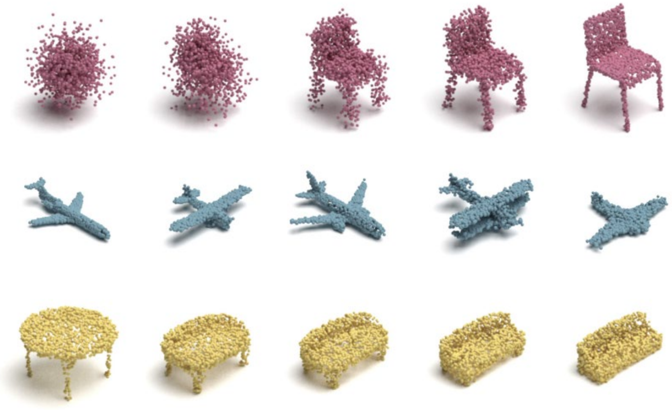
Diffusion Probabilistic Models for 3D Point Cloud Generation

Authors: Shitong Luo, Wei Hu
Presented by Ayush Agarwal

Problem Setting

Point Cloud Generation:

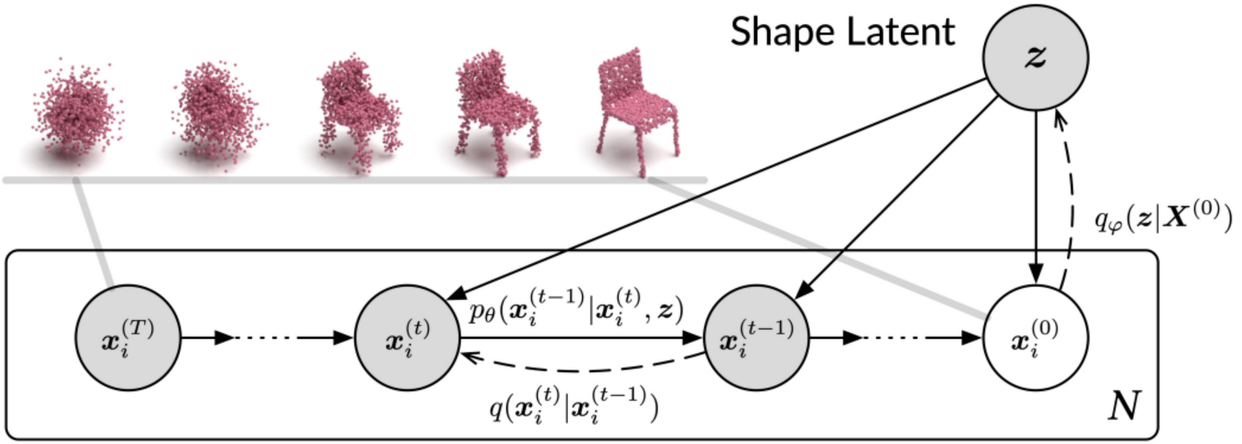
- Previous works use adversarial loss which is hard to optimize, are not permutationally invariant (e.g. autoregressive models), have limitations to generate a fixed number of points, or have an invertibility requirement



Paper Contributions

- Novel point cloud generative model motivated by Markov diffusion process in thermodynamics
  - reduces the learning objective to learning the Markov diffusion kernel in a simple functional form
- Mathematical formulation of tractable learning objective maximizing a variational lower bound on the likelihood of point clouds conditioned on shape latents
- Strong quantitative and qualitative results on point cloud generation, autoencoding, and unsupervised representation learning

Formulation of Diffusion Probabilistic Model



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

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

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

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

$$q(\boldsymbol{X}^{(1:T)}|\boldsymbol{X}^{0}) = \prod_{i=1}^{N} q(\boldsymbol{x}_i^{(1:T)}|\boldsymbol{x}_i^{(0)}),$$

$$p_{\boldsymbol{\theta}}(\boldsymbol{X}^{(0:T)}|\boldsymbol{z}) = \prod_{i=1}^{N} p_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{(0:T)}|\boldsymbol{z}).$$

