3D Shape Reconstruction from 2D Image with Disentangled Attribute Flow

Anonymous CVPR submission

Paper ID 2686

Abstract

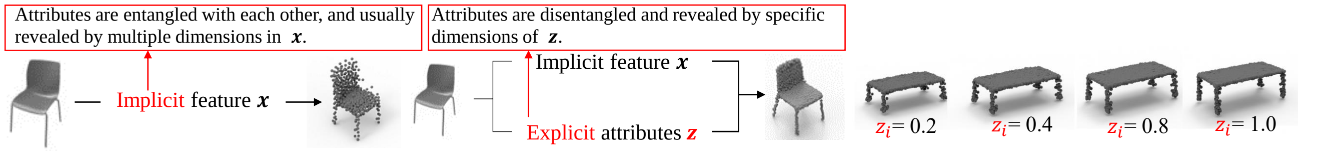
Reconstructing 3D shape from a single 2D image is a challenging task, which needs to estimate the detailed 3D structures based on the semantic attributes from 2D image, such as the length/bending of an airplane wings. So far, most of the previous methods still struggle to extract these semantic attributes in the 3D reconstruction process. Since the semantic attributes hidden in a single image are usually distributed in various semantic levels and entangled with each other, it is still challenging to reconstruct 3D shape with detailed structures represented in the input image. To address this problem, we propose 3DAttriFlow to explicitly disentangle and extract semantic attributes through different semantic levels in the input images. These disentangled semantic attributes will be integrated into the 3D shape reconstruction process, which can provide definite guidance to the reconstruction of specific attribute on 3D shape. As a result, the 3D decoder can explicitly capture high-level semantic features at the bottom of the network, and utilize low-level features at the top of the network, which allows to reconstruct more accurate 3D shapes, in terms of both overall and detailed shape structures. Note that the explicit disentangling is learned without extra labels, where the only supervision used in our training is the input image and its corresponding 3D shape. Our comprehensive experiments on ShapeNet dataset demonstrate that 3DAttriFlow outperforms the state-of-the-art shape reconstruction methods, and we also validate its generalization ability on shape completion task. Moreover, 3DAttriFlow also enables intermediate and semantic-aware shape manipulation by interpolating certain dimensions of the learned features.

1. Introduction

Reconstructing a 3D shape from a 2D image (2D-to-3D reconstruction) is a crucial task for bridging the gap between the 2D and 3D visual understanding. The typical paradigm is to firstly capture the semantic features of the 2D images through an image encoder, and then correctly reconstruct them in 3D space through a 3D decoder. Among the multiple representation forms of 3D shapes (i.e. voxel, point cloud and mesh), this paper mainly focuses on reconstructing 3D point cloud from the input image, due to its lightweight storage consumption and capability of representing various complicated shapes.

As addressed by the typical paradigm of most previous methods [15,26,31,35], the key of 2D-to-3D reconstruction is how to precisely interpret the semantic attribute from images into the 3D space. Thanks to the recent progress of 2D computer vision, there are many well-known methods (e.g. AlexNet [9], VGG [21] and ResNet [5]) to encode semantic attributes into image features, and their efficiencies have also been proved by a wide range of cross-modal tasks (e.g. image captioning [23, 37], cross-modal retrieval [24, 41]). However, for the research of 2D-to-3D reconstruction, how to interpret visual information from 2D domain to 3D domain for accurate 3D reconstruction still remains a difficult task. Because most previous methods [15,26,27,31,35,36] only rely on the feature channels (e.g. element-wise add, feature concatenation and attention mechanism) to convey the visual information from the image encoder to the 3D decoder, which only contains implicit geometric information with limited semantic attributes as the guidance to shape reconstruction. For example, an overall geometric information such as the number of legs will determine the table to have three or four legs. Such geometric information can be easily noticed and reconstructed by the decoder. On the other hand, the detailed semantic attributes like the length or bending of legs will specifically determine the detailed shape of these legs. However, since these semantic attributes are deeply entangled with each other in image features, they can hardly be noticed by the decoder during the reconstruction process.

Moreover, semantic attributes are usually distributed at various semantic levels, and entangle with each other throughout the pyramidal hierarchy of image encoder. As a result, they can hardly be fully exploited through implicit feature channels. As a result, the previous methods usually suffer from guiding the decoder to reconstruct various visual information extracted by the encoder, which leads to the insufficient usage of semantic features for predicting 3D


Attributes are entangled with each other, and usually revealed by multiple dimensions in $\boldsymbol{x}$.
Implicit feature $\boldsymbol{x}$
Attributes are disentangled and revealed by specific dimensions of $\boldsymbol{z}$.
Implicit feature $\boldsymbol{x}$
Explicit attributes $\boldsymbol{z}$
z_i= 0.2
z_i= 0.4
z_i= 0.8
z_i= 1.0

(a) Previous method decodes 3D shape based on implicit feature $\boldsymbol{x}$.

(b) 3DAttriFlow decodes 3D shape based on implicit feature $\boldsymbol{x}$ and explicit attributes $\boldsymbol{z}$.

(c) The i$^{\text{th}}$ dimension z_i of $\boldsymbol{z}$ learned by 3DAttriFlow can control the length of chair legs.

Figure 1. Comparison between previous methods (a) and our 3DAttriFlow (b). Besides the implicit feature $\boldsymbol{x}$ of input image, 3DAttriFlow learns an additional attribute code $\boldsymbol{z}$, which can explicitly reveal a specific semantic attribute of 3D shape in (c).

shapes in the previous methods.

A straightforward solution to this problem is to build numbers of feature channels between the decoder and all the network layers in the encoder, which will increase the cost of tremendous computational time and network complexity. On the other hand, as proved by many methods of image-to-image translation (e.g. image super-resolution [29, 34], image style transferring [42]), we notice that the global feature is able to encode most of the semantic attributes for a single image, as they can be used for high quality image generation/restoration task. Therefore, a promising solution is to explore deeply into the global features extracted from the 2D images, and decode the abundant semantic attributes embedded in the global features, which will provide a detailed and definite guidance to the reconstruction process of 3D shapes. Following the above-mentioned intuition, we propose a novel neural network, named 3DAttriFlow, to explicitly decompose the semantic attributes from the 2D image, and utilize these semantic attributes for 3D shape reconstruction in a controllable way.

