3D Shape Generation and Completion through Point-Voxel Diffusion

Linqi Zhou
Stanford University

Yilun Du
MIT

Jiajun Wu
Stanford University

Figure 1: The proposed Point-Voxel Diffusion (PVD) is a new framework for generative modeling of 3D shapes. Left: tables, cars, and planes generated by our PVD. It learns to sample from a Gaussian prior and to progressively remove noise to obtain sharp shapes. Right: two possible shapes completed from a real RGB-D image, each visualized in input and canonical views.

Abstract

We propose a novel approach for probabilistic generative modeling of 3D shapes. Unlike most existing models that learn to deterministically translate a latent vector to a shape, our model, Point-Voxel Diffusion (PVD), is a unified, probabilistic formulation for unconditional shape generation and conditional, multi-modal shape completion. PVD marries denoising diffusion models with the hybrid, point-voxel representation of 3D shapes. It can be viewed as a series of denoising steps, reversing the diffusion process from observed point cloud data to Gaussian noise, and is trained by optimizing a variational lower bound to the (conditional) likelihood function. Experiments demonstrate that PVD is capable of synthesizing high-fidelity shapes, completing partial point clouds, and generating multiple completion results from single-view depth scans of real objects.

1. Introduction

Generative modeling of 3D shapes has extensive applications across vision, graphics, and robotics. To perform well in these downstream applications, a good 3D generative models should be *faithful* and *probabilistic*. A faithful model generates shapes that are realistic to humans and, in cases where conditional inputs such as depth maps are available, respects such partial observations. A probabilistic model captures the under-determined, multi-modal nature of the generation and completion problem: it may sample and produce diverse shapes from scratch or from partial observations. As shown in Figure 1, when only the back of a chair is visible, good generative models should be able to produce multiple possible completed chairs, including those with arms and those without.

Existing shape generation models can be roughly divided into two categories. The first operates on 3D voxels [43, 14, 45, 2], a natural extension of 2D pixels. While being straightforward to use, voxels demand prohibitively large memory when scaled to high dimensions, and are thus unlikely to produce results with high fidelity. The second class of models studies point cloud generation [1, 11, 48, 46, 16, 17] and has produced promising results. While being more faithful, these approaches typically view point cloud generation as a point generation process conditioned on shape encoding, which is obtained by deterministic encoders. When performing shape completion,

Project page at https://alexzhou907.github.io/pvd these approaches are therefore unable to capture the multi-modal nature of the completion problem.

Recently, a new class of generative models, named probabilistic diffusion models, have achieved impressive performance on 2D image generation [35, 13, 36]. These approaches learn a probabilistic model over a denoising process. Diffusion is supervised to gradually denoise a Gaussian noise to a target output, such as an image. Methods along this line, such as DDPM [13], are inherently probabilistic and produce highly realistic 2D images.

Extending diffusion models to 3D is, however, technically highly nontrivial: a direct application of diffusion models on either voxel and point representation results in poor generation quality. This is because, first, pure voxels are binary and therefore not suitable for the probabilistic nature of diffusion models; second, point clouds demand permutation-invariance, which imposes infeasible constraints on the model. Experiments in Section 4.1 also verifies that a straightforward extension does not lead to reasonable results.

We propose Point-Voxel Diffusion (PVD), a probabilistic and flexible shape generation model that tackles the above challenges by marrying denoising diffusion models with the hybrid, point-voxel representation of 3D shapes [24]. A point-voxel representation builds structured locality into point cloud processing; integrated with denoising diffusion models, PVD suggests a novel, probabilistic way to generate high-quality shapes by denoising a Gaussian noise and to produce multiple completion results from a partial observation, as shown in Figure 1.

A unique strength of PVD is that it is a unified, probabilistic formulation for unconditional shape generation and conditional, multi-modal shape completion. While multi-modal shape completion is a highly desirable feature in applications such as digital design or robotics, past works on shape generation primarily use deterministic shape encoders and decoders to output a single possible completion in voxels or a point cloud. In contrast, PVD can perform both unconditional shape generation and conditional shape completion in an integrated framework, requiring only minimal modifications to the training objective. It is thus capable of sampling multiple completion results depending on diffusion initialization.

Experiments demonstrate that PVD is capable of synthesizing high-fidelity shapes, outperforming multiple state-of-the-art methods. PVD also delivers high-quality results on multi-modal shape completion from partial observations such as a partial point cloud or a depth map. In particular, we show that PVD does well on multi-modal completion on multiple synthetic and real datasets, including ShapeNet [4], PartNet [27], and single-view depth scans of real objects in the Redwood dataset [5].

2. Related Works

Point cloud generative models. Many prior works have explored point cloud generation in terms of auto-encoding [1, 11, 47], single-view reconstruction [12, 10, 18, 17], and adversarial generation [34, 48, 40]. Many of them rely on directly optimizing heuristic loss functions such as Chamfer Distance (CD) and Earth Mover's Distance (EMD), which are also used to evaluate generative quality.

Some recent works take a different approach, viewing the 3D point clouds in light of probabilistic distributions. For example, Sun *et al*. [37] view the point clouds from a probabilistic perspective and introduce autoregressive generation, but doing so requires ordering of the point clouds. GAN-based models and flow-based models [21, 1, 46, 16, 17] also adopt a probabilistic view but separate shape-level distribution from point-level distribution. Among these models, PointFlow applies normalizing flow [29] to 3D point clouds, and Discrete PointFlow follows up using discrete normalizing flow with affine coupling layers [8]. Shape Gradient Fields [3], unlike flow-based works, directly learn a gradient field that samples point clouds using Langevin dynamics. Our model is different from these models in that we do not distinguish point and shape distributions, and that we directly generate entire shapes starting from random noise.

