ECE 765: Probabilistic Graphical Models

Diffusion Probabilistic Models for 3D Point Cloud Generation

Tanisha Khurana & Vikram Pande
Team: 501

Spring 2024

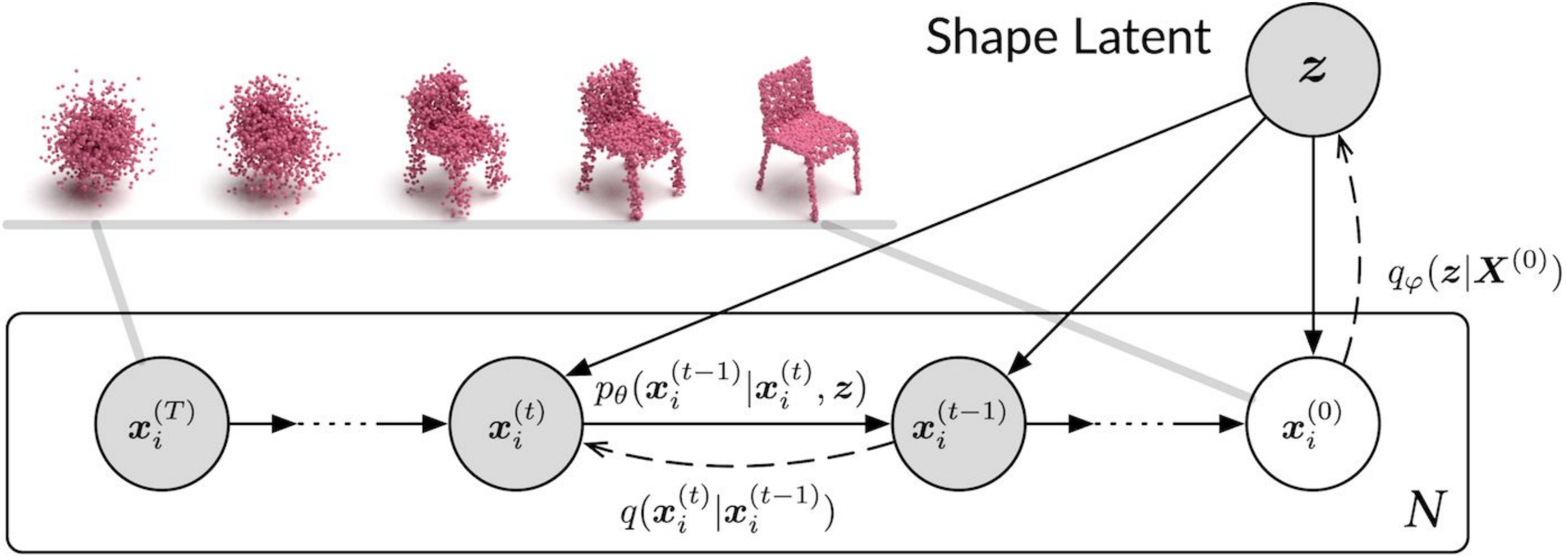
Motivation

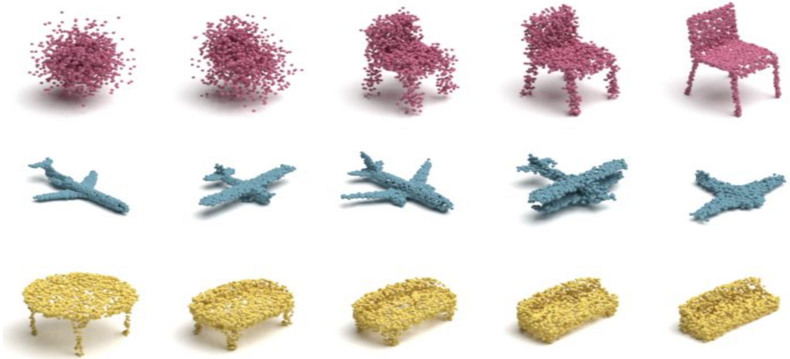
- Generative models play pivotal roles in unsupervised representation learning.
- Important in applications including shape completion, upsampling and synthesis.
- 3D Computer Vision is rapidly becoming a mainstream with generative models such as diffusion, NeRF, Gaussian Splatting, etc.

Why diffusion models?