Specifically, as shown in Figure 1, compared with the previous methods (Figure 1(a)) that only learn to reconstruct 3D shapes from an implicit image feature, the proposed 3DAttriFlow (Figure 1(b)) learns to decompose an additional attribute code to explicitly express certain semantic attributes. As a result, semantic attributes can be manipulated by interpolating specific dimensions of the attribute code (Figure 1(c) shows the manipulation of the length of the chair legs). Such process is accomplished by the *attribute flow pipe* proposed in 3DAttriFlow. By piping disentangled semantic attributes into the hierarchical generation process of point clouds through the attribute flow pipe, the decoder is able to selectively interpret semantic attributes following the hierarchy of semantic levels.

Our idea is inspired by the recent generative method of EigenGAN [6], which learns to manipulate explicit semantic attributes of human faces in an unsupervised way. However, due to the discrete nature of point clouds, the coordinates of points are merely organized in an unordered manner, which is in contrast with the image pixels arranged in an ordered grid structure. Such nature of point clouds makes the location of each point unpredictable during the generation process, until the 3-dimensional coordinates are finally revealed at the end of the decoder. Therefore, a direct implementation of EigenGAN [6] based decoder may result in failure, because the network cannot accurately predict which semantic attribute should be assigned to a specific point without knowing the location of that point. To address this problem, we propose the *deformation pipe* as the solution, which follows the idea of PMP-Net [32] to reconsider the shape generation process as a shape deformation process. That is, each point is first assigned a prior location in 3D space, and then moved to their destination to regroup as a new shape. Specifically, 3DAttriFlow moves the point cloud sampled from a 3D sphere into the target shape indicated by the 2D images. In all, our main contributions are summarized as follows.

- We propose a novel deep network, named 3DAttriFlow, for reconstructing high-quality 3D shapes from single 2D images. Compared with the previous methods, 3DAttriFlow can interpret explicit semantic attributes from images, and effectively use them to guide the decoder for detailed and high-quality 2D-to-3D shape reconstruction.

- We propose the attribute flow pipe to explicitly disentangle the semantic attributes embedded in the global feature of 2D image, which can provide definite guidance about the detailed reconstruction of semantic attributes to the 3D decoder, leading to more accurate prediction of 3D shape in terms of both overall and detailed shape structures.

- We propose the deformation pipe to offer the location priors to attribute flow pipe, where the extracted semantic attributes can be assigned to a specific point by leveraging the location of that point. As a result, 3DAttriFlow avoids the problem of assigning semantics to unordered data, and allows more accurate feature integration between the attribute flow pipe and the deformation pipe.

2. Related Work

Reconstructing 3D shapes from 2D images can be categorized according to the number of input images as: single-view 3D shape reconstruction [4, 15, 26, 27, 31] and multi-view 3D shape reconstruction [31, 35, 36]. On the other hand, according to different representation forms of 3D shapes, the related work can also be categorized as voxel-based 3D shape reconstruction [15, 27, 35, 36], point cloud based 3D shape reconstruction [3, 4, 13, 25] and mesh-based 3D shape reconstruction [26, 31]. Specifically, the proposed 3DAttriFlow in this paper belongs to the single-view 3D shape reconstruction, which is based on point clouds. The discussion of related work will be organized according to the output forms of 3D shape for convenience.

Point cloud based methods. With the rapid development of point cloud representation learning [12, 19, 20, 30], which is triggered by the pioneering work of PointNet [19], point cloud generation has been widely studied in recent years, and boosted the research of reconstructing point clouds from 2D images. Most of the point cloud based methods [3, 4, 7, 16] follow the generative way to predict the point coordinates based on the 2D images, where their efforts are made either to improve the feature communications between image encoder and 3D shape decoder [3], or impose extra supervision/constraint on the generated point clouds [7, 16, 39].

Voxel/mesh based methods. As for voxel-based reconstruction methods, the grid structure of 3D voxels is naturally applied in convolutional neural network, which simplifies the problem as translating 2D grid data to 3D grid data. Typical practice along this line is to directly utilize the CNN structure in both 2D and 3D domain, which aims to extract 2D grid feature from the input image, and reconstruct the corresponding 3D grid shape. Typical methods like 3DR2N2 [2], Pix2Vox [35] and Pix2Vox++ [36] have comprehensively explored the 3D reconstruction performance using single or multiple images as inputs. However, suffering from the cubic growth of input voxel data, the resolution for voxel data is usually limited, while further increasing the resolution will lead to unacceptable computational cost. As for mesh based methods, most of them follow the idea of deforming from a prior shape. For example, Pixel2Mesh [26] and its successor Pixel2Mesh++ [31] consider to deform an ellipsoid mesh into a target shape, which is combined with a multi-stage fusion strategy to introduce image features into the mesh deformation network. Li et al. [11] further extend such framework to capture the semantic part of object in 2D images. Pan et al. [17] improve the ability to generate complex shape by deforming mesh while modifying its typology. However, the intersection of meshes and the hypothesis of manifold surface will hinder the generation of 3D shape with inner or irregular structures.

Discussion. The reconstruction of 3D shapes from 2D images requires the deep understanding of semantic attributes in 2D images, and the correct interpretation of semantic attributes in 3D space. The above-mentioned methods either choose to directly decode the 3D shape from a global feature, or rely on feature channels to bridge the network layers between image encoder and the shape decoder. The problem is that, all these practices can only convey the implicit features from 2D images to 3D shapes, resulting in ambiguous guidance to reconstruct specific and detailed semantic attributes of 3D shape. Different from these previous methods, 3DAttriFlow proposes the solution to directly decompose the semantic attributes from the image feature, and integrate them into the shape reconstruction process, which can offer a definite guidance to the reconstruction of specific semantic attribute according to the 2D image. Moreover, the ability of attribute decomposition in 3DAttriFlow enables the decoder to flexibly reconstruct the semantic attributes following the hierarchy of semantic levels, which is in contrast to the network with fixed channels that only allows decoder to learn from fixed layers of encoder.

3. Architecture of 3DAttriFlow

The overall architecture of 3DAttriFlow is shown in Figure 2, which reconstructs a 3D point cloud with N points according to the input image. 3DAttriFlow mainly consists of two pipelines as follows. (1) The *attribute flow pipe* (see Figure 2(a)) serves to disentangle the semantic attributes from the input feature, which is usually the global feature extracted by an image encoder. (2) The *deformation pipe* (see Figure 2(b)) serves to deform the initial point cloud sampled from a 3D sphere into the target shape, which is guided by the semantic attributes from the attribute flow pipe. The structures for each pipeline are detailed below.

