DIFFUSION-BASED POINT CLOUD GENERATION WITH SMOOTHNESS CONSTRAINTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models have been popular for point cloud generation tasks. Existing works utilize the forward diffusion process as a discrete Markov Chain to convert the original point distribution into a noise distribution (e.g., standard Gaussian distribution) and learn the reverse diffusion process to recover the target point distribution from the noise distribution. However, the diffusion process can produce samples with non-uniform points on the surface without consideration of the point cloud geometric feature. To alleviate the problem, we propose a novel diffusion-based framework for point cloud generation and incorporate the local smoothness constraint into the generation process. Experiments demonstrate that the proposed model is not only capable of generating realistic shapes but also generating more uniform point clouds, outperforming multiple state-of-the-art methods.

1 INTRODUCTION

As a widely-used 3D representation, point clouds have attracted widespread attention in recent years due to their compactness, flexibility, and the nature of closing to raw sensory data. (Yue et al., 2018; Guo et al., 2020) As a result, a number of sophisticated algorithms have been proposed for analyzing point clouds in applications such as robotics, autonomous driving, and AR/VR. (Pomerleau et al., 2015) However, the raw data acquired in real applications require expensive labor effort and time cost, not to mention that the acquired point clouds are often imperfect: they may be sparse and partial due to the distances, occlusions, reflections, and the limits of devices' resolution and angles (Zhao et al., 2019). Recently, diffusion-based generative models are effective in generating 2D images and also pose a promising direction for generating 3D point clouds. However, there are clear differences between point cloud data and image data: points in a point cloud have no order and are arranged irregularly. These characteristics of point clouds make it non-trivial to leverage diffusion-based methods in generation tasks of point clouds. In addition, finding a way to ensure the uniformity on the surface of the point cloud is also a challenging problem.

Several algorithms have been proposed for point cloud generation. For example, probabilistic methods (Yang et al., 2019; Luo & Hu, 2021) treat the generation of point clouds as a probability problem. Based on the structure of VAE, to learn the distribution of distribution, first learn the distribution of a shape, and then learn the conditional distribution of points generated based on this shape. But the Pointflow (Yang et al., 2019) needs to normalize the learned distribution, which is time-consuming. ShapeGF (Cai et al., 2020) must require two stages of training. There are also some methods based on autoregressive model (Sun et al., 2020), but they need to specify an order of points during training. Li et al. (2018) proposes to combine WGAN (Arjovsky et al., 2017) loss with EMD, and other GAN-based methods like (Shu et al., 2019; Valsesia et al., 2018; Achlioptas et al., 2018) are difficult to train and not very stable due to the optimization of the adversarial loss function during the training process.

There are also several diffusion-based generative models (Luo & Hu, 2021; Zhou et al., 2021), which treat points as particles in a thermodynamic system that diffuse from the original distribution to the noise distribution. The generation process can move the noisy point locations towards the original point cloud surface. The training process aims to learn a score model to approximate the gradient of log-likelihood of the point cloud distribution, which is equivalent to learning the moving direction of the particles (points) during diffusion. However, both algorithms are the discrete generative process and thus cannot take into account the shape geometric information when learning point moving di-

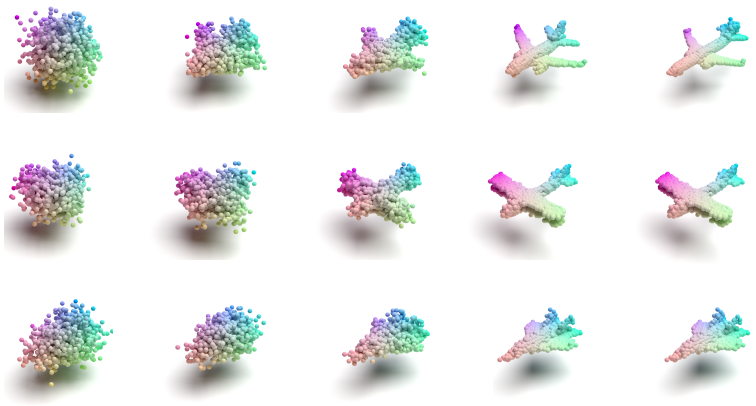

Figure 1: Visualization of the diffusion process from noise to point distribution for airplane category

rections. Compared with diffusion-based models for image data that learn the data distribution in the feature space, these models face a critical challenge of learning the 3D point location distribution. In this paper, we aim to incorporate the local geometrical constraints into diffusion-based point cloud generation process to enforce the surface smoothness. Our approach views the diffusion probabilistic model from Bayesian perspective to take the prior constraint of point cloud into consideration. Experiments demonstrate that our method is not only capable of generating realistic shapes but also generating more uniform point clouds, outperforming multiple state-of-the-art methods.