In space of 3D point cloud generation:

- Traditional generative models like VAEs, GANs, Normalizing Flows are remarkable.
- However, the irregular sampling patterns inherent in 3D point clouds, pose a unique set of challenges for direct application.
- The irregularity in point distribution complicates the direct extension of these generative models to point clouds.
- Solution: Probabilistic Generative Models: **Diffusion**


Shape Latent
$\boldsymbol{z}$
$q_\varphi(\boldsymbol{z}|\boldsymbol{X}^{(0)})$
$\boldsymbol{x}_i^{(T)}$
$\boldsymbol{x}_i^{(t)}$
$p_\theta(\boldsymbol{x}_i^{(t-1)}|\boldsymbol{x}_i^{(t)}, \boldsymbol{z})$
$\boldsymbol{x}_i^{(t-1)}$
$\boldsymbol{x}_i^{(0)}$
$q(\boldsymbol{x}_i^{(t)}|\boldsymbol{x}_i^{(t-1)})$
N



Dataset

ShapeNet

- ShapeNet is a richly-annotated, large-scale dataset of 3D shapes.
- It covers 55 common object categories with about 51,300 unique 3D models.
- x,y,z coordinates, randomly split into training, testing and validation sets by the ratio 80%, 15%, 5%
- The original paper used two categories:
    - Airplane
    - Chair
- We used Airplane, Chair, Lamp, Sofa and Table.

Methodology: What are Diffusion Models?

Diffusion: Inspired by Thermodynamics. Points gradually diffuse into a chaotic set of points.

- Conceptualizing a 3D point cloud as a dynamic thermodynamic system, each point within the cloud is analogous to an independent sample from a distribution denoted as $q(x_i^{(0)}|z)$
- z is the shape latent that determines the distribution of points. Over time, these points undergo a diffusion process, transitioning from an ordered state to a state of entropy.

Methodology: Diffusion

Forward Diffusion

- Converting original meaningful point distribution into a noise distribution.
- The forward diffusion process is modeled as a Markov chain.

$$q(x_i^{(1:T)}|x_i^{(0)}) = \prod_{t=1}^{T} q(x_i^{(t)}|x_i^{(t-1)})$$

- The diffusion kernel adds the noise to points from the previous time step.

$$q(x^{(t)}|x^{(t-1)}) = \mathcal{N}\left(x^{(t)} \,|\, (1-\beta_t)x^{(t-1)}, \beta_t I\right)$$

Reverse Diffusion

- The generation process is viewed as the reverse of the diffusion process, where points are sampled from a noise *p(xi(T))* approximating *q(xi(T))*.
- These points traverse a reverse Markov chain to form the desired shape.

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(0:T)}|\boldsymbol{z}) = p(\boldsymbol{x}^{(T)}) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(t-1)}|\boldsymbol{x}^{(t)}, \boldsymbol{z}),$$

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(t-1)}|\boldsymbol{x}^{(t)}, \boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}^{(t-1)}|\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}^{(t)}, t, \boldsymbol{z}), \beta_t \boldsymbol{I}),$$

Methodology: Training Objective

- The objective is to maximize the log-likelihood of the point cloud X(0).
- However, as directly optimizing the exact log-likelihood is intractable, the approach is to maximize its variational lower bound.

$$
\begin{aligned}
\mathbb{E}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{X}^{(0)})\right] \geq \mathbb{E}_q\Bigg[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{X}^{(0:T)}, \boldsymbol{z})}{q(\boldsymbol{X}^{(1:T)}, \boldsymbol{z}|\boldsymbol{X}^{(0)})}\Bigg] \\
= \mathbb{E}_q\Bigg[\log p(\boldsymbol{X}^{(T)}) \\
+ \sum_{t=1}^{T} \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{X}^{(t-1)}|\boldsymbol{X}^{(t)}, \boldsymbol{z})}{q(\boldsymbol{X}^{(t)}|\boldsymbol{X}^{(t-1)})} \\
- \log \frac{q_{\boldsymbol{\varphi}}(\boldsymbol{z}|\boldsymbol{X}^{(0)})}{p(\boldsymbol{z})}\Bigg].
\end{aligned}
\tag{7}
$$