3.1. Attribute Flow Pipe

As shown in Figure 2(a), the attribute flow pipe aims to extract geometric code $\{\boldsymbol{\sigma}, \boldsymbol{\mu}\}$ and semantic features $\boldsymbol{s}_i$ step-by-step from the image feature $\boldsymbol{x}$ and sphere point cloud $\{\boldsymbol{p}_k\}$, where i denotes the $\boldsymbol{i}^{\text{th}}$ step. Then, the extracted feature and code will be integrated into the deformation pipe to guide the deformation of the spherical point cloud $\{\boldsymbol{p}_k\}$. The basic architecture of attribute flow pipe consists of a feature extractor and three attribute flow modules (AF module). Specifically, for the input image, 3DAttriFlow uses ResNet18 to extract an image feature $\boldsymbol{x}$ from the input image. Then, the AF module extracts and interprets the visual information from the image feature $\boldsymbol{x}$ to the geometric information and semantic attributes, which is accomplished by the geometric sub-pipe and the semantic sub-pipe, as shown in Figure 2(c).

Geometric sub-pipe. The geometric sub-pipe aims to interpret the overall visual information from images into the geometric information, which can be utilized for 3D shape reconstruction by the deformation pipe. Inspired by the style transferring based generative methods [8, 10], which learns the *local styles* from the latent random vector, we propose to interpret the visual information encoded by image feature $\boldsymbol{x}$ into the *geometric styles* $\{\boldsymbol{\sigma}_i, \boldsymbol{\mu}_i\}$, which is

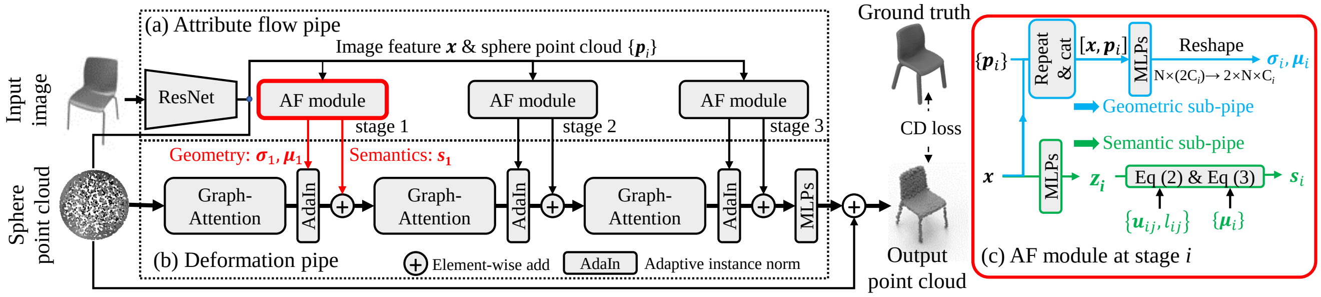


Figure 2. The overall architecture of 3DAttriFlow. 3DAttriFlow consists of two pipelines: (a) the attribute flow (AF) pipe extracts geometric code and semantic features based on the input image and initial sphere point cloud; (b) the deformation pipe deforms the point cloud according to the output of attribute flow pipe into the target shape. The detailed structure of AF module is shown in (c).

according to the location prior given by the initial point cloud $\{\boldsymbol{p}_k\}$. As shown by the geometric sub-pipe in Figure 2(c), at stage i, the image feature $\boldsymbol{x}$ is first repeated and concatenated with location priors $\{\boldsymbol{p}_k\}$ as $\{[\boldsymbol{x} : \boldsymbol{p}_k]\}$, where ":" denotes the feature concatenation. Then, followed by several multi-layer perceptrons (MLPs) and reshape operation, the image feature coupled with location priors is interpreted as geometric styles $\{\boldsymbol{\sigma}_i | \boldsymbol{\sigma}_i \in \mathbb{R}^{N\times C_i}\}$ and $\{\boldsymbol{\mu}_i | \boldsymbol{\mu}_i \in \mathbb{R}^{N\times C_i}\}$, where C_i denotes the dimension of point features in deformation pipe at stage i.

Semantic sub-pipe. The semantic sub-pipe aims to decompose explicit semantic attributes from the image feature $\boldsymbol{x}$, and represent them by the activation at certain dimension of attribute code $\boldsymbol{z}$. As a result, the deformation pipe can produce a precise 3D semantic attribute under the definite guidance given by the attribute code. Specifically, as shown in the lower-branch of Figure 2(c), at stage i, the semantic sub-pipe first squeezes the image feature $\boldsymbol{x}$ into attribute code $\boldsymbol{z}_i$ as:

$$\boldsymbol{z}_i = \phi(\boldsymbol{x}|\theta_i), \tag{1}$$

where ϕ denotes the MLP layer, and θ_i denotes the weights of MLP layer for generating $\boldsymbol{z}_i$. According to He et al. [6], for the activation z_{ij} at j^{th} dimension of the attribute code $\boldsymbol{z}_i$, an orthogonal basis $\boldsymbol{u}_{ij} \in \mathbb{R}^{N\times C_i}$ from a linear subspace $\mathcal{U}_i = \{\boldsymbol{u}_{ij}\}$ will be use to discover the semantic attribute $\hat{\boldsymbol{z}}_{\boldsymbol{ij}}$ lying behind z_{ij} as:

$$\hat{\boldsymbol{z}}_{\boldsymbol{ij}} = z_{ij} l_{ij} \boldsymbol{u}_{ij}, \tag{2}$$

where l_{ij} is a learnable weight denoting the significance of the semantic attribute discovered by the orthogonal basis $\boldsymbol{u}_{ij}$. By adding the semantic attributes $\hat{\boldsymbol{z}}_{\boldsymbol{j}}$ across all dimensions of attribute code $\boldsymbol{z}_i$, the semantic sub-pipe outputs the semantic feature $\boldsymbol{s}_i$ encoded with explicit attribute information, which is formulated as:

$$\boldsymbol{s}_i = \sum_j \hat{\boldsymbol{z}}_{ij} + \boldsymbol{\mu}_i, \tag{3}$$

where $\boldsymbol{\mu}_i$ is a learnable bias. The semantic feature $\boldsymbol{s}_i$ will be flowed into the deformation pipe to guide the reconstruction of 3D semantic attributes.

3.2. Deformation Pipe

The architecture of the deformation pipe is shown in Figure 2(b). The input at the bottom of the deformation pipe is a point set $\mathcal{P} = \{\boldsymbol{p}_i\}$, which is uniformly sampled from a 3D sphere. Note that we choose sphere as a starting shape because each point on the sphere can be regarded as a L2-regularized vector, which guarantees an isotropic shape prior input to the network. The output at the top of the deformation pipe is a set of displacement vector $\{\Delta \boldsymbol{p}_i\}$. The output of the deformation pipe is a deformed point set $\mathcal{P}^o = \{(\boldsymbol{p}_i + \Delta \boldsymbol{p}_i)\}$, which has the same shape as target point cloud $\mathcal{P}^t = \{\boldsymbol{p}_j^t\}$.