Point-voxel representation. 3D shapes were conventionally rasterized into voxel grids and processed using 3D convolution [6, 42]. Due to the correspondence between voxels and 2D pixels, many works have explored voxel-based classification and segmentation using volumetric convolution [26, 31, 19, 39, 42, 7]. Voxel-based generative models have similarly proven successful [43, 14, 45]. However, voxel grids are memory-intensive and they grow cubically with increase in dimension, so they cannot be scaled to a high resolution.

Point clouds, on the other hand, are detailed samples from smooth surfaces and do not suffer from the grid effect of usually low-resolution voxels and do not require as much memory for processing. Researchers have explored point cloud classification and segmentation [30, 30, 41] and most assume point cloud processing networks are permutation-invariant. Permutation-invariance is a strong condition to be imposed on the architecture and we empirically find that direct extension of 2D methods to either permutation-invariant point clouds or voxels do not work well. We therefore explore a separate point-voxel representation [22, 33], and our work is most related to point-voxel CNN [24], which proposes to voxelize the point clouds for 3D convolution. We use it as the backbone of our generative model due to its exploitation of the strong spacial correlation inherent in point cloud data.

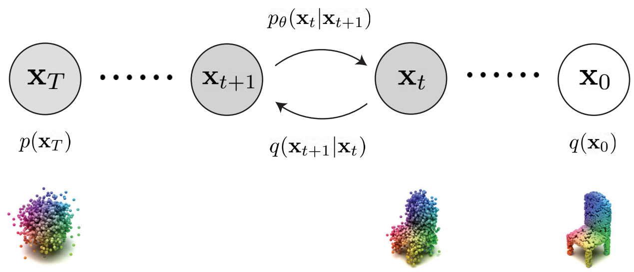


Figure 2: Visualization of the diffusion and generative process. To generate, Gaussian noise is sampled from $p(\mathbf{x}_T)$ and noise is progressively removed by $p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})$. Symmetrically, the diffusion process gradually adds noise by $q(\mathbf{x}_{t+1}|\mathbf{x}_t)$. We utilize a closed-form expression for each $q(\mathbf{x}_{t+1}|\mathbf{x}_t)$, allowing $p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})$ to be learned by simply matching the posterior $q(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0)$ of the corresponding forward transition probability.

Energy-based models and denoising diffusion models. Energy-based models (EBMs) and denoising diffusion models are two classes of generative models that formulate generation as an iterative refinement procedure. Energy based models [20, 9, 28] learn an energy landscape over input data, where local minima correspond to high-fidelity samples, which are obtained by Langevin dynamics [9, 28]. In contrast, denoising diffusion models [35, 13, 36] learn a probabilistic model over a denoising process on inputs. Diffusion is supervised to gradually denoise a Gaussian noise to a target output. This form of supervision can be seen as supervision of the gradient of a log probability distribution [36] as in score matching EBM [15, 38]. Our work builds on these related existing approaches and we explore the 3D domain, which is challenging and fundamentally different from 2D images. A concurrent work on point cloud diffusion model [25] views point cloud generation as a *conditional* generation problem and uses an additional encoder for shape latents. Ours, however, adopts an *unconditional* approach, ridding the need for additional shape encoders, and uses a different hybrid, point-voxel representation for processing shapes. In addition to generating high-quality 3D shapes, we also show that our model can be modified with no architectural change to perform on conditional generation tasks such as shape completion. We also demonstrate its effectiveness on real-world scans.

3. Point-Voxel Diffusion

In this section we introduce Point-Voxel Diffusion (PVD), a denoising diffusion probabilistic model for 3D point clouds. We start by describing our formulation, followed by the training objective for shape generation, and end with the modified objective we proposed for shape completion from partial observation. For all our discussions below, we assume each of our data points are a set of N points with xyz-coordinates and is denoted as $\mathbf{x} \in \mathbb{R}^{N\times 3}$. Our model is parameterized as a single point-voxel CNN [24].

3.1. Formulation

The denoising diffusion probabilistic model is a generative model where generation is modeled as a denoising process. Starting from Gaussian noise, denoising is performed until a sharp shape is formed. In particular, the denoising process produces a series of shape variables with decreasing levels of noise, denoted as $\mathbf{x}_T, \mathbf{x}_{T-1}, ..., \mathbf{x}_0$, where $\mathbf{x}_T$ is sampled from a Gaussian prior and $\mathbf{x}_0$ is the final output.

To learn our generative model, we define a ground truth diffusion distribution $q(\mathbf{x}_{0:T})$ (defined by gradually adding Gaussian noise to the ground truth shape), and learn a diffusion model $p_\theta(\mathbf{x}_{0:T})$, which aims to invert the noise corruption process. We factor both probability distributions into products of Markov transition probabilities:

$$\begin{aligned} q(\mathbf{x}_{0:T}) &= q(\mathbf{x}_0)\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \\ p_\theta(\mathbf{x}_{0:T}) &= p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \end{aligned} \tag{1}$$

where $q(\mathbf{x}_0)$ is the data distribution and $p(\mathbf{x}_T)$ is a standard Gaussian prior. Here, $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ is named the *forward process*, diffusing data into noise; accordingly, $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is named the *reverse process*. $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is named the *generative process*, which we learn, that generates realistic samples by approximates the reverse process. To enable closed-form evaluation, the transition probabilities are also parameterized as Gaussian distributions. We illustrate the processes in Figure 2. Given a pre-determined increasing sequence of Gaussian noise values $\beta_1, ..., \beta_T$,[1] each transition probability can be defined as

$$\begin{aligned} q(\mathbf{x}_t|\mathbf{x}_{t-1}) &:= \mathcal{N}(\sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \\ p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) &:= \mathcal{N}(\mu_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbf{I}). \end{aligned} \tag{2}$$

, where $\mu_\theta(\mathbf{x}_t, t)$ represents the predicted shape from our generative model at timestep $t-1$. Empirically, we found that setting $\sigma_t^2 = \beta_t$ works well. Intuitively, the forward process can be seen as gradually injecting more random noise to the data, with the generative process learning to progressively remove noise to obtain realistic samples by mimicking the reverse process.