Methodology: Training Objective

- This variational bound is adapted into the training objective L to be *minimized*

$$
\begin{aligned}
L(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \mathbb{E}_q\Big[\sum_{t=2}^{T} D_{\mathrm{KL}}\big(q(\boldsymbol{X}^{(t-1)}|\boldsymbol{X}^{(t)}, \boldsymbol{X}^{(0)})\|\\
p_{\boldsymbol{\theta}}(\boldsymbol{X}^{(t-1)}|\boldsymbol{X}^{(t)}, \boldsymbol{z})\big) \qquad (8)\\
-\log p_{\boldsymbol{\theta}}(\boldsymbol{X}^{(0)}|\boldsymbol{X}^{(1)}, \boldsymbol{z})\\
+ D_{\mathrm{KL}}\big(q_{\boldsymbol{\varphi}}(\boldsymbol{z}|\boldsymbol{X}^{(0)})\|p(\boldsymbol{z})\big)\Big].
\end{aligned}
$$

Methodology: Training Algorithm

Algorithm 1 Training (Simplified)

repeat
  Sample $\boldsymbol{X}^{(0)} \sim q_{\text{data}}(\boldsymbol{X}^{(0)})$
  Sample $\boldsymbol{z} \sim q_{\boldsymbol{\varphi}}(\boldsymbol{z}|\boldsymbol{X}^{(0)})$
  Sample $t \sim \text{Uniform}(\{1, \ldots, T\})$
  Sample $\boldsymbol{x}_1^{(t)}, \ldots, \boldsymbol{x}_N^{(t)} \sim q(\boldsymbol{x}^{(t)}|\boldsymbol{x}^{(0)})$
  $L_t \leftarrow \sum_{i=1}^{N} D_{\text{KL}} \left(q(\boldsymbol{x}_i^{(t-1)}|\boldsymbol{x}_i^{(t)}, \boldsymbol{x}_i^{(0)}) \middle\| p_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{(t-1)}|\boldsymbol{x}_i^{(t)}, \boldsymbol{z}) \right)$
  $L_{\boldsymbol{z}} \leftarrow D_{\text{KL}}(q_{\boldsymbol{\varphi}}(\boldsymbol{z}|\boldsymbol{X}^{(0)}) \| p(\boldsymbol{z}))$
  Compute $\nabla_{\boldsymbol{\theta}}(L_t + \frac{1}{T} L_{\boldsymbol{z}})$. Then perform gradient descent.
until converged

Methodology: Model Architecture

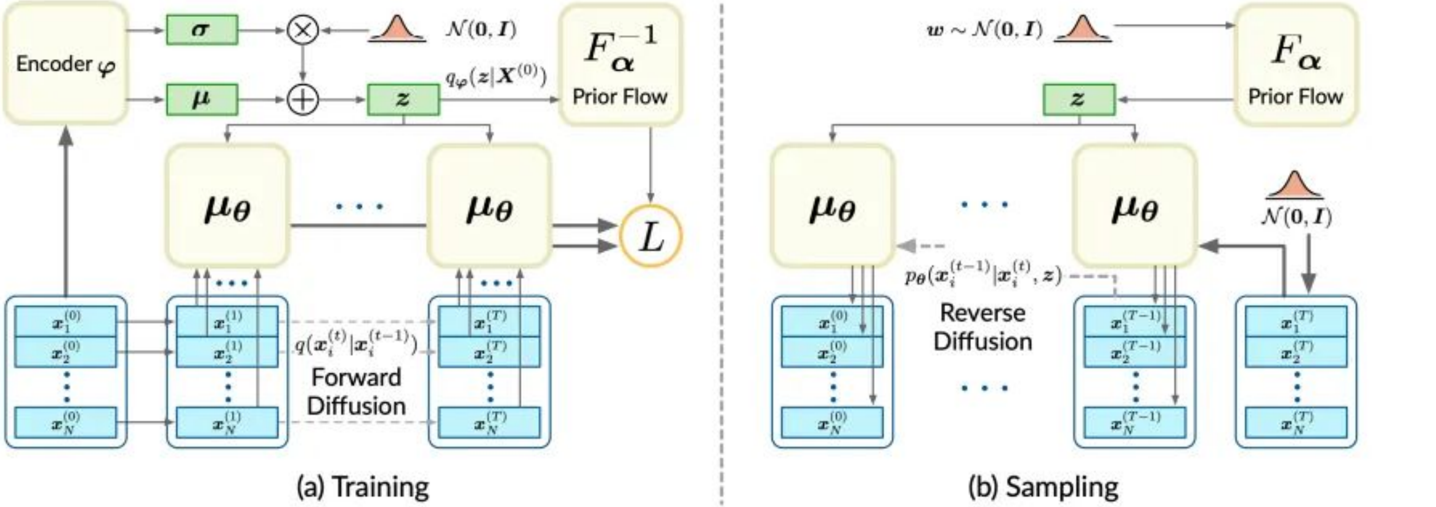


Figure 3. The illustration of the proposed model. (a) illustrates how the objective is computed during the training process. (b) illustrates the generation process.