2 Related work

2.1 Score-based Generative models

Recently, there has been appearing many deep generative models. For example, the variational auto-encoder(VAE) model(Kingma & Welling, 2013) can approximate the likelihood of the data by maximizing the Evidence lower bound(ELBO). The autoregressive(Van den Oord et al., 2016; Van Den Oord et al., 2016) is a time-series model predicting the future values based on the pase values The generative adversarial neural(GAN) model(Goodfellow et al., 2020) consists of two parts, one is the generator to generate samples, and the other is the discriminator to evaluate the authenticity of the samples, and then generate more realistic samples through adversarial learning. The flow based model(Rezende & Mohamed, 2015; Grathwohl et al., 2018; Chen et al., 2018) can obtain exact likelihood calculations. Diffusion model(Song et al., 2020; Ho et al., 2020) is another popular model leveraging the approximate inference similar to VAE to optimize the model's goal, and similarly, score-based models can learn through denosing score matching, or add continuous noise through sde(Song et al., 2020).

2.2 point cloud generation

With the rise of deep networks, there have been many works that use neural networks to generate point clouds or complete point clouds.

Early work is to directly optimize distance, for example, Chamfer distance(CD) or earth mover's distance(EMD) to improve the effect. (Achlioptas et al., 2018) But there are some problems with these methods. CD has been shown to falsely favor over-concentrated point cloud in the mode of the marginal point distributions, and accurate EMD computation can be slow.

Some work also study probability-based methods to generate point clouds. For example, point flow(Yang et al., 2019) proposes to use the VAE architecture to generate point clouds. The paper proposes to use normalizing flow to parameterize the model. ShapeGF(Cai et al., 2020) directly learns the shape distribution of latent space. When sampling, first use gan to generate a shape code z, and then use Langevin Sampling to get the final point cloud. DPM(Luo & Hu, 2021) proposes to

use conditional ddpm to generate point clouds. The point voxel model(Zhou et al., 2021) does not learn the laten distribution, but directly samples the point cloud from the noise, and cites the voxel as additional information.

Recently there are several works caring about the uniformity of the generated samples. For example, Lyu et al. (2021) proposes to firstly use the diffusion model to generate coarse results, then training another diffusion model in the second stage to refine the completion point cloud. Tang et al. (2022) proposes to incorporate various uniform priors during Generative adversarial networks(GAN) training to produce more uniform point cloud.

In this paper, we propose one simple but efficient constraint to help generating more uniform surface.

2.3 SCORE-BASED POINT CLOUD GENERATION

Due to the rise of diffusion models in recent years, it has shown good results on more and more tasks. It is no exception on point clouds. For example, ShapeGF(Cai et al., 2020) uses annealing Langevin for sampling, DPM(Luo & Hu, 2021) proposes that conditional discrete DDPM(Ho et al., 2020) is also effective, and point voxel(Zhou et al., 2021) uses the representation of points and voxel to learn to generate point clouds via a discrete DDPM.

However, discrete diffusion model will generate non-uniform surface and there will be some noise on the surface, based on these observations we propose continuous-time diffusion model and introduce smooth constraint to achieve more uniform surface.

3 BACKGROUND

3.1 DIFFUSION PROCESS

We consider the diffusion process over x indexed with time $t \in [0, 1]$, $\{x_t\}_{t=0}^1$. Here $x_0 \sim p_0(\mathbf{x}) = p_{data}$, $x_1 \sim p_1(\mathbf{x})$, where $p_1(\mathbf{x})$ approximates the Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The diffusion process can be modeled by Itô SDE:

$$dX_t = f_t(X, t)dt + g_t dw, \tag{1}$$

where f_t is a drift coefficient, g is a diffusion coefficient. The forward diffusion process smoothly transform x by adding infinitesimal noise dw at each infinitesimal step dt.