Variational Training Objective

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

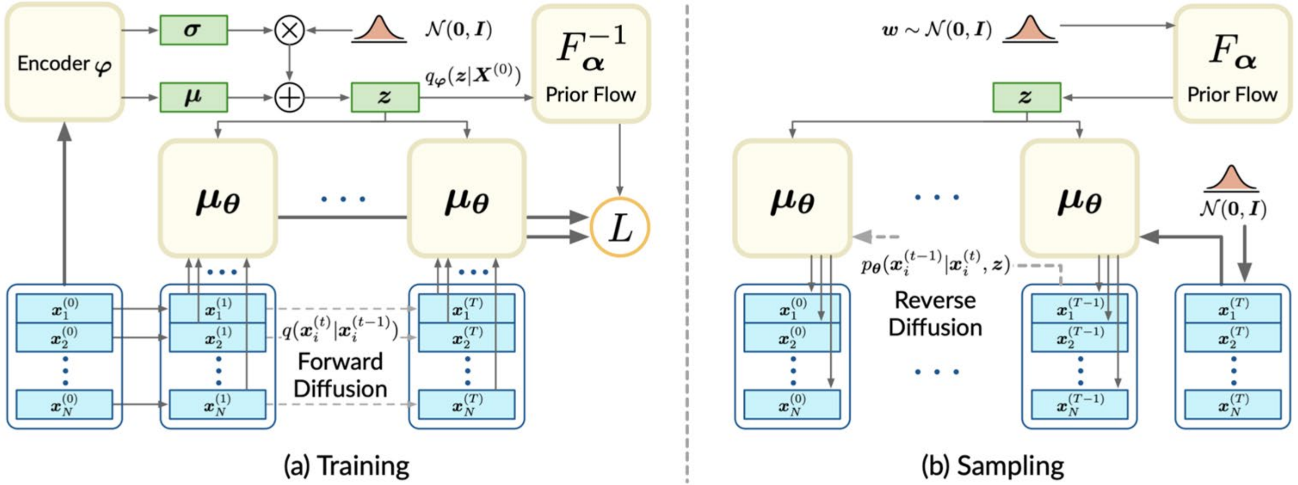
$$
\begin{aligned}
L(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \mathbb{E}_q\Big[&\sum_{t=2}^{T} \sum_{i=1}^{N} D_{\mathrm{KL}}\big(\underbrace{q(\boldsymbol{x}_i^{(t-1)} | \boldsymbol{x}_i^{(t)}, \boldsymbol{x}_i^{(0)})}_{\textcircled{1}} \,\|\, \underbrace{p_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{(t-1)} | \boldsymbol{x}_i^{(t)}, \boldsymbol{z})}_{\textcircled{2}}\big) \\
&- \sum_{i=1}^{N} \underbrace{\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i^{(0)} | \boldsymbol{x}_i^{(1)}, \boldsymbol{z})}_{\textcircled{3}} \\
&+ D_{\mathrm{KL}}\big(\underbrace{q_{\boldsymbol{\varphi}}(\boldsymbol{z} | \boldsymbol{X}^{(0)})}_{\textcircled{4}} \,\|\, \underbrace{p(\boldsymbol{z})}_{\textcircled{5}}\big)\Big].
\end{aligned}
$$

Algorithm 1 Training (Simplified)

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

Model Implementation

$$p(\boldsymbol{z}) = p_{\boldsymbol{w}}(\boldsymbol{w}) \cdot \left| \det \frac{\partial F_{\boldsymbol{\alpha}}}{\partial \boldsymbol{w}} \right|^{-1} \quad \text{where} \quad \boldsymbol{w} = F_{\boldsymbol{\alpha}}^{-1}(\boldsymbol{z}).$$



Point Cloud Generation Loss:

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

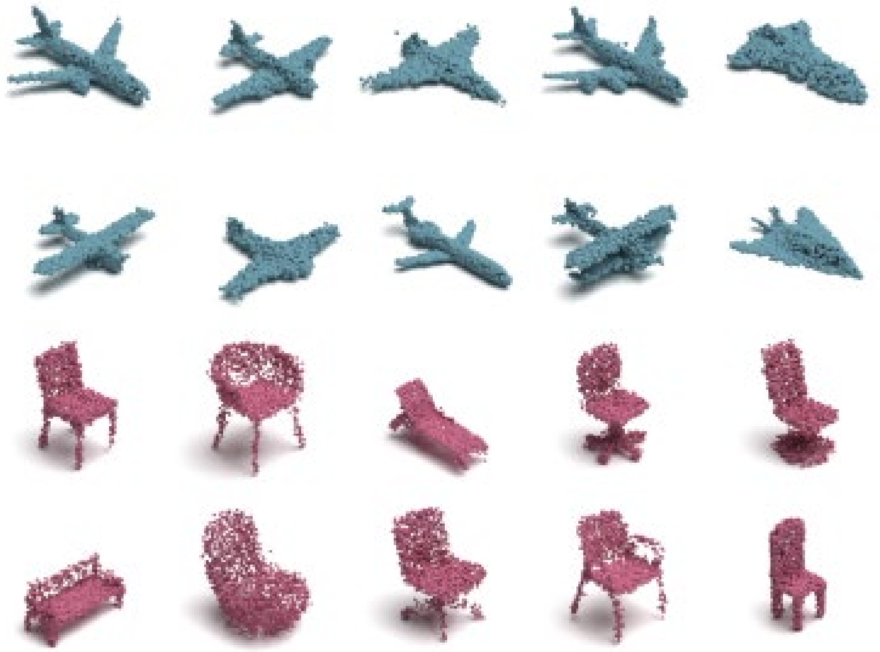
Autoencoding Loss:

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

Encoders use PointNet architecture for both point cloud generation and autoencoding

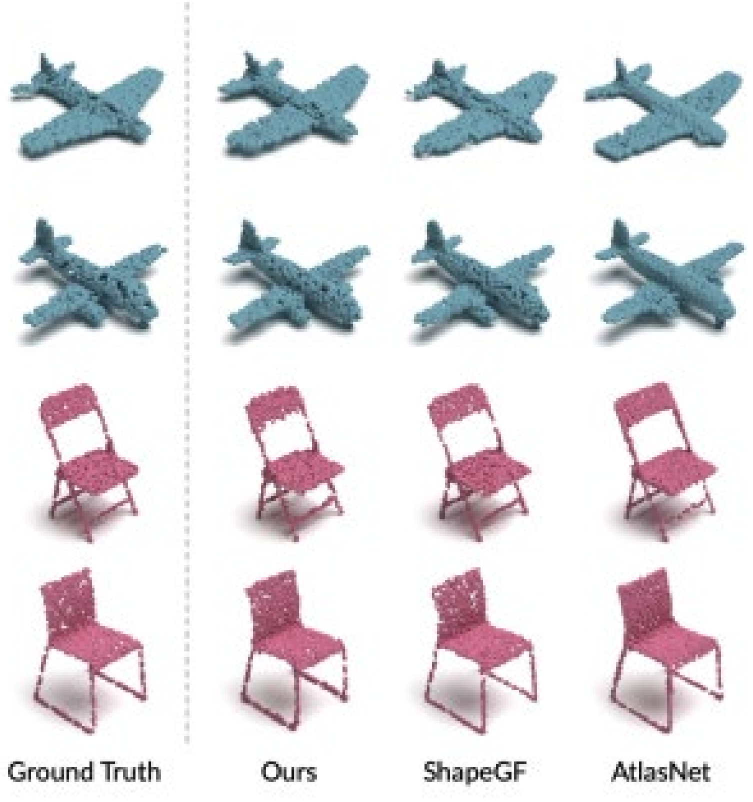
Point Cloud Generation Results

Shape	Model	MMD (↓) CD	MMD (↓) EMD	COV (%, ↑) CD	COV (%, ↑) EMD	1-NNA (%, ↓) CD	1-NNA (%, ↓) EMD	JSD (↓)
Airplane	PC-GAN [1]	3.819	1.810	42.17	13.84	77.59	98.52	6.188
	GCN-GAN [22]	4.713	1.650	39.04	18.62	89.13	98.60	6.669
	TreeGAN [19]	4.323	1.953	39.37	8.40	83.86	99.67	15.646
	PointFlow [25]	3.688	1.090	44.98	44.65	66.39	**69.36**	1.536
	ShapeGF [2]	3.306	**1.027**	**50.41**	**47.12**	**61.94**	70.51	**1.059**
	Ours	**3.276**	1.061	48.71	45.47	64.83	75.12	1.067
	Train	3.917	1.003	51.73	54.04	48.85	50.82	0.809
Chair	PC-GAN [1]	13.436	3.104	46.23	22.14	69.67	100.00	6.649
	GCN-GAN [22]	15.354	2.213	39.84	35.09	77.86	95.80	21.708
	TreeGAN [19]	14.936	3.613	38.02	6.77	74.92	100.00	13.282
	PointFlow [25]	13.631	1.856	41.86	43.38	66.13	68.40	12.474
	ShapeGF [2]	13.175	1.785	48.53	46.71	**56.17**	**62.69**	**5.996**
	Ours	**12.276**	**1.784**	**48.94**	**47.52**	60.11	69.06	7.797
	Train	13.954	1.756	53.29	54.90	49.14	48.28	3.602