Training objective. To learn the marginal likelihood $p_\theta(\mathbf{x})$, we maximize a variational lower bound of log data likelihood that involves all of $\mathbf{x}_0, ..., \mathbf{x}_T$:[1]

$$\mathrm{E}_{q(\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_0)] \geq \mathrm{E}_{q(\mathbf{x}_{0:T})}\Big[\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\Big]. \tag{3}$$

[1] We leave derivation and implementation details to Appendix.

In the above objective, the forward process $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ is fixed and $p(\mathbf{x}_T)$ is defined as a Gaussian prior, so they do not affect the learning of θ. Therefore, the final objective can be reduced to maximum likelihood given the complete data likelihood with joint posterior $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$:

$$\max_{\theta} \mathrm{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \mathbf{x}_{1:T} \sim q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\sum_{t=1}^{T} \log p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \right]. \tag{4}$$

Joint posterior $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$ can be factorized into $\prod_{t=1}^{T} q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$. Each factored ground-truth posterior is denoted as $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ and is analytically tractable. It can be shown that it is also parameterized by Gaussian distributions:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}\left(\frac{\sqrt{\tilde{\alpha}_{t-1}}\beta_t}{1-\tilde{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\tilde{\alpha}_{t-1})}{1-\tilde{\alpha}_t}\mathbf{x}_t, \frac{(1-\tilde{\alpha}_{t-1})}{1-\tilde{\alpha}_t}\beta_t\mathbf{I} \right). \tag{5}$$

where $\alpha_t = 1 - \beta_t$ and $\tilde{\alpha}_t = \prod_{s=1}^{t} \alpha_s$.[1] This property allows each timestep to learn independently, *i.e.*, each $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ only needs to match $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$.

Since both $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ and $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ are Gaussian, we can reparameterize the model to output noise and the final loss can be reduced to an $\mathcal{L}_2$ loss between the model output $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ and noise $\boldsymbol{\epsilon}$:[1]

$$\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right\|^2, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}), \tag{6}$$

Intuitively, the model seeks to predict the noise vector necessary to decorrupt the 3D shape.

Point clouds can then be generated by progressively sampling from $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ as $t = T, ..., 1$ using the following equation:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\tilde{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sqrt{\beta_t}\mathbf{z}, \tag{7}$$

where $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$, corresponding to the gradual denoising of a shape from noise.[1]

3.2. Shape Completion

Our objective can be simply modified to learn a conditional generative model given partial shapes, which we introduce in this section.

Denote a point cloud sample as $\mathbf{x}_0 = (\mathbf{z}_0, \tilde{\mathbf{x}}_0)$, where $\mathbf{z}_0 \in \mathbb{R}^{M \times 3}$ is the fixed partial shape, and any intermediate shapes as free points $\mathbf{x}_t = (\mathbf{z}_0, \tilde{\mathbf{x}}_t)$. We can then define a conditional forward process, where the partial shape is fixed at $\mathbf{z}_0$ for all time. Our conditional forward and generative processes, as well as each transition probability, can then be parametrized as

$$\begin{aligned} q(\tilde{\mathbf{x}}_t|\tilde{\mathbf{x}}_{t-1}, \mathbf{z}_0) &:= \mathcal{N}(\sqrt{1-\beta_t}\tilde{\mathbf{x}}_{t-1}, \beta_t\mathbf{I}), \\ p_\theta(\tilde{\mathbf{x}}_{t-1}|\tilde{\mathbf{x}}_t, \mathbf{z}_0) &:= \mathcal{N}(\mu_\theta(\mathbf{x}_t, \mathbf{z}_0, t), \sigma_t^2\mathbf{I}). \end{aligned} \tag{8}$$

Note that the above equations now give the forward/generative transition probabilities for the free points $\tilde{\mathbf{x}}_t$, while $\mathbf{z}_0$ stays unchanged for all timesteps. Intuitively, this process is the same as unconditional generation, while we hold the partial shape $\mathbf{z}_0$ fixed and diffuse only the missing parts.

The modified training objective also maximizes the likelihood conditioned on partial shapes $\mathbf{z}_0$:

$$\mathrm{E}_{(\tilde{\mathbf{x}}_0, \mathbf{z}_0) \sim q(x_0), \mathbf{x}_{1:T} \sim q(\mathbf{x}_{1:T}|\tilde{\mathbf{x}}_0, \mathbf{z}_0)} \left[\sum_{t=1}^{T} \log p_\theta(\tilde{\mathbf{x}}_{t-1}|\tilde{\mathbf{x}}_t, \mathbf{z}_0) \right], \tag{9}$$

where each posterior $q(\tilde{\mathbf{x}}_{t-1}|\tilde{\mathbf{x}}_t, \tilde{\mathbf{x}}_0, \mathbf{z}_0)$ is known and its derivation is similar to the unconditional generative model. Using the same reasoning as before, we can arrive at a similar $\mathcal{L}_2$ loss:

$$\mathcal{L}_t = \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\tilde{\mathbf{x}}_t, \mathbf{z}_0, t) \right\|^2, \tag{10}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$. Additionally, since the partial shape is always fixed during both forward and generative processes, we can mask away the subset of model output that affects $\mathbf{z}_0$ and minimize $\mathcal{L}_2$ distance between $\tilde{\boldsymbol{\epsilon}}(\tilde{\mathbf{x}}_t, \mathbf{z}_0, t)$ and random noise, which only affects $\tilde{\mathbf{x}}_t$. In practice, we input $\mathbf{z}_0$ and $\mathbf{x}_t$ into the model and obtain $\mathbf{x}_{t-1}$, where only the subset $\tilde{\mathbf{x}}_{t-1}$ is used for $\mathcal{L}_2$ loss. In shape completion, $\tilde{\mathbf{x}}_{t-1}$ is concatenated with $\mathbf{z}_0$ to be the input into the model again. This allows the exact same training architecture to do both generation and shape completion by simply changing the training objective.