In order to generate samples following the data distribution, we start from the prior distribution and follow the reverse diffusion process, which is defined by:

$$dX_t = [f_t(X_t) - g_t^2 \nabla_{X_t} \log p_t(X_t)]dt + g_t d\bar{w}, \tag{2}$$

The goal of the diffusion model training process is to learn the time-dependent score model $\mathbf{s}_\theta(\mathbf{x}, t)$ to minimize the least square loss between the score function and the model output

$$\min \int_0^1 \mathbb{E}_{p_t(\mathbf{x})}[\|\mathbf{s}_\theta(\mathbf{x}, t) - \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\|^2]dt \tag{3}$$

3.2 TWEEDIDE'S FORMULA

Suppose we have N samples $[x_1, x_2, x_3, ..., x_N]$ sampled from $\mathcal{N}(\mu, \sigma^2)$, where μ is unknown. And suppose μ is sampled from some prior distribution, $\mu \sim g(\cdot)$ then $x|\mu \sim \mathcal{N}(\mu, \sigma^2)$. The goal is to estimate the parameter μ.

Tweedie's formula(Robbins, 1992; Kim & Ye, 2021) gives the posterior mean of μ as:

$$\mathbb{E}[\mu|x] = \mu + \sigma^2 \nabla_x \log f(x) \tag{4}$$

where $f(x) = \int g(\mu) f(x|\mu) d\mu$ is the marginal distribution of x. We will use this formula to estimate the clean point clouds in Section4.2.2.

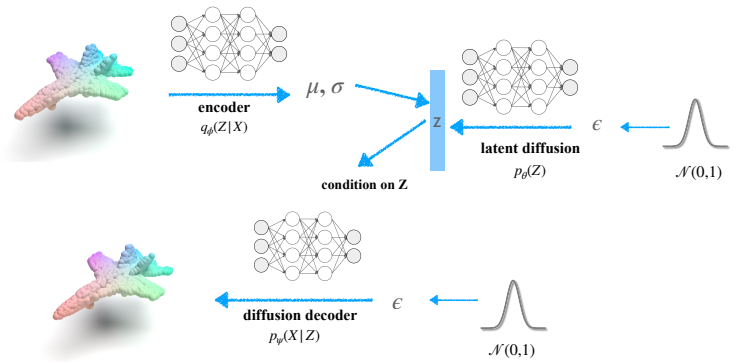

Figure 2: The overall structure of our model

4 APPROACH

A point cloud consists of a set of points in 3D space, which can be denoted by $\mathbf{X}_i = \{\boldsymbol{x}_j^i\}_{i=1}^N \in \mathbb{R}^3$, where N is the number of points, and each $\boldsymbol{x}_i$ is the 3D coordinates. In the following, we use $\boldsymbol{x}$ to represent the point cloud $\mathbf{X}$ for algebraic convenience. The dataset consists of M different point clouds $\mathcal{D} = \{\boldsymbol{x}^{(1)}, ...\boldsymbol{x}^{(M)}\}$. The goal is to develop a model that generate accurate point cloud that is faithful to the ground-truth's shape geometry and variability. The points in one point cloud are discrete positions in 3D space, and can be regarded as the particles' location in a non-equilibrium thermodynamic system. Therefore we consider the point cloud generation as a particle continuous diffusion process, and model it as an Itô diffusion process.

However, the problem poses several unique challenges for diffusion-based point cloud generation. First, the dataset contains diverse shapes even for a single object, but learning the marginal distribution across different shape is not useful. Thus, this requires to model the dataset's shape distribution and individual point cloud distribution separately. Second, diffusion probabilistic model learns the intermediate movement of the points and is not capable to consider the point cloud local geometric information, which may generate non-smooth shape during sampling. Third, to generate a diverse and faithful point cloud, it requires learning the global shape distribution accurately.

To overcome the above challenge, we propose a diffusion probabilistic model with smoothness constraint to capture both the global and the local geometric features of point clouds. The overall structure is shown in Figure2. . The model consists of an encoder, a diffusion decoder and a latent diffusion model. The encoder $q_\phi(Z|X)$ learns a low dimensional latent variable distribution for the given point cloud input to encode its global geometric shape distribution, Given a global shape encoding, the diffusion decoder aims to reconstruct the original shape geometry through a reverse diffusion process. Furthermore, to model the distribution of the latent shape Z accurately, we use a latent diffusion model to transform a standard normal distribution to the aggregated posterior of the encoder.

