PointFlow: 3D Point Cloud Generation with Continuous Normalizing Flows

Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu,
Serge Belongie, Bharath Hariharan

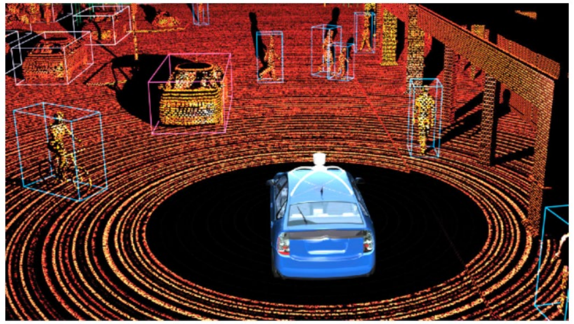
ICCV 2019

Presented by Te Gusi,Su Jiajun

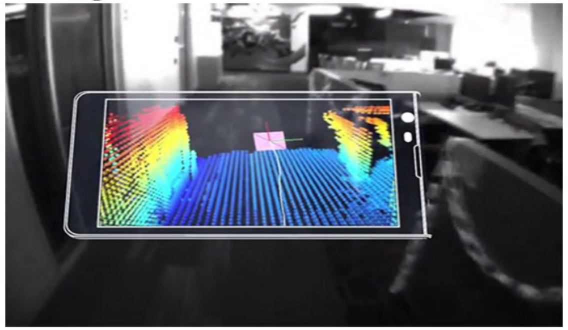
Table of contents

Background

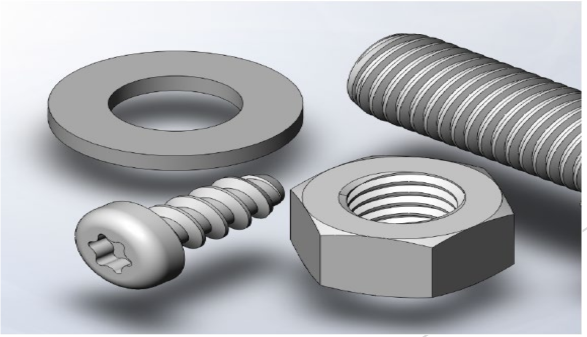