To predict the displacement vector $\{\Delta \boldsymbol{p}_i\}$ for each point, we follow Wang et al. [30] to extract point features from multiple input point set P through the graph attention modules, which forms a three-stages point feature learning framework. At i^{th} stage, the deformation pipe takes both the geometric styles $\{\boldsymbol{\sigma}_i, \boldsymbol{\mu}_i\}$ and semantic feature $\boldsymbol{s}_i$ as input, and infers the displacement for each point, according to the geometric information and semantic attribute interpreted from the image feature. For convenience, we denote the point features generated in stage i as $\mathcal{Q}^i = \{\boldsymbol{q}_k^i\}$.

For the geometric styles $\{\boldsymbol{\sigma}_i, \boldsymbol{\mu}_i\}$, we follow the practice of style transfer [8] to introduce the adaptive instance normalization, which is used to adapt point features according to the geometric information encoded in the geometric styles. The formulation is given as:

$$\hat{\boldsymbol{q}}_k^i = \boldsymbol{\sigma}_{ik} \cdot \frac{\boldsymbol{q}_k^i - \mu(\boldsymbol{q}_k^i)}{\sigma(\boldsymbol{q}_k^i)} + \boldsymbol{\mu}_{ik}, \tag{4}$$

where $\mu(\boldsymbol{q}_k^i)$ and $\sigma(\boldsymbol{q}_k^i)$ denote the mean and deviations of $\boldsymbol{q}_k^i$ estimated by moving average algorithm, respectively. $\boldsymbol{\sigma}_{ik}$ and $\boldsymbol{\mu}_{ik}$ denote the vector at k^{th} row of $\boldsymbol{\sigma}_i$ and $\boldsymbol{\mu}_i$, respectively.

After the adaptation of point feature according to the geometric styles, the semantic feature $\boldsymbol{s}_i$ is integrated into the point feature $\hat{\boldsymbol{q}}_k^i$ through MLP layer and element-wise add, given as:

$$\hat{\boldsymbol{q}}_k^i \leftarrow \hat{\boldsymbol{q}}_k^i + \phi(\boldsymbol{s}_i|\theta_{\boldsymbol{s}_i}). \tag{5}$$

Table 1. 2D-to-3D reconstruction on ShapeNet dataset in terms of per-point L1 Chamfer distance $\times 10^2$ (lower is better).

Methods	Average	Plane	Bench	Cabinet	Car	Chair	Display	Lamp	Loud.	Rifle	Sofa	Table	Tele.	Vessel
3DR2N2 [2]	5.41	4.94	4.80	4.25	4.73	5.75	5.85	10.64	5.96	4.02	4.72	5.29	4.37	5.07
PSGN [3]	4.07	2.78	3.73	4.12	3.27	4.68	4.74	5.60	5.62	2.53	4.44	3.81	3.81	3.84
Pixel2mesh [26]	5.27	5.36	5.14	4.85	4.69	5.77	5.28	6.87	6.17	4.21	5.34	5.13	4.22	5.48
AtlasNet [4]	3.59	2.60	3.20	3.66	3.07	4.09	4.16	4.98	4.91	2.20	3.80	3.36	3.20	3.40
OccNet [15]	4.15	3.19	3.31	3.54	3.69	4.08	4.84	7.55	5.47	2.97	3.97	3.74	3.16	4.43
3DAttriFlow(Ours)	**3.02**	**2.11**	**2.71**	**2.66**	**2.50**	**3.33**	**3.60**	**4.55**	**4.16**	**1.94**	**3.24**	**2.85**	**2.66**	**2.96**

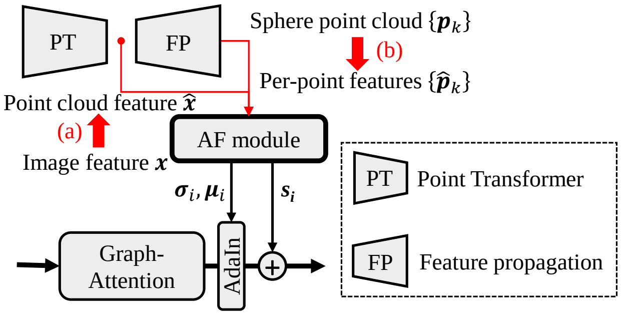

PT
FP
Sphere point cloud $\{\boldsymbol{p}_k\}$
(b)
Per-point features $\{\hat{\boldsymbol{p}}_k\}$
Point cloud feature $\hat{\boldsymbol{x}}$
(a)
Image feature $\boldsymbol{x}$
AF module
$\boldsymbol{\sigma}_i, \boldsymbol{\mu}_i$
$\boldsymbol{s}_i$
Graph-Attention
AdaIn
PT
Point Transformer
FP
Feature propagation

Figure 3. Illustration of extending 3DAttriFlow to 3D shape completion task. This is achieved by (a) replacing the input of attribute flow pipe with the global point cloud feature learned by the PointTransformer, and (b) by replacing the image feature with the per-point feature learned by feature propagation module, respectively.

At the top of the deformation pipe, we use MLP layers to transform point features into 3-dimensional displacement vectors $\{\Delta\boldsymbol{p}_k\}$, and finally output the deformed shape as $\{\boldsymbol{p}_k + \Delta\boldsymbol{p}_k\}$.

3.3. Extension to Shape Completion

3DAttriFlow can also be used to predict the missing part of an incomplete shape, which can be achieved by replacing the image encoder with the 3D point cloud encoder (such as PointTransformer [40]) in the attribute flow pipe. As a result, the input image feature $\boldsymbol{x}$ is replaced by the point cloud feature $\hat{\boldsymbol{x}}$. Inspired by PMP-Net [32], we find that the per-point features of incomplete point cloud can be used as the location prior, which is more self-adaptive compared with the fixed spherical point cloud. Therefore, we replace the sphere point cloud $\{\boldsymbol{p}_k\}$ with the per-point features $\{\hat{\boldsymbol{p}}_k\}$, which are learned by the feature propagation module specified in PointNet++ [20]. The modification to the attribute flow pipe is illustrated in Figure 3. To further improve the completion performance, we follow the coarse-to-fine strategy adopted by most of the completion methods [18, 33] to introduce an additional refining module from VRCNet [18], which aims to refine the detailed shape of predicted point clouds.