4. Experiments

We demonstrate here that our model outperforms previous point generative models in Section 4.1, is capable of completing partial shapes sampled from single views in Section 4.2, and can generate diverse shapes given partial shape constraints in Section 4.3. Architecture and hyperparameter details are provided in Appendix.

4.1. Shape Generation

Data. We choose ShapeNet [4] Airplane, Chair, and Car to be our main datasets for generation, following most previous works [46, 17, 3, 16]. We use the provided datasets in [46], which contain 15,000 sampled points for each shape. We sample 2,048 points for training and testing, respectively, and process our data following procedures provided in PointFlow [46].

Evaluation metrics. Previous works such as [46, 17, 3, 16] have used Chamfer Distance (CD) and Earth Mover's Distance (EMD) as their distance metrics in calculating Jensen-Shannon Divergence (JSD), Coverage (COV), Minimum Matching Distance (MMD), and 1-Nearest Neighbor

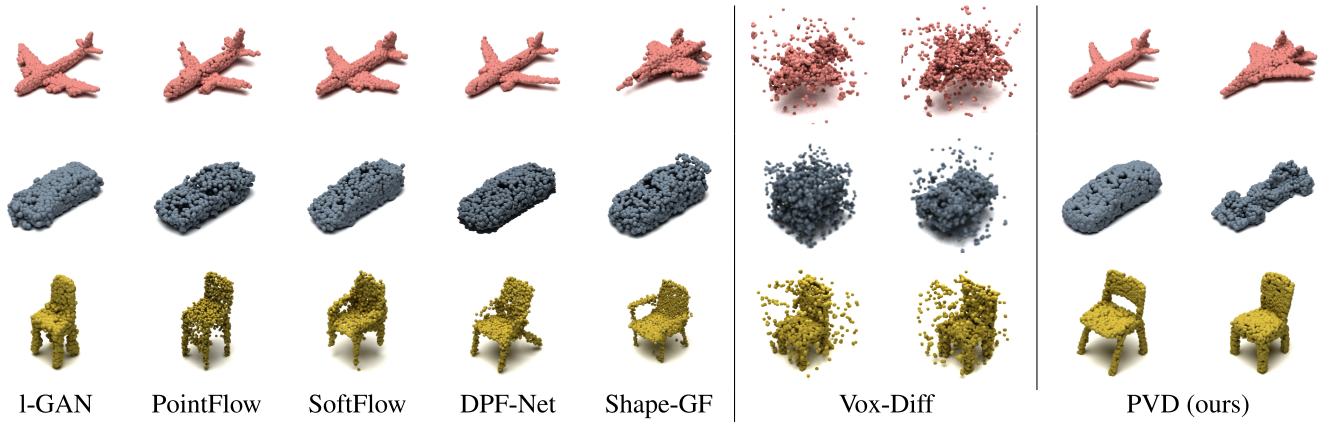


Figure 3: Results on unconditional shape generation with 2,048 points. The l-GAN results are from the EMD variant.

	Airplane		Chair		Car	
	CD	EMD	CD	EMD	CD	EMD
r-GAN [1]	98.40	96.79	83.69	99.70	94.46	99.01
l-GAN (CD) [1]	87.30	93.95	68.58	83.84	66.49	88.78
l-GAN (EMD) [1]	89.49	76.91	71.90	64.65	71.16	66.19
PointFLow [46]	75.68	70.74	62.84	60.57	58.10	56.25
SoftFlow [16]	76.05	65.80	59.21	60.05	64.77	60.09
DPF-Net [17]	75.18	65.55	62.00	58.53	62.35	54.48
Shape-GF [3]	80.00	76.17	68.96	65.48	63.20	56.53
PVD (ours)	**73.82**	**64.81**	**56.26**	**53.32**	**54.55**	**53.83**

Table 1: Generation results on Airplane, Chair, Car compared with baselines using 1-NN as the metric. Both CD and EMD as the distance measure are calculated. Lower scores indicate better quality and diversity.

(1-NN), which are four main metrics to measure generative quality. However, as discussed by [46], JSD, COV, and MMD each has limitations and does not necessarily indicate better quality. Some generation results achieve even better scores than ground-truth datasets. 1-NN is robust and correlates with generation quality, as supported by [46], which also proposes 1-NN as the better metric. Therefore, we use 1-NN directly for evaluating generation quality and we provide comparison of remaining metrics in Appendix. As we also discover that EMD score can vary widely depending on its implementation, We evaluate all baselines using our implementation of the metrics.

Baselines and results. We quantitatively compare our results with r-GAN [1], l-GAN [1], PointFlow [46], DPF-Net [17], SoftFlow [16], and Shape-GF [3] on generating 2048 points. In evaluating the baselines, we follow the same data processing and evaluation procedure as PointFlow, and follow the provided baseline implementations to evaluate their models. Our comparisons are shown in Table 1. Our model noticeably achieves the best generation quality.