Methodology: Model Architecture

Point Cloud Generator

- Point cloud generative model is inspired by Normalizing Flows employing affine coupling layers.
- **PointNet** for encoding and reverse Markov chain for sampling

$$L_G(\boldsymbol{\theta}, \boldsymbol{\varphi}, \boldsymbol{\alpha}) = \mathbb{E}_q\Big[\sum_{t=2}^{T}\sum_{i=1}^{N} D_{\mathrm{KL}}\big(q(\boldsymbol{x}_i^{(t-1)}|\boldsymbol{x}_i^{(t)}, \boldsymbol{x}_i^{(0)})\| p_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{(t-1)}|\boldsymbol{x}_i^{(t)}, \boldsymbol{z})\big) - \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{(0)}|\boldsymbol{x}_i^{(1)}, \boldsymbol{z}) + D_{\mathrm{KL}}\Big(q_{\boldsymbol{\varphi}}(\boldsymbol{z}|\boldsymbol{X}^{(0)})\Big\| p_{\boldsymbol{w}}(\boldsymbol{w}) \cdot \Big|\det \frac{\partial F_{\boldsymbol{\alpha}}}{\partial \boldsymbol{w}}\Big|^{-1}\Big)\Big]. \tag{15}$$

Point Cloud AutoEncoder

- Encoder, utilizes PointNet with parameters ϕ, while decoding relies on the reverse diffusion process
- This decoding process is conditioned on the latent code produced by the encoder.

$$L(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \mathbb{E}_q\Big[\sum_{t=2}^{T}\sum_{i=1}^{N} D_{\mathrm{KL}}\big(q(\boldsymbol{x}_i^{(t-1)}|\boldsymbol{x}_i^{(t)}, \boldsymbol{x}_i^{(0)})\| p_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{(t-1)}|\boldsymbol{x}_i^{(t)}, E_{\boldsymbol{\varphi}}(\boldsymbol{X}^{(0)}))\big) - \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{(0)}|\boldsymbol{x}_i^{(1)}, E_{\boldsymbol{\varphi}}(\boldsymbol{X}^{(0)}))\Big]. \tag{16}$$

Evaluation Metric

Generator

- **MMD (Minimum Matching Distance)**: Measures the fidelity of the generated samples.
- **COV (Coverage Score)**: detects mode-collapse.
- **1-NN Classifier Accuracy**
- **JSD (Jason Shannon-Divergence)**: similarity between the point distributions of the generated set and the reference set

AutoEncoder: Reconstruction

- **CD (Chamfer Distance)**: standard metric to measure the shape dissimilarity between point clouds in point cloud completion
- **EMD (Earth Mover Distance)**: Calculate the distance between the measured sample and the nominal sample.

Our Results

	# Iterations	AutoEncoder		Generator			
		CD	EMD	MMD	COV	1-NN A	JSD
Airplane	50000	0.182	–	0.0032	0.4876	0.6359	0.0099
Chair	50000	0.542	–	0.0476	0.4964	0.6159	0.0089
Lamp	20000	0.594	–	0.0122	0.4752	0.6016	0.0080
Sofa	20000	0.643	–	0.0526	0.5065	0.6232	0.0091
Table	30000	0.688	–	0.0574	0.5075	0.5937	0.0093