- 3D Data



Robot Perception



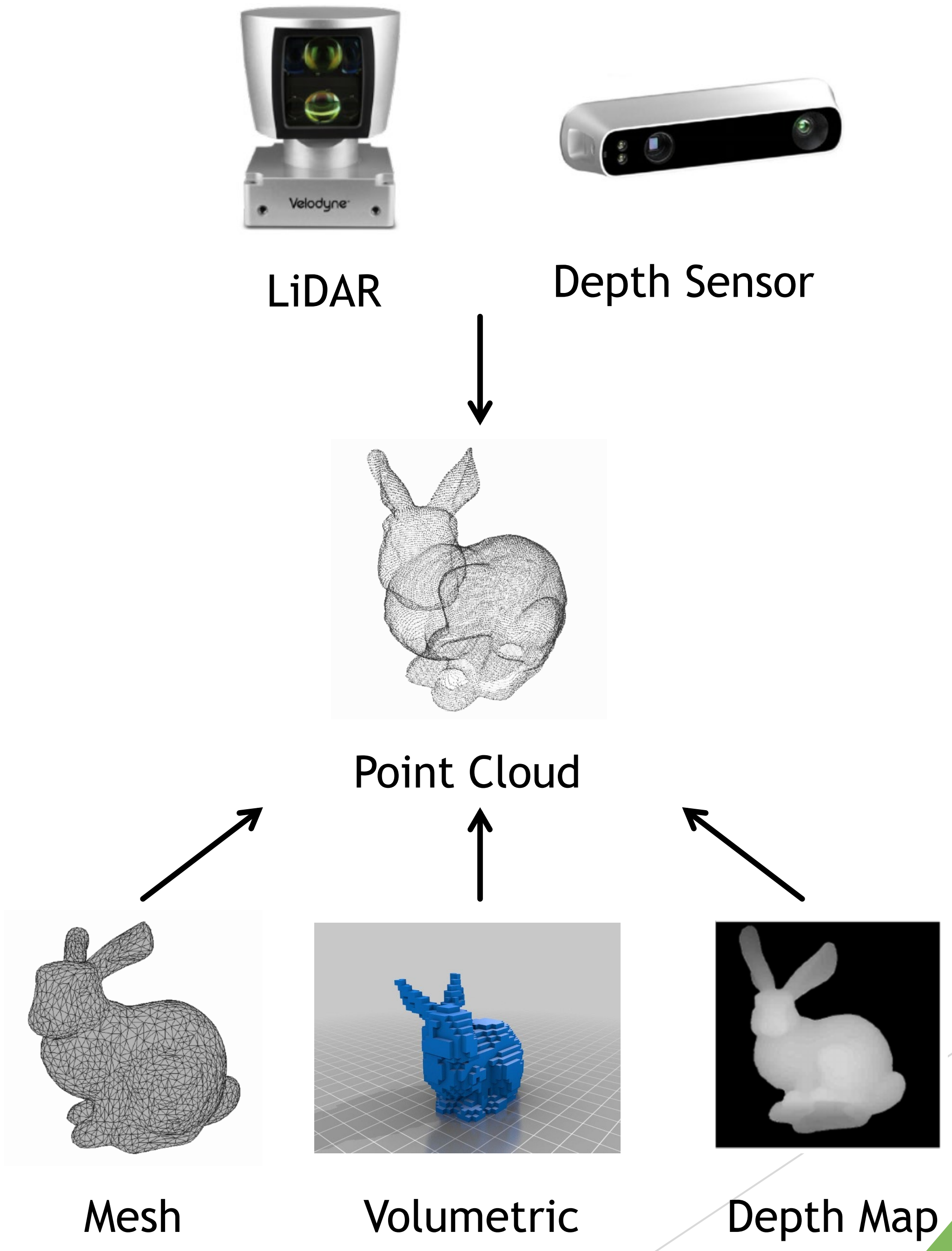
Augmented Reality



Shape Design

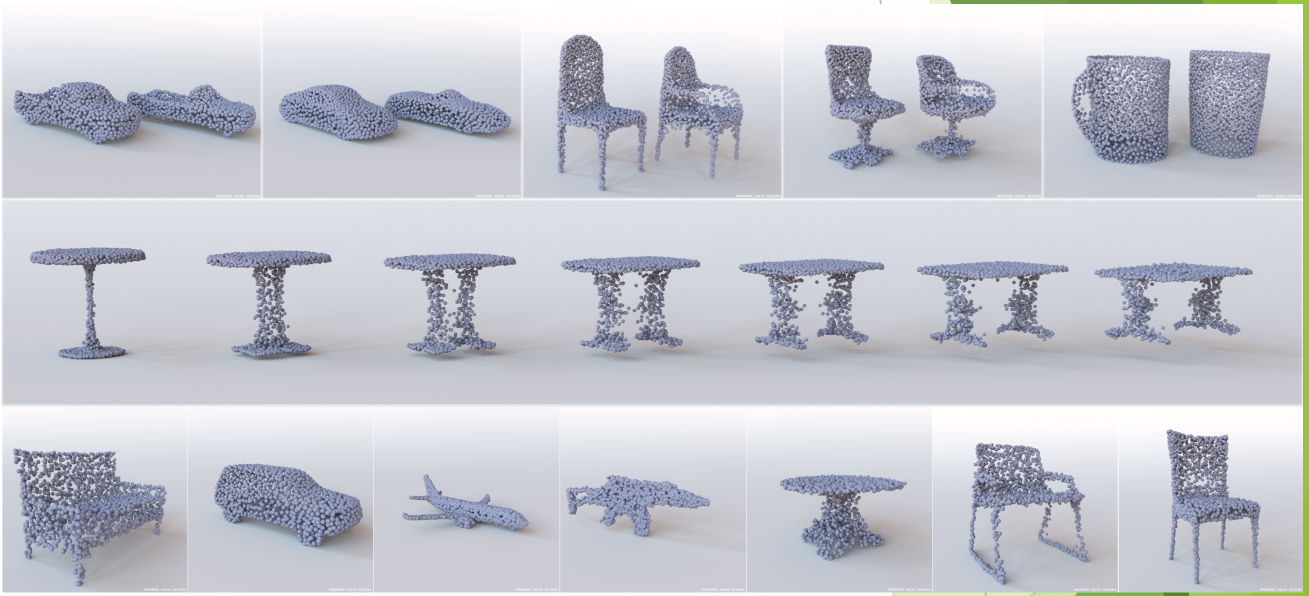
Background

- 3D Data
- 3D Geometry Representations



Background

- 3D Data
- 3D Geometry Representations
- Generative Learning
  - Auto Encoder[1]
  - GAN[2]
  - Autoregressive Model[3]