4.1 PROPOSED OVERALL MODEL

The goal of our model is to learn the point cloud distribution over a diverse shape. As indicated by Li et al. (2018), for point cloud generation, it is not useful to learn the marginal distribution $p(\mathbf{X})$ across the shapes. Thereby we model the distribution into factorized format and incorporate the global geometry into consideration: $p_{\theta,\psi}(\mathbf{X}, \mathbf{z}) = p_\psi(\mathbf{X}|\mathbf{z})p_\theta(\mathbf{z})$, where $\mathbf{z} \in \mathbb{R}^d$ is a latent d-dimensional variable that represents the global geometry feature of point cloud, $p_\theta(\mathbf{z})$ represent the global geometry prior distribution. $p_\psi(\mathbf{X}|\mathbf{z})$ is a conditional generative model given a specific global geometry condition. The global geometry posterior is approximated through variational posterior model $q_\phi(\mathbf{z}|\mathbf{X})$. To optimize the generative model, we maximize the Evidence Lower Bound(ELBO) on the data log-likelihood as Equation5. In the following section, we will introduce the conditional generative model and the prior model in detail.

$$\max_{\phi,\psi,\theta} \mathcal{L}(\phi,\psi,\theta) = \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})}\big[\log p_\psi(\mathbf{X}|\mathbf{Z})\big] - \mathrm{KL}\left(q_\phi(z|x) \parallel p_\theta(z)\right) \tag{5}$$
$$= \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})}\big[\log p_\psi(\mathbf{X}|\mathbf{Z})\big] + \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})}\big[\log p_\theta(\mathbf{Z})\big] + \mathbb{H}\big[q_\phi(\mathbf{Z}|\mathbf{X})\big].$$

4.1.1 DIFFUSION BASED PRIOR MODEL

The prior model aims to capture the distribution of the global geometry of all point cloud shapes. A common choice is to use a fixed prior (e.g., $p(\mathbf{z}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$) in variational auto-encoder, but it can produce holes, then there exist mismatch between the aggregated posterior $\sum_i q_\phi(\mathbf{z}|\mathbf{X}^{(i)})$ and the prior $p(\mathbf{z})$. Then this can lead to some regions in the latent space that cannot be decoded to data-like samples.

In our proposed model, we use a learnable prior model $p_\theta(\mathbf{z})$, the prior is flexible, such that it can catch up with the aggregated posterior, then they can match each other better, thus alleviating the problem of prior holes. Here we propose to use the diffusion process to learn the prior over the shape distribution. Specifically, given a global geometry encoding $\mathbf{z}_0 \sim q_\phi(\mathbf{z}_0|\mathbf{X})$, we diffused z_0 through a SDE defined in Eq. 1 to the standard Normal distribution $p(\mathbf{z}_1) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, and we train a time-dependent score model $s_\theta(\mathbf{z}_t, t)$ to approximate the true score $\nabla_{\mathbf{z}_t} p_\theta(\mathbf{z}_t)$. For the generation process, the reversed SDE based on Eq. 2 is employed to generate samples that following the prior distribution $p_\theta(\mathbf{z}_0)$. For the score model, we use several layers of 1D convolution with residual connection. The details are in the experiment section.

4.1.2 DIFFUSION BASED CONDITIONAL DECODER

In this section, we introduce the framework of conditional diffusion model in the decoder part. Here we consider the point cloud generation as a conditional diffusion process, where the condition is the global geometry encoding $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{X})$. Similarly, we diffuse the points of the point cloud (particles) to the standard Normal distribution and learn a conditional score model $s_\phi(\mathbf{X}_t, \mathbf{z}, t)$ to approximate the reverse moving direction of the points during the forward diffusion process (equivalent to the score $\nabla_{\mathbf{X}_t} p_\psi(\mathbf{X}_t|\mathbf{z})$). For sampling, given a global geometry condition $\mathbf{z}$, we can use the score model $s_\phi(\mathbf{X}_t, \mathbf{z}, t)$ through the reverse SDE to move the noisy point cloud to the original point cloud distribution.

Specifically, the input of this network is the perturbed point cloud with Gaussian noise sampled from $\mathbf{X}_t \sim p_t(\mathbf{X}_t|\mathbf{X})$ and the Fourier positional encoding of the time vector $t_{pe} = pe(t)$. We also concat the latent code z as the condition to the input of the each layer.