3.4. Training loss

The orthogonality of $\mathcal{U}_i$ is guaranteed by the regularization of orthogonality loss, which is defined as:

$$\mathcal{L}_{\text{Orth}} = \sum_{i \in 1,2,3} \|\mathcal{U}_i^{\text{T}}\mathcal{U}_i\| - 1. \tag{6}$$

The deformed shape conditioned by images and incomplete shapes is regularized by the ground truth point cloud through Chamfer distance (CD) defined as:

$$\begin{aligned}\mathcal{L}_{\text{CD}}(\mathcal{P}^{\text{o}}, \mathcal{P}^{\text{t}}) = & \frac{1}{2N}\sum_{\boldsymbol{p}^{\text{o}} \in \mathcal{P}^{\text{o}}} \min_{\boldsymbol{p}^{\text{t}} \in \mathcal{P}^{\text{t}}} \|\boldsymbol{p}^{\text{o}} - \boldsymbol{p}^{\text{t}}\|_2 \\ & + \frac{1}{2N}\sum_{\boldsymbol{p}^{\text{t}} \in \mathcal{P}^{\text{t}}} \min_{\boldsymbol{p}^{\text{o}} \in \mathcal{P}^{\text{o}}} \|\boldsymbol{p}^{\text{t}} - \boldsymbol{p}^{\text{o}}\|_2.\end{aligned} \tag{7}$$

The total training loss is formulated as

$$\mathcal{L} = \mathcal{L}_{\text{CD}} + \alpha\mathcal{L}_{\text{Orth}}, \tag{8}$$

where α is a balance factor to determine the weight of $\mathcal{L}_{\text{Orth}}$. In this paper, α is set to 100 for all experiments.

4. Experiments

In this section, we experimentally evaluate the effectiveness of 3DAttriFlow in 2D-to-3D reconstruction task, and analyze its generalization ability through point cloud completion task. The ablation studies will focus on the effectiveness of each part of 3DAttriFlow, and visually analyze the extracted semantic attributes by shape manipulation.

4.1. 2D-to-3D Reconstruction on ShapeNet dataset

Dataset briefs and evaluation metric. We follow the experimental settings of OccNet [15] to evaluate our 3DAttriFlow on ShapeNet dataset [1]. The whole dataset consists of 43,783 mesh object with 13 categories, which will be divided into training, validation and testing following the same strategy of OccNet [15]. Since our method focuses on the reconstruction of 3D point cloud from 2D images, we follow AtlasNet [4] to uniformly sample 30k points on the mesh surface of 3D object as the ground truth for training. Following previous methods [15,26], we use L1 Chamfer distance described by Eq. (7) as the evaluation metric. In order to compare with other methods of reconstructing 3D voxel or mesh, we follow OccNet [15] to sample 2,048 points from their output surface, and then calculate the L1 Chamfer distance with the ground truth. As for voxel-based methods, we additionally transfer their voxel output into mesh, then apply the point cloud sampling on the mesh surface.

Table 2. Point cloud completion on MVP dataset in terms of per-point L2 Chamfer distance ×10^4 (lower is better).

Methods	Average	Plane	Cabinet	Car	Chair	Lamp	Sofa	Table	Water.	Bed	Bench	Shelf	Bus	Guitar	Motor.	Pistol	Skate.
PCN [38]	9.80	4.22	8.92	6.49	12.46	19.54	9.92	12.45	8.78	19.0	9.0	13.39	5.15	1.87	6.03	6.04	4.70
TopNet [22]	10.34	4.09	9.71	7.36	13.46	20.53	11.21	12.46	8.50	18.98	8.58	15.15	5.47	2.13	7.19	7.33	4.15
MSN [14]	7.98	2.59	8.86	6.54	10.22	12.64	9.08	9.69	7.08	15.58	6.38	11.31	5.23	1.37	4.63	4.72	3.06
CRN [28]	7.34	2.45	8.62	5.97	8.95	11.16	8.63	9.30	6.43	14.93	6.11	10.39	4.97	1.67	4.33	4.47	3.39
VRCNet [18]	5.96	2.17	7.83	5.52	7.31	8.29	7.42	7.07	5.15	11.18	4.76	7.03	4.40	1.15	3.75	3.54	2.31
PMPNet [32]	6.24	1.99	8.84	6.36	7.77	6.18	8.72	7.71	5.19	11.77	5.07	8.34	5.27	1.27	3.95	3.57	2.35
SnowflakeNet [33]	5.86	2.04	7.76	5.61	7.07	7.42	6.92	7.13	5.05	11.32	4.87	7.72	4.46	1.16	3.94	3.52	3.64
3DAttriFlow(Ours)	**5.06**	**1.59**	**7.40**	**5.44**	**6.05**	**5.01**	**6.81**	**6.14**	**4.25**	**10.62**	**3.73**	**6.53**	**4.30**	**0.95**	**3.27**	**2.78**	**1.78**

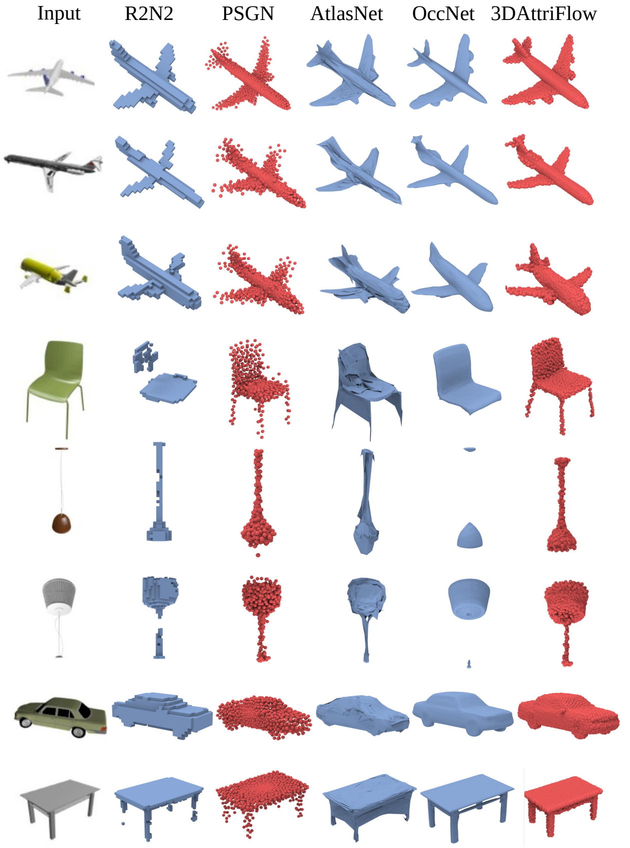


Figure 4. Visual comparison of 2D-to-3D reconstruction results with different methods under ShapeNet dataset.