x,y,z raw point cloud data

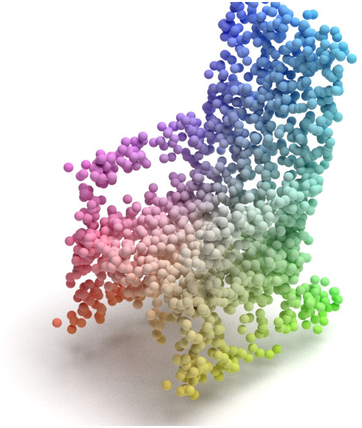
```
array([[[-0.23950163,  0.21508539,  0.39805728],
        [-0.89497674, -0.15010259,  0.276079  ],
        [-0.2747079 , -0.04428494, -0.35782924],
        ...,
        [ 0.9087178 ,  0.00593776, -0.31872767],
        [ 0.14997448,  0.26067868,  0.2277957 ],
        [-0.9432953 , -0.01340136,  0.33928266]],

       [[ 0.0688122 ,  0.04726226,  0.47401586],
        [ 0.77067447, -0.08566985, -0.512486  ],
        [-0.5523629 , -0.17466778,  0.06257565],
        ...,
        [ 0.74809444, -0.19067813, -0.2040249 ],
        [ 0.7164183 , -0.05665219,  0.1697916 ],
        [-0.65651166, -0.06780273,  0.04480147]],

       [[ 0.02701738,  0.0520025 , -0.20445892],
        [-0.6351372 , -0.30502018,  0.25018412],
        [ 0.35378572, -0.31176838, -0.16094889],
        ...,
        [-0.45100102,  0.4199101 ,  0.22620867],
        [-0.53301924,  0.44934765,  0.2534398 ],
        [-0.5871303 ,  0.27686968, -0.24000262]],

       ...,

       [[-0.68636906, -0.00289637, -0.03626729],
        [-0.87321544, -0.05864154,  0.2667144 ],
        [ 0.49311697, -0.01479942, -0.19119963],
        ...,
        [ 0.2581919 , -0.03121711,  0.26054335],
        [ 0.01824323,  0.02131341, -0.49892598],
        [-0.86516035, -0.13010944,  0.24836345]],
```

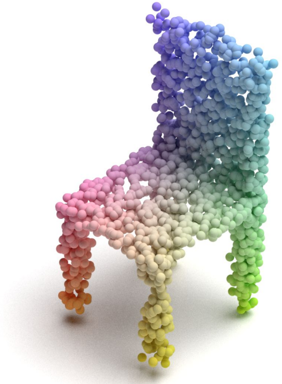
Point Cloud array shape- (474, 2048, 3)
474 point clouds each with 474 generated types of categories i.e different variations of planes.

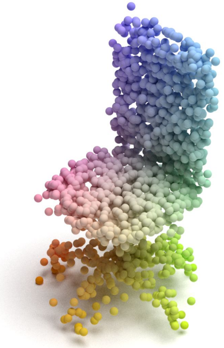
2048 points of 3 coordinates x,y,z

Visualized using mitsubi and open3d open source python libraries.