We also investigated pure voxel and point representations for shape generation. We noticed that simply extending diffusion models to pure point representation using conventional permutation-invariant architectures such as PointNet++ [32] fails to generate any visible shapes. Extending diffusion models to pure voxel representation generates noisy results due to the binary nature of voxels which is different from our Gaussian assumption. We visually compare with baselines including a voxel diffusion model (Vox-Diff) in Figure 3. For Vox-Diff, 2048 points are sampled from voxel surfaces. We provide additional quantitative comparison in Appendix.

4.2. Shape Completion

In various graphics applications users usually do not have access to all viewpoints of an object. A shape often needs to be completed knowing a partial shape from a single depth map. Therefore, the ability to complete partial shapes become practically useful. In this section, we use the same model architecture (see Appendix) from Section 4.1 and test our shape completion models.

Data. For shape completion, we use the benchmark provided by GenRe [49], which contains renderings of each shape in ShapeNet from 20 random views. We sample 200 points as our partial point clouds obtained from the provided depth images, and we evaluate shape completion on all 20 partial shapes per ground-truth sample.

Metrics. For shape completion, as the ground-truth data are involved, Chamfer Distance and Earth Mover's Distance suffice to evaluate the reconstruction results.

Baselines. Since our approach is probabilistic, we selected major distribution-fitting models such as PointFlow [46], DPF-Net [17], and SoftFlow [16] for comparison. We directly evaluate pre-trained models, if provided, otherwise we re-train them using baselines' provided implementation on our benchmark. We also compared with Shape-GF as its encoder can similarly receive an arbitrary

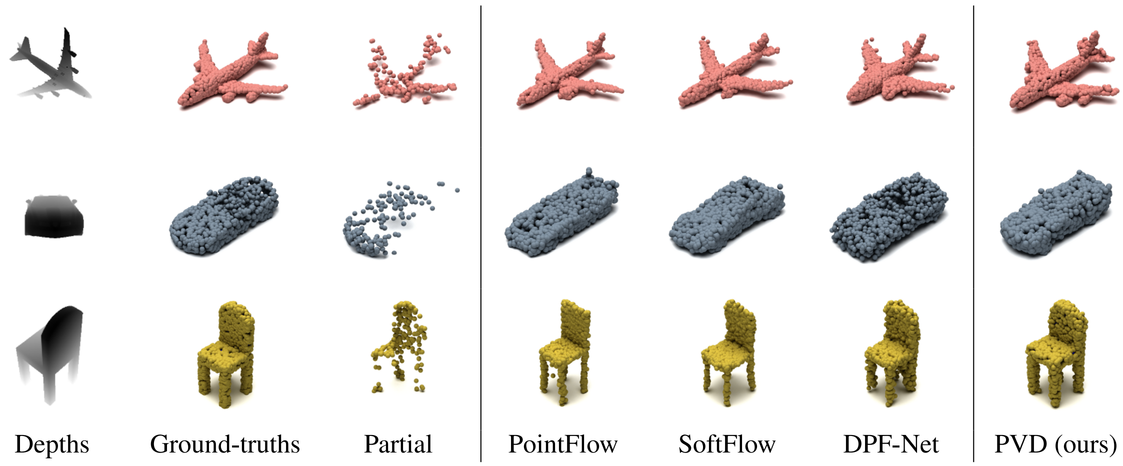


Figure 4: Our shape completion visualization (right) compared to baseline models (left). From left to right: depth images, ground-truth shapes, partial shapes sampled from depth images, completion from baselines, and our results.

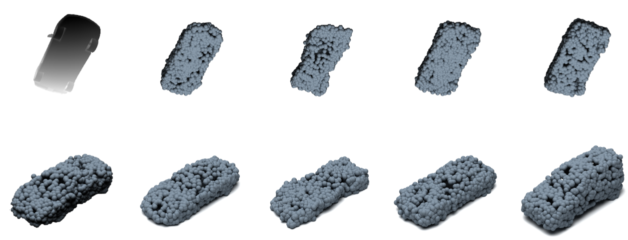
Category	Model	CD	EMD
Airplane	SoftFlow [16]	0.4042	1.198
	PointFlow [46]	**0.4030**	1.180
	DPF-Net [17]	0.5279	1.105
	PVD (ours)	0.4415	**1.030**
Chair	SoftFlow [16]	2.786	3.295
	PointFlow [46]	**2.707**	3.649
	DPF-Net [17]	2.763	3.320
	PVD (ours)	3.211	**2.939**
Car	SoftFlow [16]	1.850	2.789
	PointFlow [46]	1.803	2.851
	DPF-Net [17]	**1.396**	2.318
	PVD (ours)	1.774	**2.146**

Table 2: Quantitative comparison against baselines. CD is multiplied by 10^3 and EMD is multiplied by 10^2.

number of points. However, it is experimentally found that the model is sensitive to the input partial shapes and completion is not realistic after Langevin sampling. Therefore, we leave them out of the comparison.

Results. Quantitative results are presented in Table 2 and a visual comparison is shown in Figure 4. From Table 2, we observe that our model achieves best on EMD scores while worse on CD compared to some baselines. First, we note EMD is a better metric for measuring completion quality because by solving the linear assignment problem it forces model outputs to have the same density as the ground-truths [23] and it is known that CD is blind to visual inferiority [1]. Our better EMD score is more indicative of higher visual quality.