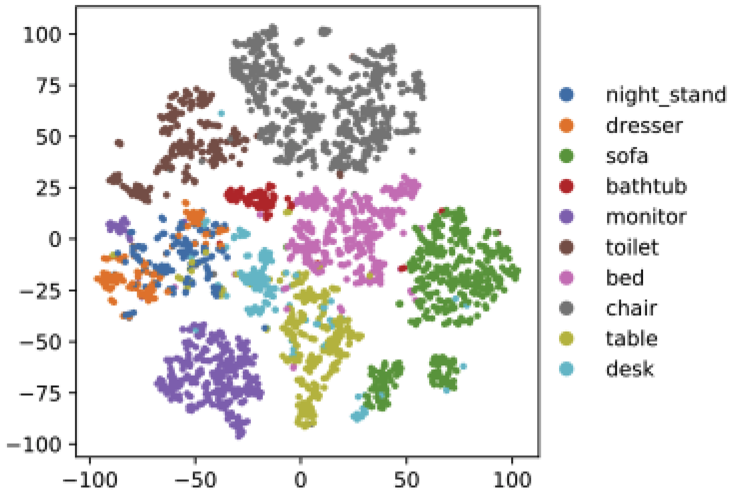


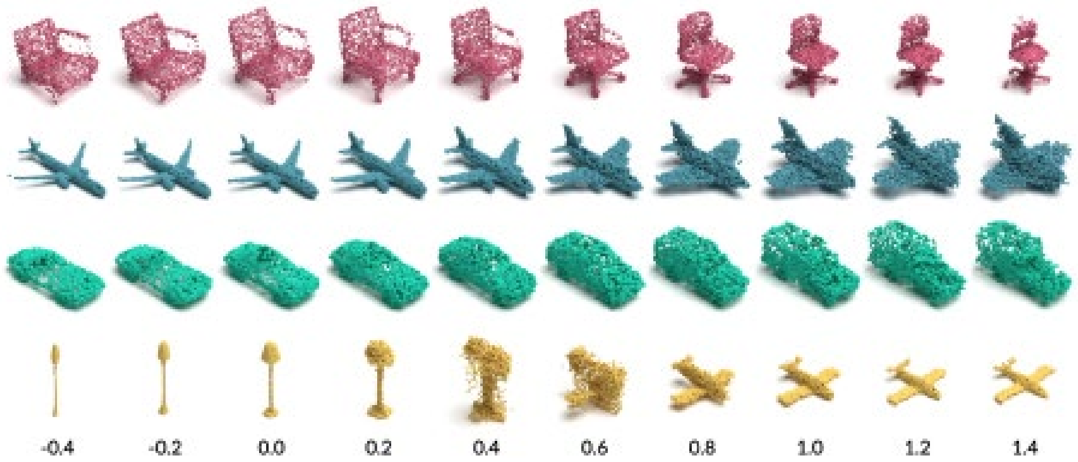
Autoencoding Results

Dataset	Metric	Atlas (S1)	Altas (P25)	PointFlow	ShapeGF	Ours	Oracle
Airplane	CD	2.000	**1.795**	2.420	2.102	2.118	1.016
	EMD	4.311	4.366	3.311	3.508	**2.876**	2.141
Car	CD	6.906	6.503	5.828	**5.468**	5.493	3.917
	EMD	5.617	5.408	4.390	4.489	**3.937**	3.246
Chair	CD	5.479	**4.980**	6.795	5.146	5.677	3.221
	EMD	5.550	5.282	5.008	4.784	**4.153**	3.281
ShapeNet	CD	5.873	5.420	7.550	5.725	**5.252**	3.074
	EMD	5.457	5.599	5.172	5.049	**3.783**	3.112



Unsupervised Representation Learning Results





Model	ModelNet10	ModelNet40
AtlasNet [10]	91.9	86.6
PC-GAN (CD) [1]	**95.4**	84.5
PC-GAN (EMD) [1]	**95.4**	84.0
PointFlow [25]	93.7	86.8
ShapeGF [2]	90.2	84.6
Ours	94.2	**87.6**

Paper Takeaways

Strengths:

- Intuitive method as Markov diffusion process
- Permutationally invariant
- Can sample as few/many points as needed
- Matches SOTA performance

Weaknesses/Future Improvements:

- Doesn’t improve much on SOTA performance
- Need to go through T steps of reverse diffusion in order to sample a point cloud