4.1.3 TRAINING ALGORITHM

In the training objective of Eq.**??**, we estimate the entropy term $\mathbb{H}\big[q_\phi(\mathbf{z}|\boldsymbol{x})\big]$ via Monte Carlo samples. And our decoder is a continuous diffusion based conditional decoder, then the reconstruction term $\mathbb{E}_{q_\phi(\mathbf{z}|\boldsymbol{x})}\big[\log p_\psi(\boldsymbol{x}|\mathbf{z})\big]$ is approximated by mean square loss of the score function

$$\mathbb{E}_{q_\phi(\mathbf{z}|\boldsymbol{x})}\big[\log p_\psi(\boldsymbol{x}|\mathbf{z})\big] = \mathbb{E}_{t,p_t(\boldsymbol{x})}\big[\frac{g(t)^2}{2}\|\mathbf{s}_\psi(\boldsymbol{x}|\mathbf{z},t) - \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}|\mathbf{z})\|_2^2\big]\big] \tag{6}$$

, and we optimize the cross-entropy term through score matching following :

$$\mathbb{E}_{q_\phi(\mathbf{z}|\boldsymbol{x})}\big[\log p_\theta(\mathbf{z})\big] = \mathbb{E}_{t,q_\phi(\mathbf{z}^0,\mathbf{z}^t|\boldsymbol{x})}\Big[\frac{g(t)^2}{2}\|\mathbf{s}_\theta(\mathbf{z}^t,t) - \nabla_{\mathbf{z}} \log p_t(\mathbf{z})\|_2^2\Big] \tag{7}$$

where $t \sim U[0,1]$.

4.2 SMOOTHNESS CONSTRAINT

In the subsection4.1.2, we introduced the diffusion-based decoder model to generate point cloud condition on some global geometry encoding. The diffusion process is similar to those models for image data. However, compared with diffusion-based models for image data that learn the data distribution in the feature space, there is a critical challenge of learning the 3D point location distribution, which should obey the geometrical constraint on the 3D object. In this section, we aim to incorporate the local geometrical constraints into diffusion-based point cloud generation process to enforce the surface smoothness.

Algorithm 1 Training Procedure

Input: point cloud training data $\boldsymbol{x}$
for epoch = 0 to n **do**
  Sample $x_0 \sim p_{data}(x), t, t' \sim$ Uniform$(0, 1)$
  Encoder: $z_0 = q_\phi(z|x_0), z_{t'} \sim p_t(z'_t|z_0)$
  Latent diffusion loss: $L_z^t \leftarrow \frac{g(t)^2}{2}\|\mathbf{s}_\theta(\mathbf{z}^t, t) - \nabla_{\mathbf{z}} \log p_t(\mathbf{z})\|_2^2$ // Equation (6)
  Sample $x_t \sim p_t(x_t|x_0)$
  Reconstruction loss function $L_x^t \leftarrow \frac{g(t)^2}{2}\|\mathbf{s}_\psi(\boldsymbol{x}|\mathbf{z}, t) - \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}|\mathbf{z})\|_2^2$ // Equation (7)
  Apply stochastic gradient on the neural network parameters with $\nabla_\theta(L_z^t + L_x^t + \mathbb{H}\big[q_\phi(\mathbf{Z}|\boldsymbol{x})\big])$
end for

4.2.1 Constructing Graph On point cloud

We construct a discrete graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ on the point cloud, whose vertex set $\mathcal{V}$ is the points of the point cloud $x = \{x_i\}_{i=1}^n$. The edge set $\mathcal{E}$ is constructed according to the thresholded Gaussian function

$$e_{ij} = \exp\left(-\frac{d_{ij}^2}{2\sigma^2}\right) \text{ if } d_{ij} < r, 0 \text{ otherwise.} \tag{8}$$

where $d_{ij} = ||\mathbf{x}_i - \mathbf{x}_j||^2$ is the Euclidean distance between two points x_i and x_j, σ is the hyperparameter that controls the magnitude of edge weight. Then we can construct the adjacency matrix$\mathbf{A}$ with each entry $\mathbf{A}_{ij} = e_{ij}$, the diagonal degree matrix $\mathbf{D}$, and the Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$.