Next, we investigate the reason our CD is inferior to our baselines. We discover that a typical case when our CD is higher is as shown in Figure 5, where from the input



GT	PointFlow	SoftFlow	DPF-Net	PVD (ours)
CD	8.633	1.501	1.197	1.921
EMD	1.124	1.856	1.109	0.919

Figure 5: Typical case when CD is higher than baseline models. Column 1 shows input depth image and ground-truth point clouds. The next columns show completion from the input viewpoint (top) and from the canonical viewpoint (bottom). CD is multiplied by 10^3 and EMD is multiplied by 10^2 scores.

view the ground-truth shape is largely unknown. The baseline models tend to output a mean shape when encountered with such an unconventional angle. Naturally, mean shapes are more frequently closer to the ground-truths than other shapes, as exemplified by the figure. However, with each noise initialization, our model seeks a possible completion that matches well with the partial shape and may be further away from the ground-truth than the mean shape. In the case shown, our completion is a van instead of a sedan but is equally realistic.

Our model also enables controlled completion given multiple partial shapes, and we leave details to Appendix.

4.3. Multi-Modal Completion

Our baselines for shape completion adopt an encoder-decoder structure that takes in a partial shape and outputs a single completion. While some offer impressive results, their completion ability is deterministic, much different

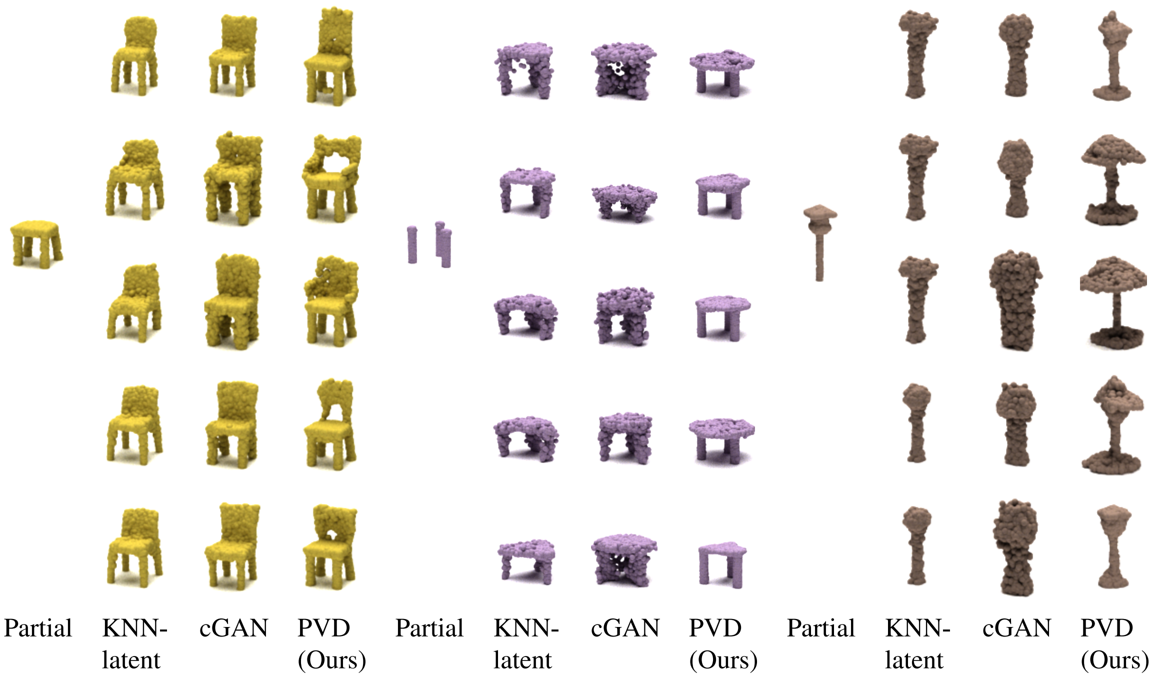


Figure 6: Multi-modal completion visualization on PartNet. Each column presents five completion modes from a model.

	TMD $\times 10^2$				MMD $\times 10^3$			
	Chair	Table	Lamp	Avg.	Chair	Table	Lamp	Avg.
KNN-latent [44]	0.96	1.37	1.95	1.43	1.42	1.42	1.88	1.57
cGAN [44]	1.75	1.99	1.94	1.89	1.61	1.56	2.13	1.77
PVD (ours)	1.91	1.70	5.92	**3.18**	1.27	1.03	1.98	**1.43**

Table 3: Quantitative comparison for multi-modal completion on PartNet. TMD (higher the better) measures completion diversity, and MMD (lower the better) measures completion quality. Chamfer Distance (CD) is used as the distance measure.

from humans who can often imagine different completion possibilities given single views. Our PVD, however, adopts a probabilistic approach to shape completion, where each noise initialization can result in a different completion.

Data. We follow experiment setups from cGAN [44] and train our model on Chair, Table, and Lamp from PartNet [27]. 1024 points are given and 2048 points are generated as completion. In addition, we show diversity of our completion on ShapeNet. Different from PartNet, ShapeNet's partial shapes are sampled from custom-rendered depth images.

Metrics. We follow cGAN [44] and use Total Mutual Difference (TMD) to measure diversity and Minimal Matching Distance (MMD) to measure quality with Chamfer Distance (CD) as distance measure. Since our model is different from cGAN and only completes the 1024 free points, TMD is calculated only on the free points for our model and on a subsampled set of 1024 points for our baselines. MMD is calculated using the completed 2048 points and re-sampled 2048 ground-truth points.

Results on PartNet. Our model is compared with cGAN [44] and KNN-latent [44]. Results are shown in Table 3 and visual comparisons are shown in Figure 6. Our model outperforms both baselines in terms of average diversity and quality.

Results on ShapeNet. The shape completion model trained in Section 4.2 is directly used to demonstrate completion diversity on ShapeNet, as shown in Figure 7. We choose a bottom view of a chair and show that, in the top row, all of our completion results match well with the constrained viewpoint, and in the bottom row, our completion results are noticeably diverse from the canonical viewpoint.