Challenge

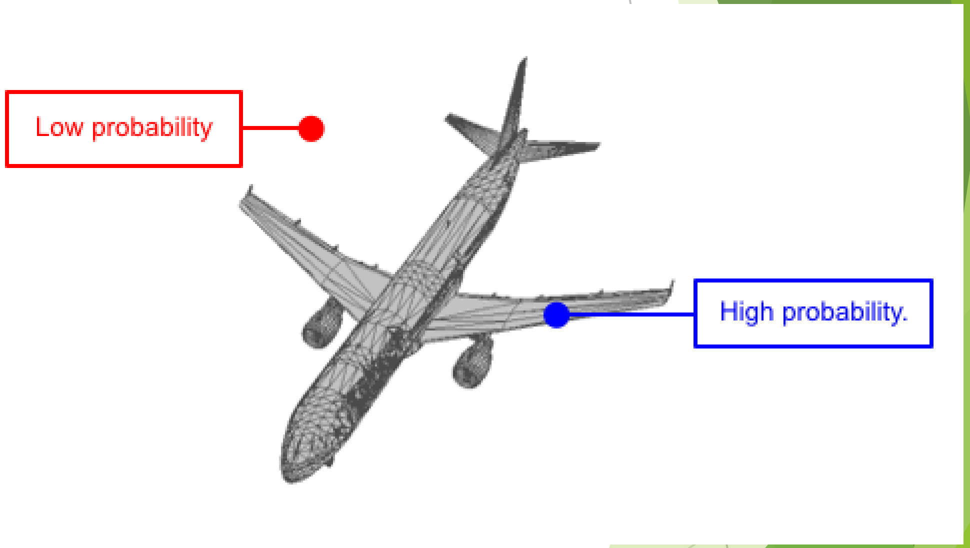
1. Modeling distribution of distribution

   Each sample is a distribution of points

   Overall shape is also a distribution

2. Estimating probability densities

   Implicit density of GAN models



Methods

- Continuous Normalizing Flow + VAE

Methods

Continuous Normalizing Flow

- Let $f_1, \ldots, f_n$ denote a series of invertible transformations, y denotes a latent variable.
- Probability density

$$\log P(x) = \log P(y) - \sum_{k=1}^{n} \log \left| \det \frac{\partial f_k}{\partial y_{k-1}} \right|$$