4.2.2 Graph Laplacian Smoothness Constraint

In order to incorporate the 3D point cloud geometry constraint into the generation process, we consider the gradient of the posterior distribution instead of the likelihood. Given a sample $\boldsymbol{x}$, suppose the constraint $\mathbf{H}$ on the sample takes the form $p(\mathbf{H}|\boldsymbol{x})$ then we have the log posterior as:

$$\log p(\boldsymbol{x}|\mathbf{H}) = \log p(\mathbf{H}|\boldsymbol{x}) + \log p(\boldsymbol{x}) - logp(\mathbf{H}) \tag{9}$$

then we can get the score by taking the gradient on log posterior:

$$\nabla_x \log p(\boldsymbol{x}|\mathbf{H}) = \nabla_x \log p(\boldsymbol{x}) + \nabla \log p(\mathbf{H}|\boldsymbol{x}) \tag{10}$$

For point cloud generation, we aim to enforce the information that x is smooth with respect to the underlying graph, so that it can generate smoother surface, we add the constraint over the one-step estimation of the clean point clout. $\boldsymbol{x}_t$ can be sampled from $p(\boldsymbol{x}_t|\boldsymbol{x}_0)$, then after learning the score model $s_{\theta*}(\boldsymbol{x}_t, t)) = \nabla \log p(\boldsymbol{x}_t|\boldsymbol{x})$, the clean point cloud $\hat{\mathbf{x}}$ is estimated through the Tweedie's formula from any noisy one $\boldsymbol{x}_t$.

$$\hat{\boldsymbol{x}} = (\boldsymbol{x}_t + b_t^2 s_\theta(\boldsymbol{x}_t, t))/a_t \tag{11}$$

Where b_t, a_t are the specific parameters for different diffusion process.

We consider the constraint given on the one-step estimated clean point cloud $\boldsymbol{x}$ is proportional to the graph Laplacian regularizer(GLR) on the coordinates:

$$\log p(\mathbf{H}|\boldsymbol{x}) \propto -\hat{\boldsymbol{x}}^T \mathbf{L} \hat{\boldsymbol{x}} \tag{12}$$

where x is the graph signal, and L is the graph Laplacian matrix. Accordingly, the reverse diffusion under the additional smoothness constraint can be represented by

$$\mathbf{x}_{t_{i-1}} = \mathbf{x}_{t_i} + [f_t(X_t) - g_t^2 \mathbf{s}(\theta)]\delta t - \alpha \nabla_x \{x^T L x\} + g(\mathbf{x}_i)\mathbf{z}, \mathbf{z} \sim \mathcal{N}(0, 1) \tag{13}$$

where α is the weight controlling the relative weight of the smootheness constraint.

5 Experiment

In this section, we describe the experimental results on the point cloud generation task. The goal of the experiment is to quantitatively compare our proposed framework with several SOTA point cloud generative models on the generation quality. Section 5.3.1 describes the metrics we used to evaluate the performance. Section 5.4 compares the model with six state-of-the-art point cloud generation frameworks. Section 5.5 analyzes the model performance via the uniform loss.

Algorithm 2 Sampling Procedure

Input: score model $\mathbf{s}_\theta$
Sample $\boldsymbol{\epsilon}, \mathbf{x}_{init} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})$
latent sampling: $\epsilon \xrightarrow{\text{diffusion}} \mathbf{z}$
$\mathbf{x}_{t_n} = \mathbf{x}_{init}$
for i = n to 1 **do**
  sampling time $t_i = \Delta(i)$
  $\mathbf{s} = \mathbf{s}_\theta(\mathbf{x}, \mathbf{z}, \mathbf{t})$
  $\mathbf{x}_{t_{i-1}} = \mathbf{x}_{t_i} + [f_t(X_t) - g_t^2 \mathbf{s}(\theta)]\delta t - \alpha \nabla_x \{x^T L x\} + g(\mathbf{x}_i)\mathbf{z}, \mathbf{z} \sim \mathcal{N}(0, 1)$
end for
return $\mathbf{x}_{t_0}$

5.1 Experiment setup

5.2 Baseline methods

We compare our proposed model against several SOTA models for point cloud generation. Those model can be divided into two categories: diffusion-based models and non-diffusion-based models. The non-diffusion-based models include r-GAN(Achlioptas et al., 2018), GCN-GAN(Valsesia et al., 2018), TreeGAN(Shu et al., 2019), PointFlow(Yang et al., 2019), The diffusion-based models include ShapeGF(Cai et al., 2020), DPM(Luo & Hu, 2021).

5.2.1 dataset

We conduct all the experiments on the ShapeNet dataset(Chang et al., 2015). ShapeNet dataset contains 51,127 shapes from 55 categories. We followed the Yang et al. (2019); Cai et al. (2020) to split the dataset. Each point cloud contains 2048 points uniformly sampled on the surfaces. Here we compare the performance on the airplane and chair category.