Results on real scans. We further investigate how our model pre-trained on ShapeNet can perform on scans of real objects. We use the Redwood 3DScans dataset [5] and test our model on partial shapes of chairs and tables, sampled from its depth images. Since the GenRe benchmark [49] does not provide table data, the training data for the table category are generated by randomly sampling 20 views from ShapeNet meshes, following GenRe's procedure. Within each example, we present the real RGB-D scans and ground-truths from the input views.

Figure 8 shows results on two views of two different chairs and tables. For chair scans, the left example shows the front view, and the completion results only vary slightly

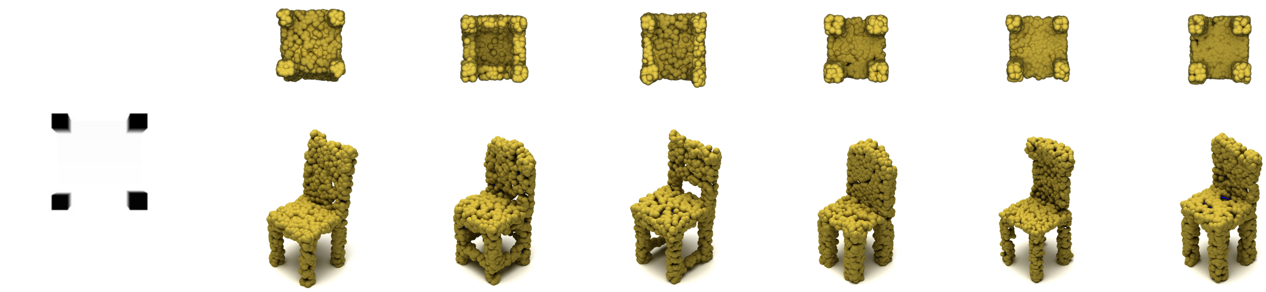


Figure 7: Multi-modal completion results on ShapeNet. Left: ground-truth bottom view depth image of a chair. Right: six different possible shape completion results. Top: completion from the depth image viewpoint. Bottom: completion from the canonical viewpoint.

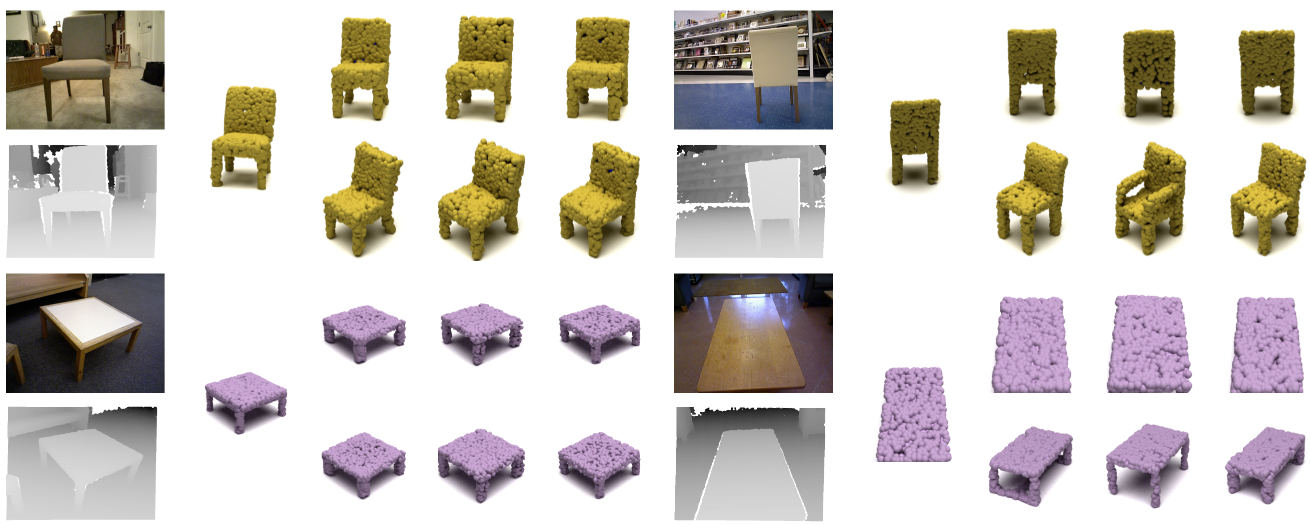


Figure 8: Application of our model on scans from the Redwood 3DScans dataset. PVD takes partial point clouds induced from the depth maps, not the RGB image as input. Left: from a more complete view, the model outputs stable, similar completions. Right: from an uncertain viewpoint, the model outputs multiple completions with a larger variation.

across different runs. The right chair example shows a back view, and the uncertainty allows more varied completion. Similarly, the left table scan shows a large part of the table, so the completion varies less than the right example, which only shows the top of the table.

5. Conclusion

We have introduced PVD, a unified framework for both shape generation and shape completion. Our model, trained on a simple $\mathcal{L}_2$ loss, is based on diffusion probabilistic models and learns to reverse a diffusion process by progressively removing noise from noise-initialized samples. A minor modification on the objective also results in a shape completion model without the need for any architectural change. Experimentally, we show the failure with straight-forward extension of diffusion models to either pure voxel or point representations. With the point-voxel representation, our model demonstrates superior generative power and impressive shape completion quality. Unlike most baseline models which use deterministic encoder-decoder structures, PVD can output multiple possible completion results given a partial shape. Additionally, it can complete real 3D scans, thus offering practical usage in various downstream applications.