Quantitative comparison. The results of 2D-to-3D reconstruction are shown in Table 1, in which 3DAttriFlow achieves the superior performance over the other compared counterpart methods. Especially, PSGN [3] and AtlasNet [4] are point cloud based methods, which are most relevant to 3DAttriFlow. However, 3DAttriFlow achieves more than 25% performance gain over these two methods. As we discussed in Sec. 1, the above-mentioned two methods adopt the typical paradigm for 2D-to-3D reconstruction, where AtlasNet [4] directly decodes the whole shape based on the implicit input of global feature, and PSGN [3] exploits the feature channels between encoder and decoder for introducing various levels of semantics. None of these practices can learn the explicit semantic features from the image, but only tries to decode shapes from implicit global feature or intermediate layers of encoder. In contrast, 3DAttriFlow can exploit both implicit and explicit semantic attributes learned from the image, which is through the geometric sub-pipe and semantic sub-pipe, respectively. As a result, 3DAttriFlow is able to predict the details of 3D shape based on more definite guidance from the explicit semantic attributes, and achieves better performance than its counterparts.

Qualitative comparison. The visual comparison of 2D-to-3D reconstruction is shown in Figure 4. Note that for AtlasNet, we follow its original visualization settings to exhibit the reconstructed mesh instead of point cloud. Compared with the other methods, 3DAttriFlow reconstructs the better details on a wide range of object categories. For example, on the chair category (the 5^{th} row of Figure 4), the legs are missing in the prediction of OccNet, while the chair predictions of PSGN and AtlasNet are ambiguous and full of noise. As for the plane category, both PSGN and AtlasNet fail to reconstruct the detailed shape of engines in the 1^{st} and the 2^{nd} row, while OccNet cannot make the correct prediction of engines steadily (failure in the 2^{nd} row).

4.2. 3D Completion on MVP dataset

Dataset briefs and evaluation metric. We follow the experimental settings of VRCNet [18] to evaluate our 3DAttriFlow on MVP dataset [18]. The dataset consists of 16 categories of incomplete/complete point clouds generated by models selected from ShapeNet, and is then divided into training set (62,400 shape pairs), and testing set (41,600 shape pairs). Following previous methods [18, 22, 32], we use L2 Chamfer distance as the evaluation metric.

Quantitative comparison. The comparison of point cloud completion is shown in Table 2. Compare with the current state-of-the-art completion method SnowflakeNet [33], 3DAttriFlow further improves the performance by 13.7% in terms of L2-CD. The intuition behind the completion task is the same as the 2D-to-3D reconstruction task, which is to predict a 3D shape based on a given input. In the case of point cloud completion, the input is the incomplete 3D shape. The better performance achieved by 3DAttriFlow can be dedicated to more comprehensive and explicit understanding about the semantic attributes, which is through the semantic sub-pipe in AF module. For example, in order to infer the length of an missing chair legs, a semantic code explicitly controlling such attribute is able to guide the de-

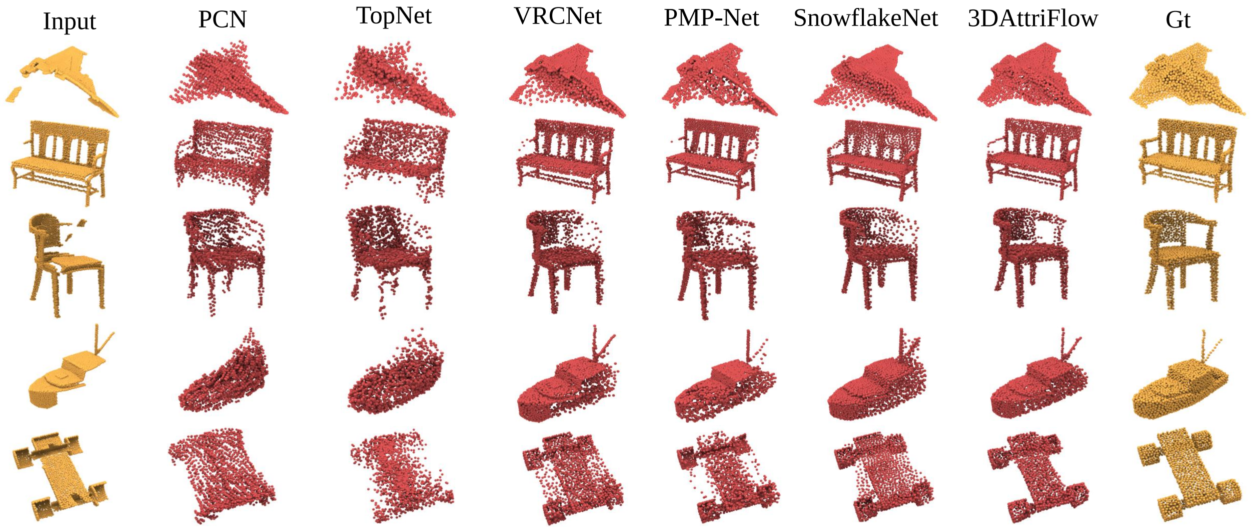


Figure 5. Visual comparison of point cloud completion results with different methods under MVP dataset.

coder to make a more precise prediction. In contrast, for the compared methods in Table 2, their decoders have to make the prediction from the implicit features, where the attribute of legs are entangled with the others in the implicit features.

Qualitative comparison. In Figure 5, we qualitatively compare 3DAttriFlow with the other completion methods on MVP dataset, from which we can find that 3DAttriFlow produces more precise and consistent complete shapes than other methods. Take the completion of chairs in the 2nd and the 3rd rows as examples, the predictions of chair-back and the missing chair armrest made by 3DAttriFlow are apparently better than the other methods. As for the skateboard in the 5th row, all of the five compared methods mess the wheels with the board, while 3DAttriFlow can produce a clean and detailed shape of the target skateboard.

4.3. Ablation Studies

In this subsection, all quantitative analysis results are typically conducted under four categories (i.e. plane, car, chair and table). By default, all the experimental settings are kept the same as in Sec. 4.1, except for modified part described in each ablation experiment below.