5.3 Implementation detail

Hyperparameters: We set the latent code dimension is 256. We sample point clouds with N = 2048 points. For the decoder score modle, We utilized Adam with learning rate $r = 2 \times 10^{-3}$ for encoder, and learning rate $r = 2 \times 10^{-4}$ for decoder score model, 1×10^{-4} for the prior score model. We used $\beta_1 = 0.9, \beta_2 = 0.999$ for the adam optimization. The learning rate was kept constant for the first 1000 epoches, and decreased linearly from 1000 to 2000 epoches. The batch size is 32. For the score model inference, we set time step to 1000 steps, we utilized VPSDE model, and $\beta_{\text{min}} = 0.01$, $\beta_{\text{max}} = 8$.

Model architecture: For the encoder model, our model takes point clouds as input, we use the PointNet (Qi et al., 2017) architecture to map it into a 512-dimension latent feature code $\mathbf{z}$. For the decoder and the prior score model, we chose the OccNet (Mescheder et al., 2019). We stacked 6 ResNet-blocks with 256 dimensions for every hidden layer.

5.3.1 Evaluation metrics

For point cloud generation task, suppose P_r is the set of reference point clouds, P_g is the set of generated point clouds, we adopt the following metrics:

1. Minimum Matching Distance(MMD)

$$MMD(P_r, P_g) = \frac{1}{|P_r|} \sum_{P \in P_r} \min_{Q \in P_g} D(P, Q) \tag{14}$$

   MMD calculates the average of the closest distances to each reference point cloud so that it measures the quality of the generated samples.

2. Coverage score(COV)

$$COV(P_r, P_g) = \frac{|\{\arg\min_{P \in P_r} D(P, Q) | Q \in P_g\}|}{|P_r|} \tag{15}$$

Table 1: Comparison of shape generation on ShapeNet. MMD is multiplied by 10^2. Numbers in bold indicate the best two methods

Category	Model	MMD($10^2, \downarrow$)	COV(%,$\uparrow$)	1-NNA(%,$\downarrow$)
Airplane	r-GAN	13.287	19.75	100.00
	GCN	15.535	5.93	99.12
	Tree	16.662	6.91	100.00
	PF	**7.576**	41.98	**82.22**
	ShapeGF	**7.364**	41.98	**83.46**
	DPM	8.22	32.96	90.86
	Ours	7.86	**42.96**	85.68
	Ours+constraint	7.64	**46.42**	**83.46**
Chair	r-GAN	32.688	8.31	99.92
	GCN	25.781	6.34	96.48
	Tree	36.545	8.76	99.92
	PF	19.190	44.41	**72.28**
	ShapeGF	**18.79**	**46.37**	**59.82**
	DPM	21.37	30.00	90.31
	Ours	20.21	43.7	76.23
	Ours+constraint	**19.13**	**45.31**	73.3

COV measures the variety of the generated samples so that it can detect mode collapse.

3. 1-NN classifier accuracy(1-NNA) to evaluate the performance. 1-NNA score classifies the generated samples and the reference point could. If the quality of the generated samples is better, the 1-NNA score should be close to 50%.

For the distance metrics D, we choose Earth Mover's distance (EMD).

$$EMD(X, Y) = \min_{\phi: X \to Y} \sum_{x \in X} ||x - \phi(x)||_2$$

5.4 Generation analysis

For point cloud generation, we compare our model with the baseline models on the airplane and chair category. We consider two settings of our proposed model. "Ours" represents the model without smoothness constraint and the other represents the model with smoothness constraint. To prevent the metrics focus more on the scale rather than the quality, we follow ShapeGF, DPM(Luo & Hu, 2021), to normalize the generated point cloud and reference point cloud into a bounding box of $[-1, 1]^3$. The evaluation results are summarized in Table 1. For the airplane category, we can observe that our proposed model outperform the baseline models in COV metrics. For 1-NN classifier accuracy, our model performs comparable with the ShapeGF. Our proposed model with constraint achieve comparative performance with the baseline methods. It can be noted that our model with smoothness constraint increase the performance compared with the model without the constraint. Thus, it validates the effectiveness of the proposed constraint. We can observe the similar trends on the chair dataset. We also visualize some generated sample in Fig. 3

5.5 Uniform analysis

Furthermore, to better demonstrate the performance of the constraint, we incorporate the uniform loss(Li et al., 2019) to evaluate the model. This loss calculates the local and non-local uniformity via chi-squared model. We evaluated the mean value on the normalized point clouds with the percentage of $\{0.002, 0.006, 0.008, 0.012, 0.015\}$. The results are shown in Table 2