Methods

Continuous Normalizing Flow

- Extend to the continuous model by defining continuous-time dynamic

$$\frac{\partial y(t)}{\partial t} = f(y(t), t)$$

- Continuous normalizing flow(CNF) is formulated by

$$x = y(t_0) + \int_{t_0}^{t_1} f(y(t), t) dt, \quad y(t_0) \sim P(y)$$

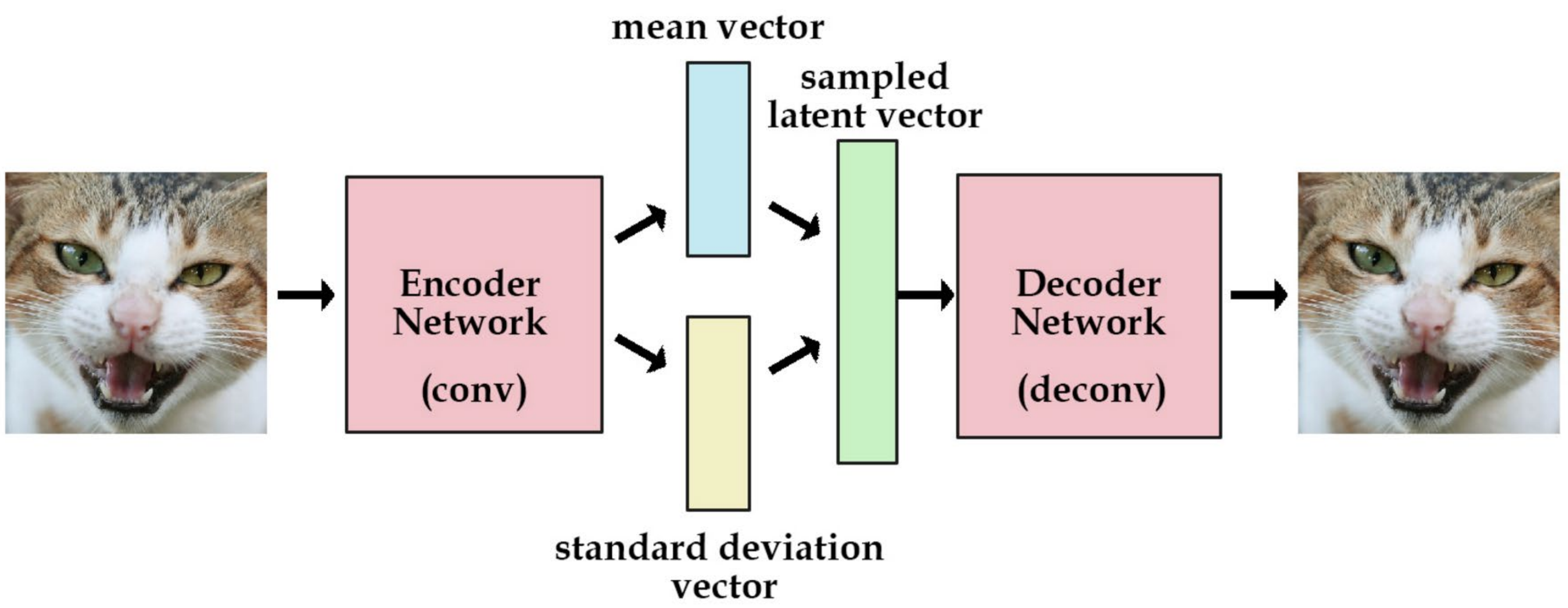
$$\log P(x) = \log P(y(t_0)) - \int_{t_0}^{t_1} \mathrm{Tr}\left(\frac{\partial f}{\partial y(t)}\right) dt$$