Acknowledgements: Yilun Du is funded by an NSF graduate fellowship. This work is in part supported by ARMY MURI grant W911NF-15-1-0479, Samsung Global Research Outreach (GRO) program, Amazon Research Award (ARA), IBM, Stanford HAI, and Autodesk.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3D point clouds. In *ICML*, 2018. 1, 2, 5, 6

[2] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Generative and discriminative voxel modeling with convolutional neural networks. *arXiv:1608.04236*, 2016. 1

[3] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. In *ECCV*, 2020. 2, 4, 5

[4] 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:1512.03012*, 2015. 2, 4

[5] Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. A large dataset of object scans. *arXiv:1602.02481*, 2016. 2, 7

[6] 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*, 2016. 2

[7] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In *MICCAI*, 2016. 2

[8] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *ICLR*, 2017. 2

[9] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In *NeurIPS*, 2019. 3

[10] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3D object reconstruction from a single image. In *CVPR*, 2017. 2

[11] Matheus Gadelha, Rui Wang, and Subhransu Maji. Multiresolution tree networks for 3D point cloud processing. In *ECCV*, 2018. 1, 2

[12] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché approach to learning 3D surface generation. In *CVPR*, 2018. 2

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 2, 3

[14] Wenlong Huang, Brian Lai, Weijian Xu, and Zhuowen Tu. 3D volumetric modeling with introspective neural networks. In *AAAI*, 2019. 1, 2

[15] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *JMLR*, 6:695–709, 2005. 3

[16] Hyeongju Kim, Hyeonseung Lee, Woo Hyun Kang, Joun Yeop Lee, and Nam Soo Kim. SoftFlow: Probabilistic framework for normalizing flow on manifolds. In *NeurIPS*, 2020. 1, 2, 4, 5, 6

[17] Roman Klokov, Edmond Boyer, and Jakob Verbeek. Discrete point flow networks for efficient point cloud generation. In *ECCV*, 2020. 1, 2, 4, 5, 6

[18] Andrey Kurenkov, Jingwei Ji, Animesh Garg, Viraj Mehta, JunYoung Gwak, Christopher Choy, and Silvio Savarese. DeformNet: Free-form deformation network for 3D shape reconstruction from a single image. In *WACV*, 2018. 2

[19] Truc Le and Ye Duan. PointGrid: A deep network for 3D shape understanding. In *CVPR*, 2018. 2

[20] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting Structured Data*, 2006. 3

[21] Chun-Liang Li, Manzil Zaheer, Yang Zhang, Barnabas Poczos, and Ruslan Salakhutdinov. Point cloud GAN. In *ICLR Workshop*, 2019. 2

[22] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. PointCNN: Convolution on x-transformed points. In *NeurIPS*, 2018. 2

[23] Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and Shi-Min Hu. Morphing and sampling network for dense point cloud completion. In *AAAI*, 2020. 6

[24] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3D deep learning. In *NeurIPS*, 2019. 2, 3

[25] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3D point cloud generation. In *CVPR*, 2021. 3

[26] Daniel Maturana and Sebastian Scherer. VoxNet: A 3D convolutional neural network for real-time object recognition. In *IROS*, 2015. 2

[27] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *CVPR*, 2019. 2, 7

[28] Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. Learning non-convergent non-persistent short-run MCMC toward energy-based model. In *NeurIPS*, 2019. 3

[29] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *JMLR*, 22(57):1–64, 2021. 2

[30] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017. 2

[31] Charles R Qi, Hao Su, Matthias Nießner Angela Dai Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view CNNs for object classification on 3D data supplementary material. In *CVPR*, 2016. 2

[32] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 5

[33] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN: Point-voxel feature set abstraction for 3D object detection. In *CVPR*, 2020. 2

[34] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3D point cloud generative adversarial network based on tree structured graph convolutions. In *ICCV*, 2019. 2

[35] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 2, 3

[36] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 2, 3

[37] Yongbin Sun, Yue Wang, Ziwei Liu, Joshua E Siegel, and Sanjay E Sarma. PointGrow: Autoregressively learned point cloud generation with self-attention. In *WACV*, 2020. 2

[38] Kevin Swersky, Marc'Aurelio Ranzato, David Buchman, Benjamin M Marlin, and Nando de Freitas. On autoencoders and score matching for energy based models. In *ICML*, 2011. 3

[39] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs. In *ICCV*, 2017. 2

[40] Diego Valsesia, Giulia Fracastoro, and Enrico Magli. Learning localized generative models for 3D point clouds via graph convolution. In *ICLR*, 2019. 2

[41] 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 TOG*, 38(5):1–12, 2019. 2

[42] Zongji Wang and Feng Lu. VoxSegNet: Volumetric cnns for semantic part segmentation of 3D shapes. *IEEE TVCG*, 26(9):2919–2930, 2020. 2

[43] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *NeurIPS*, 2016. 1, 2

[44] Rundi Wu, Xuelin Chen, Yixin Zhuang, and Baoquan Chen. Multimodal shape completion via conditional generative adversarial networks. In *ECCV*, 2020. 7

[45] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Zhu Song-Chun, and Ying Nian Wu. Learning descriptor networks for 3D shape synthesis and analysis. In *CVPR*, 2018. 1, 2

[46] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. PointFlow: 3D point cloud generation with continuous normalizing flows. In *ICCV*, 2019. 1, 2, 4, 5, 6

[47] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. FoldingNet: Point cloud auto-encoder via deep grid deformation. In *CVPR*, 2018. 2

[48] Maciej Zamorski, Maciej Zięba, Piotr Klukowski, Rafał Nowak, Karol Kurach, Wojciech Stokowiec, and Tomasz Trzciński. Adversarial autoencoders for compact representations of 3D point clouds. *CVIU*, 193:102921, 2020. 1, 2

[49] Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Joshua B Tenenbaum, William T Freeman, and Jiajun Wu. Learning to reconstruct shapes from unseen classes. In *NeurIPS*, 2018. 5, 7