6 Conclusions

In this paper, we propose a new diffusion-based model for point cloud generation by incorporating not only the global geometry distribution but the local smoothness constraint into the generation

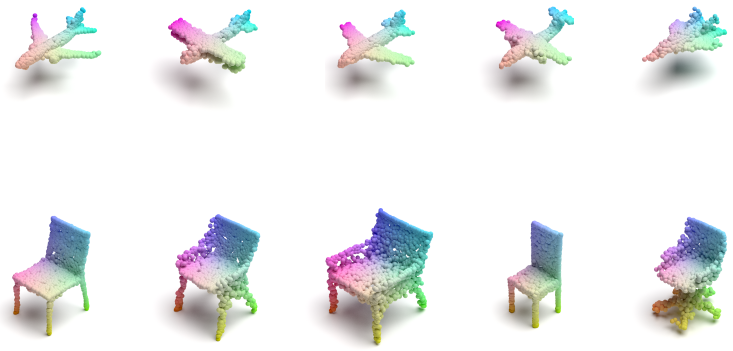

Figure 3: Visualization of some generated samples by our model

Table 2: Comparison of the uniform performance on ShapeNet

Category	Model	Uniform(↓)
Airplane	ShapeGF	0.005
	DPM	0.008
	Ours	0.007
	Ours+constriant	0.006
Chair	ShapeGF	0.0128
	DPM	0.0171
	Ours	0.0152
	Ours+constriant	0.013

process. Experiments show that the proposed framework can not only generate realistic samples but also smoother point clouds. The limitation of our approach is that we only consider the smoothness constraint on the point cloud. In the future, we may add more geometric or topology constraint into the framework to improve the quality.

REFERENCES

Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pp. 40–49. PMLR, 2018.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. In *European Conference on Computer Vision*, pp. 364–381. Springer, 2020.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(12):4338–4364, 2020.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Kwanyoung Kim and Jong Chul Ye. Noise2score: tweedie's approach to self-supervised image denoising without clean images. *Advances in Neural Information Processing Systems*, 34:864–874, 2021.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Chun-Liang Li, Manzil Zaheer, Yang Zhang, Barnabas Poczos, and Ruslan Salakhutdinov. Point cloud gan. *arXiv preprint arXiv:1810.05795*, 2018.

Ruihui Li, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-gan: a point cloud upsampling adversarial network. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7203–7212, 2019.

Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2837–2845, 2021.

Zhaoyang Lyu, Zhifeng Kong, Xudong Xu, Liang Pan, and Dahua Lin. A conditional point diffusion-refinement paradigm for 3d point cloud completion. *arXiv preprint arXiv:2112.03530*, 2021.

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*, pp. 4460–4470, 2019.

François Pomerleau, Francis Colas, Roland Siegwart, et al. A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends® in Robotics*, 4(1):1–104, 2015.

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 conference on computer vision and pattern recognition*, pp. 652–660, 2017.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pp. 1530–1538. PMLR, 2015.

Herbert E Robbins. An empirical bayes approach to statistics. In *Breakthroughs in statistics*, pp. 388–394. Springer, 1992.

Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3d point cloud generative adversarial network based on tree structured graph convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3859–3868, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Yongbin Sun, Yue Wang, Ziwei Liu, Joshua Siegel, and Sanjay Sarma. Pointgrow: Autoregressively learned point cloud generation with self-attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 61–70, 2020.

Yingzhi Tang, Yue Qian, Qijian Zhang, Yiming Zeng, Junhui Hou, and Xuefei Zhe. Warpinggan: Warping multiple uniform priors for adversarial 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6397–6405, 2022.

Diego Valsesia, Giulia Fracastoro, and Enrico Magli. Learning localized generative models for 3d point clouds via graph convolution. In *International conference on learning representations*, 2018.

Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016.

Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pp. 1747–1756. PMLR, 2016.

Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4541–4550, 2019.

Xiangyu Yue, Bichen Wu, Sanjit A Seshia, Kurt Keutzer, and Alberto L Sangiovanni-Vincentelli. A lidar point cloud generator: from a virtual world to autonomous driving. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pp. 458–464, 2018.

Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. Pointweb: Enhancing local neighborhood features for point cloud processing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5565–5573, 2019.

Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5826–5835, 2021.