- We can apply ordinary differential equation(ODE) solver to estimate the output

Methods

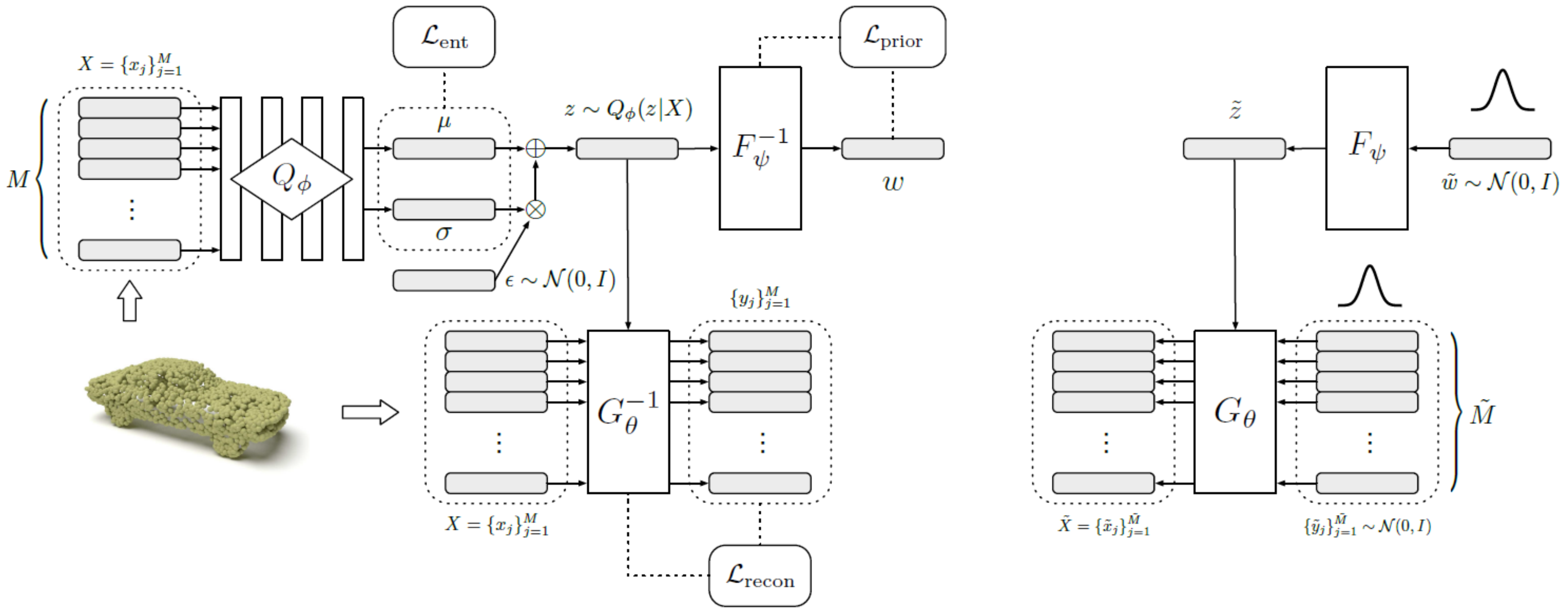
Variational Auto Encoder

- The variational auto-encoder (VAE) is a framework that allows one to learn P(X) from a dataset of observations of X.



Methods PointFlow

- Encoder: $Q_{\Phi}(z|X)$ encodes a point cloud into a shape representation z
- Decoder/Generator: $P_{\theta}(X|z)$ models the distribution of points given the shape representation
- Prior: $F_{\psi}(z|w)$ model the shape prior by transforming a simple Gaussian distribution w



(a) Training (Auto-encoding)

(a) Test (Sampling)

Methods PointFlow

- ELBO $\phi^*, \psi^*, \theta^* = \arg\max_{\phi,\psi,\theta} \sum_{X \in \mathcal{X}} \mathcal{L}(X; \phi, \psi, \theta).$
- Posterior Entropy: L_{ent}
  - Models the entropy of the approximated posterior

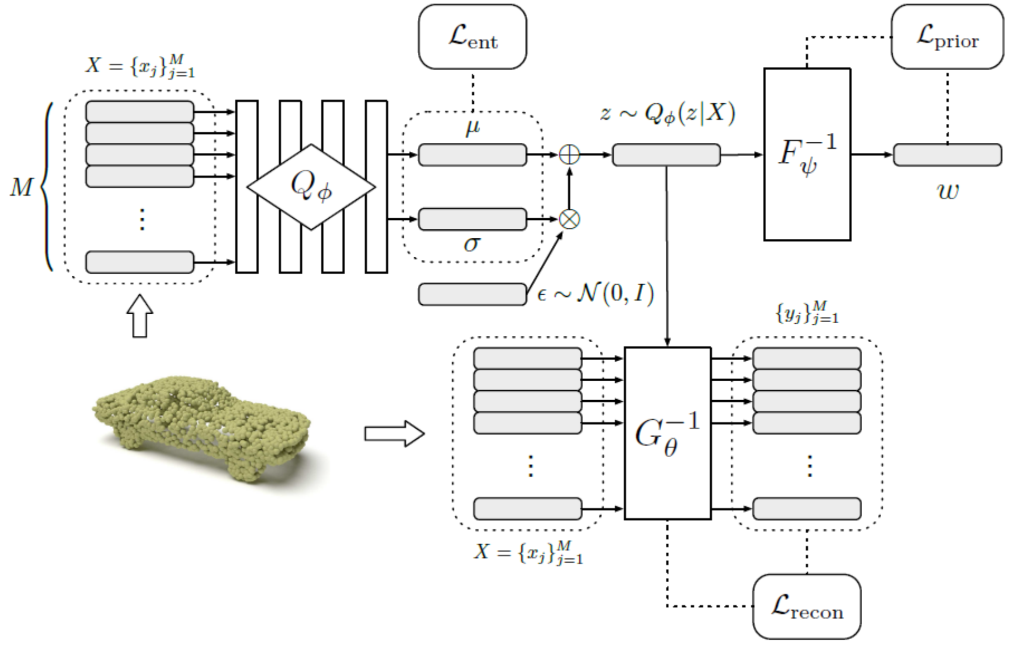
$$\mathcal{L}_{\text{ent}}(X;\phi) \triangleq H[Q_\phi(z|X)]$$

