicted points adopted.

Methods	pla.	ben.	cab.	car	cha.	mon.	lam.	spe.	fir.	cou.	tab.	cel.	wat.	mean
PSG	1.47	1.98	2.46	1.98	2.28	2.44	4.25	3.63	2.07	2.46	2.29	1.77	2.87	2.36
AtlasNet	0.83	1.21	1.68	1.53	1.56	1.66	2.32	2.54	0.63	1.72	1.50	1.21	1.28	1.48
DCG wo/Graph (ours)	0.83	1.13	1.62	1.54	1.52	1.61	2.18	2.45	0.58	1.62	1.45	1.19	1.27	1.43
DCG (ours)	**0.77**	**1.05**	**1.54**	**1.46**	**1.44**	**1.54**	**2.02**	**2.39**	**0.55**	**1.53**	**1.33**	**1.14**	**1.20**	**1.35**

Table 3: Shape autoencoding using the Chamfer distance in units of 10^3, with 2500 predicted points adopted.

method can consistently achieve superior performance on all three performance metrics. Given the identical input data and point clouds sampled from the same CAD models, performance improvement can only be explained by the novel network structure of the DCG net. Qualitative results of comparative evaluation are illustrated in Figure 5, which shows our DCG can preserve the details of tiny object parts and constrain predicted points close to satisfy its geometric manifolds. Moreover, in comparison with the AtlasNet directly generating a surface at one stage, our coarse-to-fine hierarchical learning can significantly reduce the size of learning parameters in decoders (the right column of Table 1).

Effect of Graph Convolution Encoding. Tables 1, 2 and 3 also compare the proposed DCG with and without graph convolution operation. Specifically, to generate latent vector θ_2 at the second cascade stage, DCG and DCG wo/Graph denote the model structure whether it adds additional graph encoded feature from $\mathcal{P}_1$ (highlighted in a red dashed rectangle of Figure 2) or not. As shown in Tables 1, 2 and 3, our DCG net can beat its variant without graph convolutional encoding (i.e. the DCG wo/Graph) on the CD and F1 metrics, but perform comparable on the HD metric in Table 1. Different performance on the CD and HD can be caused by feature inconsistency on complex low-dimensional manifolds based on coarse point surface. However, without graph convolution encoding, our DCG still performs better than other competitors.

Effect of Cascade Structure. In this experiment on evaluating the ensemble structure of decoders, DCG network adopts 5×5 and 10×1 indicating the number of MLPs in each cascade stage. For example, $5{\times}5$ denotes five MLPs ($n = 5$) in the coarse shape generation and five ($m = 5$) for shape refinement. Note that, each MLP has the identical structure and our DCG with both settings generates the same size of point sets for a fair comparison. Our DCG method (i.e. results shown in Table 1 employing the $5{\times}5$ structure) can outperform its variant in the 10×1 structure by reducing 3.3% on the mean Chamfer distance.

Effect of Weighting Strategies in the Dynamic Loss. We further conduct one more experiment on evaluation of fixed, linear and exponential decay of weights in the dynamic loss, whose results are reported in Table 4. We can conclude that 1) jointly learning on losses from different cascade levels ($\alpha = 0.5$, linear and exponential decay) performs better than the only loss on the final point predictions ($\alpha = 0$); 2) weights decay favorable for coarse-to-fine network optimization achieves competitive performance compared to fixed weights in general. Such an observation verifies our motivation to design the dynamic loss.

	$\alpha = 0$	$\alpha = 0.5$	lin. decay	exp. decay
mean	4.41	4.26	4.18	**4.09**

Table 4: Single-view reconstruction comparison on weighting strategies of the dynamic loss. Reported results on the Chamfer distance in units of 10^3 are category-independently trained.

5 Conclusion

In this paper, we generates point-based surface in two cascade stages – coarse shape generation and shape refinement. An ablation study confirms that all components in the proposed DCG improve generation performance. On the ShapeNet dataset, our DCG net achieves the state-of-the-art performance – 4.09 and 1.35 on the CD metric for single-view shape reconstruction and shape autoencoding, which outperforms the AtlasNet by 8.7% – 18.2%.

Acknowledgements

This work is supported in part by the Program for Guangdong Introducing Innovative and Enterpreneurial Teams (Grant No.: 2017ZT07X183), the National Natural Science Foundation of China (Grant No.: 61771201), and the SCUT Program (Grant No.: D6192110).

References

[Bronstein *et al.*, 2017] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, pages 18–42, 2017.

[Bruna *et al.*, 2013] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[Chang *et al.*, 2015] 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.

[Choy *et al.*, 2016] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, pages 628–644, 2016.

[Defferrard *et al.*, 2016] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, pages 3844–3852, 2016.

[Fan *et al.*, 2017] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, pages 605–613, 2017.

[Girdhar *et al.*, 2016] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *ECCV*, pages 484–499, 2016.

[Groueix *et al.*, 2018] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *CVPR*, pages 216–224, 2018.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[Huang *et al.*, 2017] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017.

[Kar *et al.*, 2017] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *NIPS*, pages 365–376, 2017.

[Karpathy *et al.*, 2014] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014.

[Kazhdan *et al.*, 2006] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *SGP*, volume 7, 2006.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[Li *et al.*, 2018] Jiaxin Li, Ben M Chen, and Gim Hee Lee. So-net: Self-organizing network for point cloud analysis. In *CVPR*, pages 9397–9406, 2018.

[Pan *et al.*, 2018] Junyi Pan, Jun Li, Xiaoguang Han, and Kui Jia. Residual MeshNet: Learning to deform meshes for single-view 3d reconstruction. In *3DV*, pages 719–727, 2018.

[Qi *et al.*, 2017a] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017.

[Qi *et al.*, 2017b] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, pages 5099–5108, 2017.

[Simon *et al.*, 2018] Martin Simon, Stefan Milz, Karl Amende, and Horst-Michael Gross. Complex-YOLO: An euler-region-proposal for real-time 3d object detection on point clouds. In *ECCV*, pages 0–0, 2018.

[Tang *et al.*, 2009] Min Tang, Minkyoung Lee, and Young J Kim. Interactive hausdorff distance computation for general polygonal models. In *TOG*, page 74, 2009.

[Tang *et al.*, 2019] Jiapeng Tang, Xiaoguang Han, Junyi Pan, Kui Jia, and Xin Tong. A skeleton-bridged deep learning approach for generating meshes of complex topologies from single RGB images. *arXiv preprint arXiv:1903.04704*, 2019.

[Tatarchenko *et al.*, 2016] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Multi-view 3d models from single images with a convolutional network. In *ECCV*, pages 322–337, 2016.

[Tatarchenko *et al.*, 2017] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *ICCV*, pages 2088–2096, 2017.

[Tulsiani *et al.*, 2017] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *CVPR*, pages 2626–2634, 2017.

[Wang *et al.*, 2018a] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV*, pages 52–67, 2018.

[Wang *et al.*, 2018b] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *arXiv preprint arXiv:1801.07829*, 2018.

[Yan *et al.*, 2016] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *NIPS*, pages 1696–1704, 2016.

[Yang *et al.*, 2017] Bo Yang, Hongkai Wen, Sen Wang, Ronald Clark, Andrew Markham, and Niki Trigoni. 3d object reconstruction from a single depth view with adversarial learning. In *ICCV*, pages 679–688, 2017.

[Yang *et al.*, 2018] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *CVPR*, pages 206–215, 2018.

[Yuan *et al.*, 2018] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. PCN: Point completion network. In *3DV*, pages 728–737, 2018.