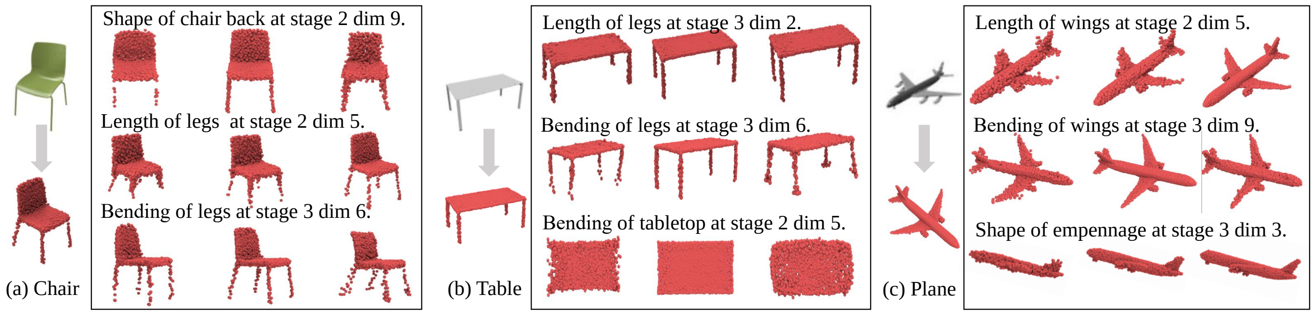
Analysis of each sub-pipe in AF module. We analyze the effectiveness of each sub-pipe of 3DAttriFlow by removing/replacing modules from the original network structure (denoted as *Full*). Specifically, we develop four different variations for comparison: (1) *w/o semantic sub-pipe* is the variation removing semantic sub-pipe from the AF module; (2) *w/o geometric sub-pipe* is the variation removing geometric sub-pipe from the AF module; (3) *semantic MLPs* is the variation replacing semantic sub-pipe with simple MLP layer, where the output is directly added to the features in deformation pipe; (4) *geometric MLPs* is the variation replacing geometric sub-pipe with simple MLPs, where the output is added to the features in deformation pipe. The results are shown in Table 3, from which we can find that our Full model achieves the best results over all four variations. Such result proves the effectiveness of each part to 3DAttriFlow.

Moreover, we additionally address two conclusions. First, by comparing w/o geometric sub-pipe and w/o semantic sub-pipe to the Full model, we can find that semantic sub-pipe has a relatively less impact on the performance of 2D-to-3D reconstruction than geometric sub-pipe. The reason is that, although semantic sub-pipe can explicitly disentangle and extract the semantic attribute from 2D images, there always exist certain semantic attributes that cannot be explicitly captured or disentangled. Therefore, an implicit representation is still necessary for encoding such implicit semantic attributes in images. Second, by comparing geometric-MLP and semantic-MLP to the only-MLPs, we can find that both geometric sub-pipe and semantic sub-pipe are more effective than simple MLPs, which proves the effectiveness of the network designation of the two sub-pipes.

Table 3. The effect of each sub-pipe to 3DAttriFlow in terms of L1-CD×10^2.

Steps.	avg.	plane	car	chair	table
w/o semantic sub-pipe	3.16	2.58	2.80	3.89	3.35
w/o geometric sub-pipe	3.41	2.66	3.07	4.23	3.68
semantic-MLP	3.12	2.53	2.85	3.81	3.30
geometric-MLP	3.08	**2.47**	2.73	3.80	3.30
only-MLP	3.21	2.67	2.82	3.91	3.45
Full	**3.03**	2.49	**2.69**	**3.73**	**3.23**

Visualization of semantic attributes controlled by semantic code z. The semantic code z is expected to encode explicit semantic attribute into the activation of a single dimension, which aims to provide definite guidance for the reconstruction of 3D shape. In order to visually analyze the encoded semantic attributes captured by z, we go through


Shape of chair back at stage 2 dim 9.
Length of legs at stage 2 dim 5.
Bending of legs at stage 3 dim 6.
Length of legs at stage 3 dim 2.
Bending of legs at stage 3 dim 6.
Bending of tabletop at stage 2 dim 5.
Length of wings at stage 2 dim 5.
Bending of wings at stage 3 dim 9.
Shape of empennage at stage 3 dim 3.

(a) Chair (b) Table (c) Plane

Figure 6. Visualization of 3D shape manipulation through semantic code z. Each row of the sub-figure shows corresponding shape deformation results caused changing the value of a single dimension of semantic code z, which proves that semantic code is able to control explicit semantic attributes during the 2D-to-3D reconstruction process. Since we cannot control which semantic attribute is encoded by network, we manually go through the dimensions of semantic code z and reveal the learned semantic attributes.

the dimensions of z and observe the shape deformations caused by interpolating single dimensions of z, as shown in Figure 6. Specifically, we illustrate our observations of 3 attributes for each of the 3 categories, which proves that semantic code z successfully captures the explicit semantic attribute, and effectively reveals the reconstruction of the corresponding part of 3D shape. For example, as for the reconstruction of chair (Figure 6(a)), the code z learns two specific semantic attributes of legs, which are the bending (encoded by 6th dimension at stage 2) and the length (encoded by the 5th dimension at stage 3), respectively. From the visualization results of Figure 6(c), we can find that changing the value of activation will result in obvious deformation of corresponding semantic attributes. Moreover, by observing the extracted semantic attributes across three categories, we can find that semantic code z is able to generalize its learned attributes into multiple categories, as the same attributes of bending and length can also be found in the table and plane categories.

Table 4. The effect of code dimension in terms of L1-CD×10^2 (baseline marked by "*").

Dims.	avg.	plane	car	chair	table
4	3.14	2.60	2.81	3.84	3.30
8	3.11	2.51	2.73	3.83	3.35
18*	**3.03**	**2.49**	**2.69**	**3.73**	**3.23**
32	3.20	2.58	2.75	3.85	3.34

Analysis of dimension of semantic code z. Since each dimension of semantic code z can potentially encode a certain semantic attribute, in this part, we discuss the capability of semantic code z for encoding semantic attributes in terms of code dimensions. We report the results under code dimension of 4, 8 and 32 following the power of 2, and compare it with our default setting 18 in Table 4. From the results we can find that with 18-dimensional semantic code 3DAttriFlow achieves the best performance, while the other settings cause a relatively small performance drop. The reason is that for small dimensions, the semantic code can only encode the limited semantic attributes, which is insufficient for predicting a detailed 3D shape. On the other hand, large dimensions may have the problems for learning orthogonal bases to represent the semantic attributes.