- Prior: L_{prior}
  - Encourages the encoded shape representation to have a high probability under the prior

$$\mathcal{L}_{\text{prior}}(X;\psi,\phi) \triangleq \mathbb{E}_{Q_\phi(z|x)}[\log P_\psi(z)]$$

- Reconstruction likelihood: L_{recon}
  - Describe the reconstruction log likelihood of the input point set

$$\mathcal{L}_{\text{recon}}(X;\theta,\phi) \triangleq \mathbb{E}_{Q_\phi(z|x)}[\log P_\theta(X|z)]$$



(a) Training (Auto-encoding)

Experiments

Measurements

- Point Cloud to Point Cloud
  - Chamfer Distance:

    Measures the squared distance between each point in one set to its nearest neighbor in the other set.
  - Earth Mover’s distance:

    Measures the squared distance between two bijection set.

Experiments

Measurements

- Sets/Distribution Pairwise
  - Jensen-Shannon Divergence (JSD)
  - Coverage (COV)
  - Minimum matching distance (MMD)
  - 1-nearest neighbor accuracy (1-NNA)

1-NN classifier classifies it as coming from **real** or **fake** according to the label of its nearest sample.

$$\text{1-NNA}(S_g, S_r) = \frac{\sum_{X \in S_g} \mathbb{I}[N_X \in S_g] + \sum_{Y \in S_r} \mathbb{I}[N_Y \in S_r]}{|S_g| + |S_r|}$$