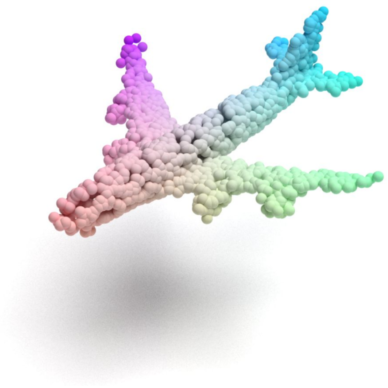
Chair

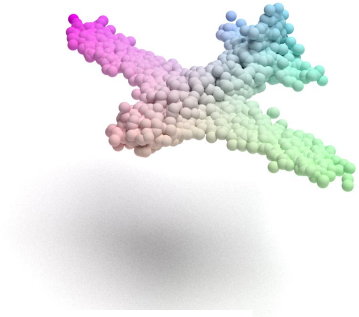


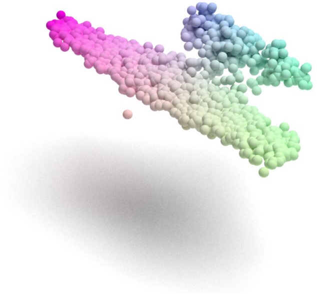




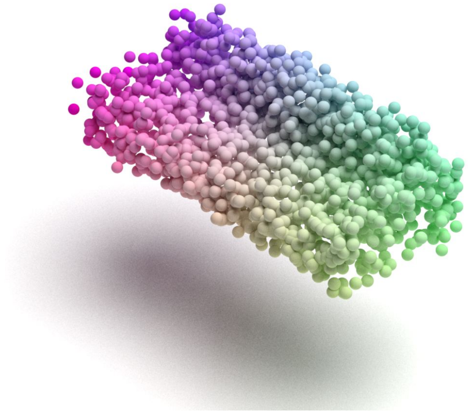
Plane

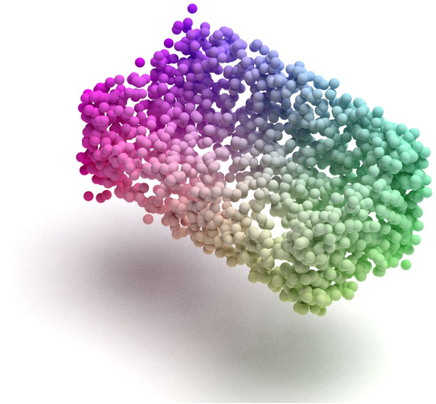




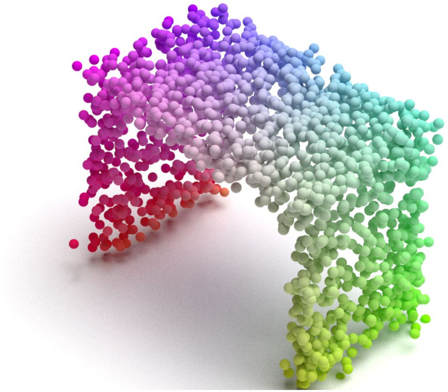


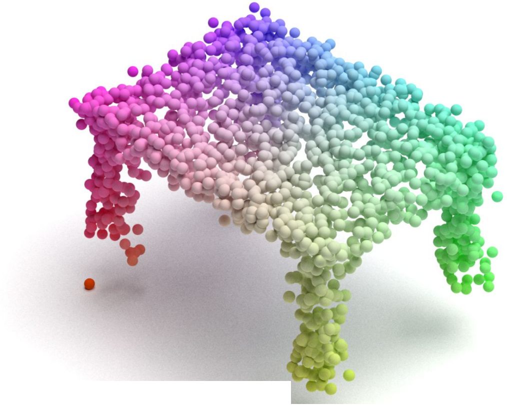
Sofa

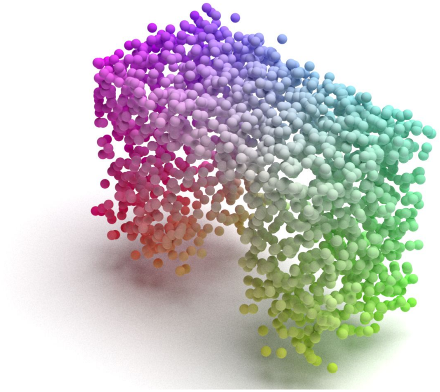




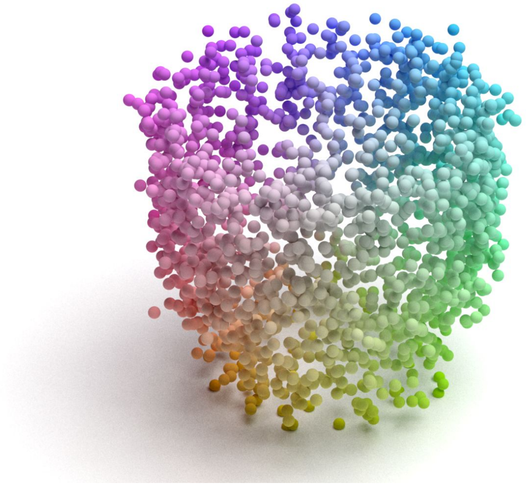
Table

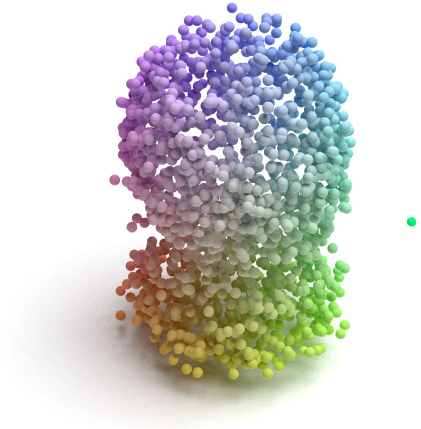






Lamp





Conclusion

- The paper presented a novel probabilistic generative model for point clouds, taking inspiration from the diffusion process in non-equilibrium thermodynamics.
- The model leverages a reverse diffusion Markov chain, conditioned on a shape latent, providing a robust foundation for generating realistic point clouds.
- At the core of the methodology is a tractable training objective derived from the variational bound of point cloud likelihood. This objective empowers the model to effectively capture point cloud distributions, guided by the shape latent.
- Experimental results demonstrate that the proposed model achieves the state-of-the-art performance in point cloud generation and auto-encoding.

References

1. *Luo, Shitong and Wei Hu. "Diffusion Probabilistic Models for 3D Point Cloud Generation." 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021): 2836-2844.*
2. *Ho, Jonathan, Ajay Jain and P. Abbeel. "Denoising Diffusion Probabilistic Models." ArXiv abs/2006.11239 (2020): n. pag.*