5. Conclusions and Limitations

In this paper, we propose 3DAttriFlow to reconstruct 3D shapes from 2D images. Compared with the previous methods, which merely learn to reconstruct the 3D shape based on the implicit features, 3DAttriFlow takes the advantage of a novel attribute flow pipe to explicitly extract semantic attributes from the implicit feature, which makes the 3D shape prediction more accurate based on the extracted semantic attributes. To overcome the problem of generating discrete point cloud data, the deformation pipe is proposed to combine with the attribute pipe, which provides location priors for the extracted semantic attributes. Comprehensive experiments on ShapeNet dataset for 2D-to-3D reconstruction and MVP dataset for point cloud completion have proved the effectiveness of 3DAttriFlow, and the visualization of shape manipulation also demonstrates the ability of 3DAttriFlow to extract and control the explicit semantic attributes of 3D shapes.

The limitations and possible future work of 3DAttriFlow can be addressed as follows. Although the semantic code z is able to learn the explicit semantic attributes and encode them into certain dimensions, it cannot always learn a meaningful or disentangled semantic attributes for every dimension. In experiments, we observe that some dimensions may have effect to several attributes, while others may have little effect on the output shape. In our opinion, this can be dedicated to the information loss/compression during the extraction process of global image feature, which may cause the semantic attribute missing or deeply entangled with each other. Therefore, the feature channels connecting multiple layers of encoder to the attribute flow pipe is still necessary, in order to fully utilize the ability of semantic attribute extraction of 3DAttriFlow.

References

[1] Angel X Chang, Thomas Funkhouser, Leonidas J 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:1512.03012*, 2015. 5

[2] 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 *Proceedings of the European Conference on Computer Vision*, pages 628–644, 2016. 3, 5

[3] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3D object reconstruction from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 605–613, 2017. 3, 5, 6

[4] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3D surface generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 216–224, 2018. 2, 3, 5, 6

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1

[6] Zhenliang He, Meina Kan, and Shiguang Shan. EigenGAN: Layer-wise eigen-learning for GANs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 2, 4

[7] Li Jiang, Shaoshuai Shi, Xiaojuan Qi, and Jiaya Jia. GAL: Geometric adversarial loss for single-view 3D-object reconstruction. In *Proceedings of the European Conference on Computer Vision*, pages 802–816, 2018. 3

[8] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 3, 4

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012. 1

[10] Ruihui Li, Xianzhi Li, Ka-Hei Hui, and Chi-Wing Fu. SP-GAN: Sphere-guided 3D shape generation and manipulation. *ACM Transactions on Graphics*, 40(4):1–12, 2021. 3

[11] Xueting Li, Sifei Liu, Kihwan Kim, Shalini De Mello, Varun Jampani, Ming-Hsuan Yang, and Jan Kautz. Self-supervised single-view 3D reconstruction via semantic consistency. In *Proceedings of the European Conference on Computer Vision*, pages 677–693, 2020. 3

[12] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. PointCNN: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems*, volume 31, pages 820–830, 2018. 3

[13] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3D object reconstruction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 3

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

[15] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy Networks: Learning 3D reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 1, 2, 3, 5

[16] KL Navaneet, Priyanka Mandikal, Mayank Agarwal, and R Venkatesh Babu. CapNet: Continuous approximation projection for 3D point cloud reconstruction using 2D supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8819–8826, 2019. 3

[17] Junyi Pan, Xiaoguang Han, Weikai Chen, Jiapeng Tang, and Kui Jia. Deep mesh reconstruction from single RGB images via topology modification networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9964–9973, 2019. 3

[18] Liang Pan, Xinyi Chen, Zhongang Cai, Junzhe Zhang, Haiyu Zhao, Shuai Yi, and Ziwei Liu. Variational relational point completion network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8524–8533, 2021. 5, 6

[19] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 3

[20] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Conference on Neural Information Processing Systems*, 2017. 3, 5

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 1

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

[23] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015. 1

[24] Bokun Wang, Yang Yang, Xing Xu, Alan Hanjalic, and Heng Tao Shen. Adversarial cross-modal retrieval. In *Proceedings of the 25th ACM International Conference on Multimedia*, pages 154–162, 2017. 1

[25] Jinglu Wang, Bo Sun, and Yan Lu. MVPNet: Multi-view point regression networks for 3D object reconstruction from

a single image. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8949–8956, 2019. 3

[26] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2Mesh: Generating 3D mesh models from single RGB images. In *Proceedings of the European Conference on Computer Vision*, pages 52–67, 2018. 1, 2, 3, 5

[27] Peng-Shuai Wang, Chun-Yu Sun, Yang Liu, and Xin Tong. Adaptive O-CNN: A patch-based deep representation of 3D shapes. *ACM Transactions on Graphics*, 37(6):1–11, 2018. 1, 2, 3

[28] Xiaogang Wang, Marcelo H Ang Jr, and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 790–799, 2020. 6

[29] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. ESRGAN: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision Workshops*, 2018. 2

[30] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions On Graphics*, 38(5):1–12, 2019. 3, 4

[31] Chao Wen, Yinda Zhang, Zhuwen Li, and Yanwei Fu. Pixel2Mesh++: Multi-view 3D mesh generation via deformation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1042–1051, 2019. 1, 2, 3

[32] Xin Wen, Peng Xiang, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. PMP-Net: Point cloud completion by learning multi-step point moving paths. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7443–7452, 2021. 2, 5, 6

[33] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. SnowflakeNet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5499–5509, 2021. 5, 6

[34] Lei Xiao, Salah Nouri, Matt Chapman, Alexander Fix, Douglas Lanman, and Anton Kaplanyan. Neural supersampling for real-time rendering. *ACM Transactions on Graphics*, 39(4):142–1, 2020. 2

[35] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2Vox: Context-aware 3D reconstruction from single and multi-view images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2690–2698, 2019. 1, 2, 3

[36] Haozhe Xie, Hongxun Yao, Shengping Zhang, Shangchen Zhou, and Wenxiu Sun. Pix2Vox++: Multi-scale context-aware 3D object reconstruction from single and multiple images. *International Journal of Computer Vision*, 128(12):2919–2935, 2020. 1, 2, 3

[37] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057. PMLR, 2015. 1

[38] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. PCN: Point completion network. In *2018 International Conference on 3D Vision*, pages 728–737. IEEE, 2018. 6

[39] Xuancheng Zhang, Rui Ma, Changqing Zou, Minghao Zhang, Xibin Zhao, and Yue Gao. View-aware geometry-structure joint learning for single-view 3D shape reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 3

[40] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021. 5

[41] Liangli Zhen, Peng Hu, Xu Wang, and Dezhong Peng. Deep supervised cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10394–10403, 2019. 1

[42] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2223–2232, 2017. 2