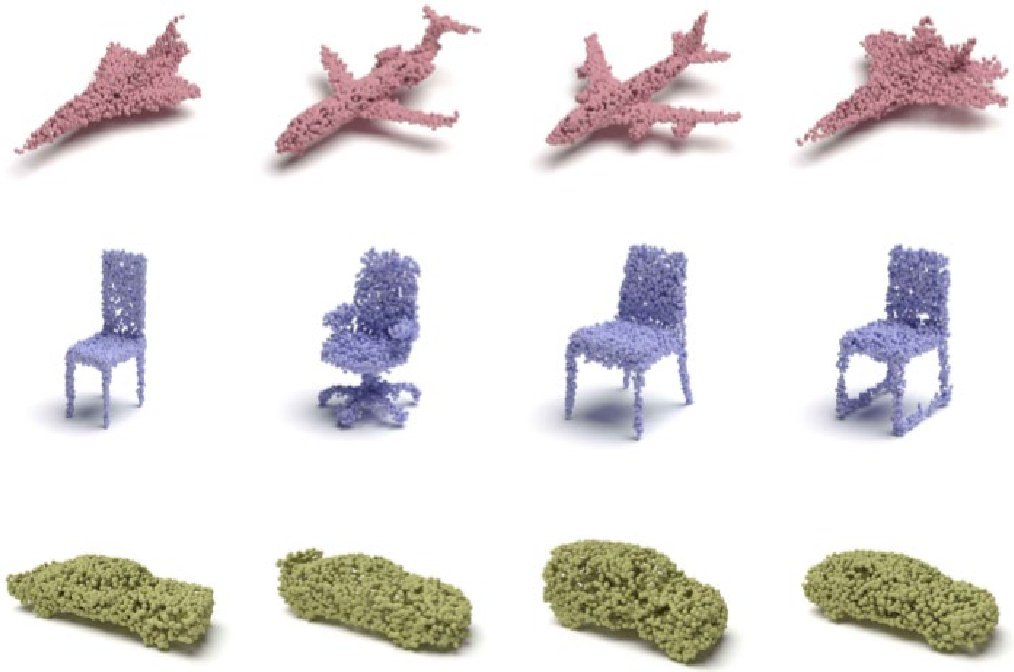
Experiments Result

Generation

Category	Model	# Parameters (M)		JSD (↓)	MMD (↓)		COV (%, ↑)		1-NNA (%, ↓)	
		Full	Gen		CD	EMD	CD	EMD	CD	EMD
Airplane	r-GAN	7.22	6.91	7.44	0.261	5.47	42.72	18.02	93.58	99.51
	l-GAN (CD)	1.97	1.71	4.62	0.239	4.27	43.21	21.23	86.30	97.28
	l-GAN (EMD)	1.97	1.71	**3.61**	0.269	3.29	**47.90**	**50.62**	87.65	85.68
	PC-GAN	9.14	1.52	4.63	0.287	3.57	36.46	40.94	94.35	92.32
	PointFlow (ours)	**1.61**	**1.06**	4.92	**0.217**	**3.24**	46.91	48.40	**75.68**	**75.06**
	Training set	-	-	6.61	0.226	3.08	42.72	49.14	70.62	67.53
Chair	r-GAN	7.22	6.91	11.5	2.57	12.8	33.99	9.97	71.75	99.47
	l-GAN (CD)	1.97	1.71	4.59	2.46	8.91	41.39	25.68	64.43	85.27
	l-GAN (EMD)	1.97	1.71	2.27	2.61	**7.85**	40.79	41.69	64.73	65.56
	PC-GAN	9.14	1.52	3.90	2.75	8.20	36.50	38.98	76.03	78.37
	PointFlow (ours)	**1.61**	**1.06**	**1.74**	**2.42**	7.87	**46.83**	**46.98**	**60.88**	**59.89**
	Training set	-	-	1.50	1.92	7.38	57.25	55.44	59.67	58.46
Car	r-GAN	7.22	6.91	12.8	1.27	8.74	15.06	9.38	97.87	99.86
	l-GAN (CD)	1.97	1.71	4.43	1.55	6.25	38.64	18.47	63.07	88.07
	l-GAN (EMD)	1.97	1.71	2.21	1.48	5.43	39.20	39.77	69.74	68.32
	PC-GAN	9.14	1.52	5.85	1.12	5.83	23.56	30.29	92.19	90.87
	PointFlow (ours)	**1.61**	**1.06**	**0.87**	**0.91**	**5.22**	**44.03**	**46.59**	**60.65**	**62.36**
	Training set	-	-	0.86	1.03	5.33	48.30	51.42	57.39	53.27

Experiments Result

Generation



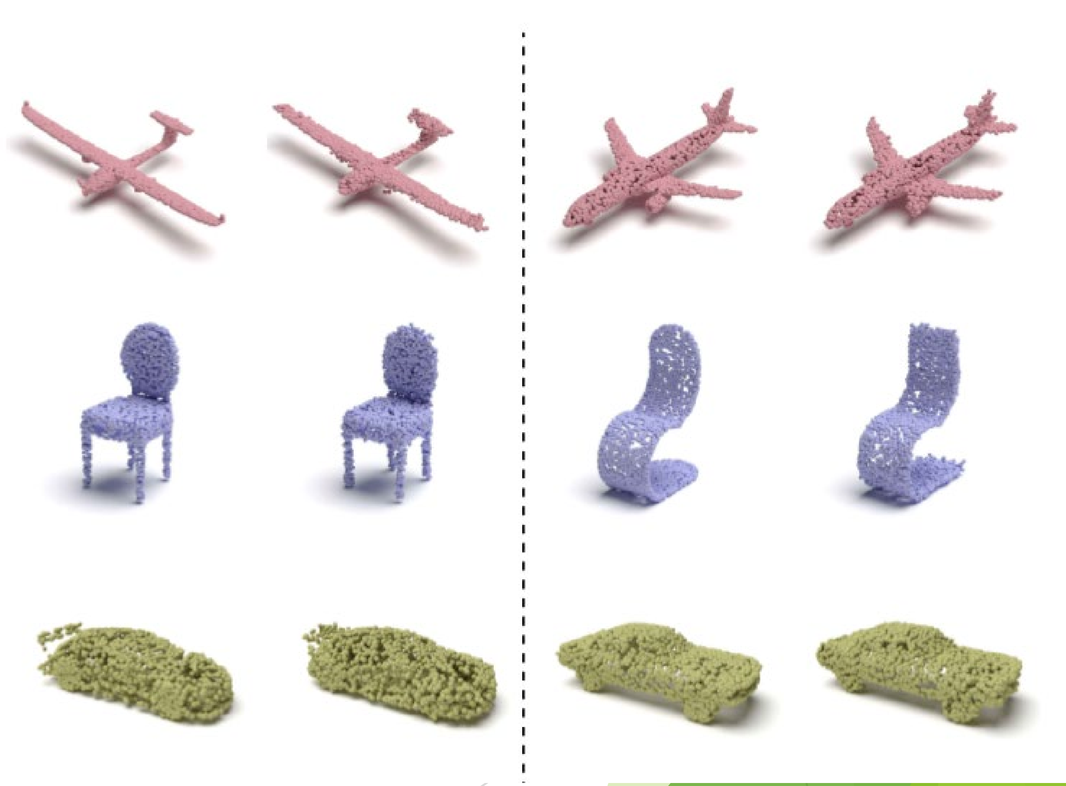
Examples of generated point clouds

Experiments Result

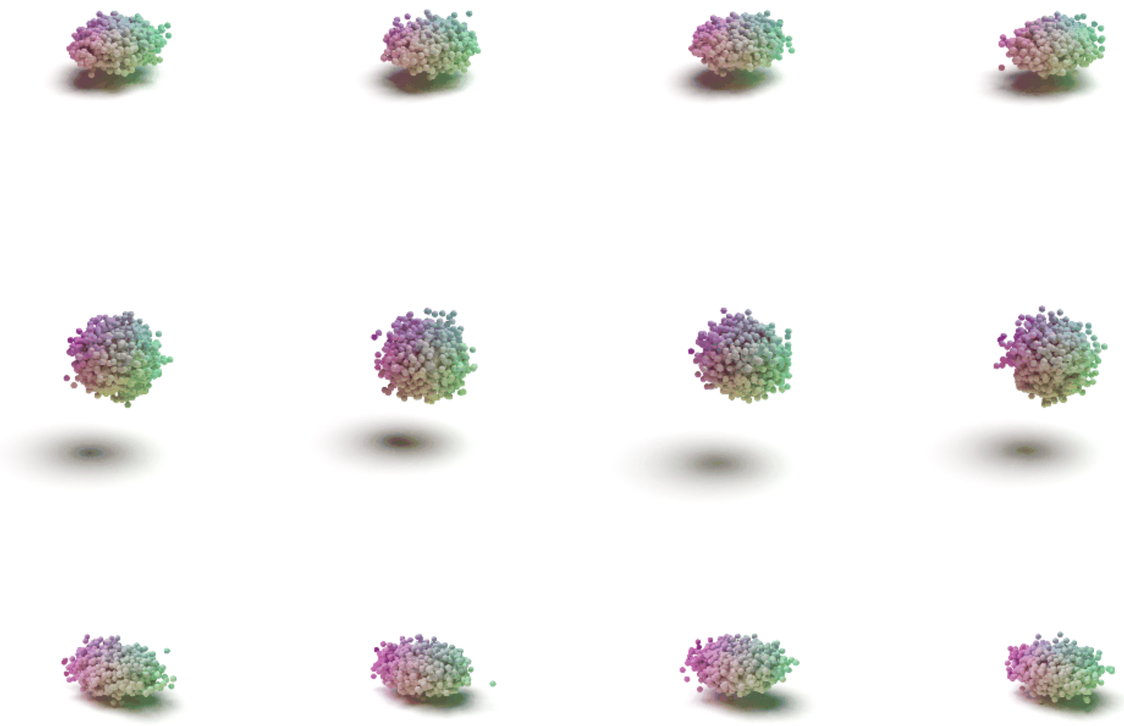
Reconstruction

Model	# Parameters (M)	CD	EMD
l-GAN (CD) [1]	1.77	7.12	7.95
l-GAN (EMD) [1]	1.77	8.85	5.26
AtlasNet [17]	44.9	**5.13**	5.97
PointFlow (ours)	**1.30**	7.54	**5.18**

Trained with reconstruction loss only



Experiments Result



Conclusion

- Propose a point cloud generative model based on normalizing flow
- Modeling point cloud and shape with different distribution
- Future work could be extended to multimodality

Reference

1. Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In ICML, 2018.
2. Yongbin Sun, Yue Wang, Ziwei Liu, Joshua E Siegel, and Sanjay E Sarma. Pointgrow: Autoregressively learned point cloud generation with self-attention.
3. 3D Point Cloud Generative Adversarial Network Based on Tree Structured Graph Convolutions. Dong Wook Shu, Sung Woo Park, and Junseok Kwon. In ICCV, 2019