ChartPointFlow for Topology-Aware 3D Point Cloud Generation

Takumi Kimura*, Takashi Matsubara†, and Kuniaki Uehara‡
***Kobe University, †Osaka University, ‡Osaka Gakuin University**
kimura@ai.cs.kobe-u.ac.jp

ACM multimedia Chengdu, China OCT 20-24 2021

Introduction

Background

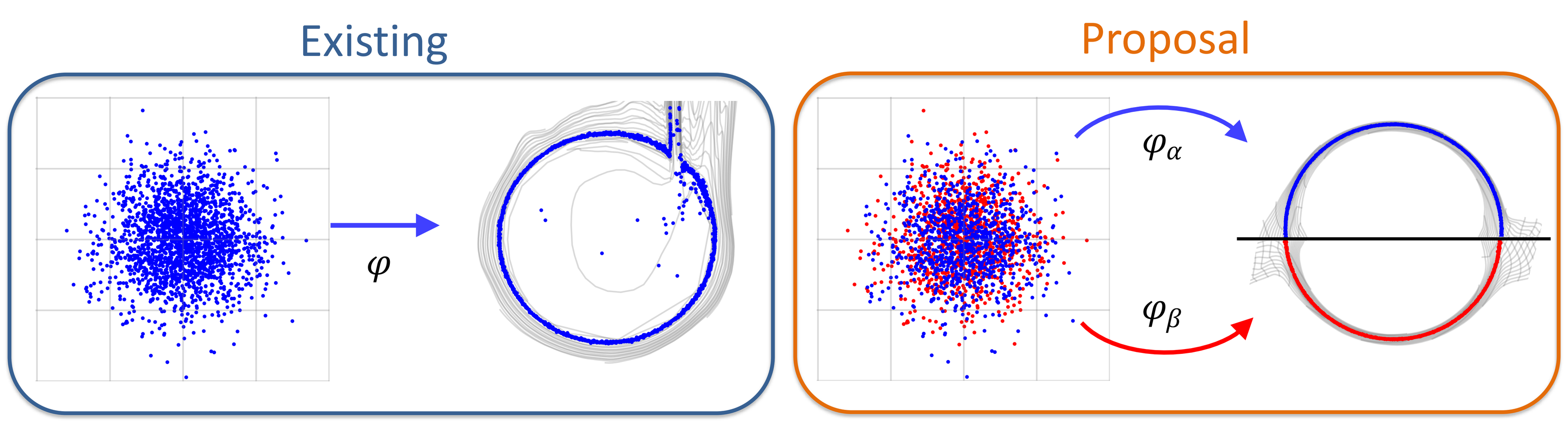
- A point cloud is becoming popular because it can capture high-resolution and is easier to manipulate.
- Point cloud generative model is useful for 3D computer vision tasks, such as shape completion, shape synthesis, and super-resolution.
- Most studies trained flow-based generative models to maximize the likelihood instead of a heuristic quality measurement of a generated point cloud.

Difficulties

- A flow-based generative model fails in expressing a point cloud that has a manifold-like structure because
  - a bijective map does not exist between a Euclidean space and a manifold with holes.
  - a point cloud is often composed of multiple subparts, some of which are disconnected.

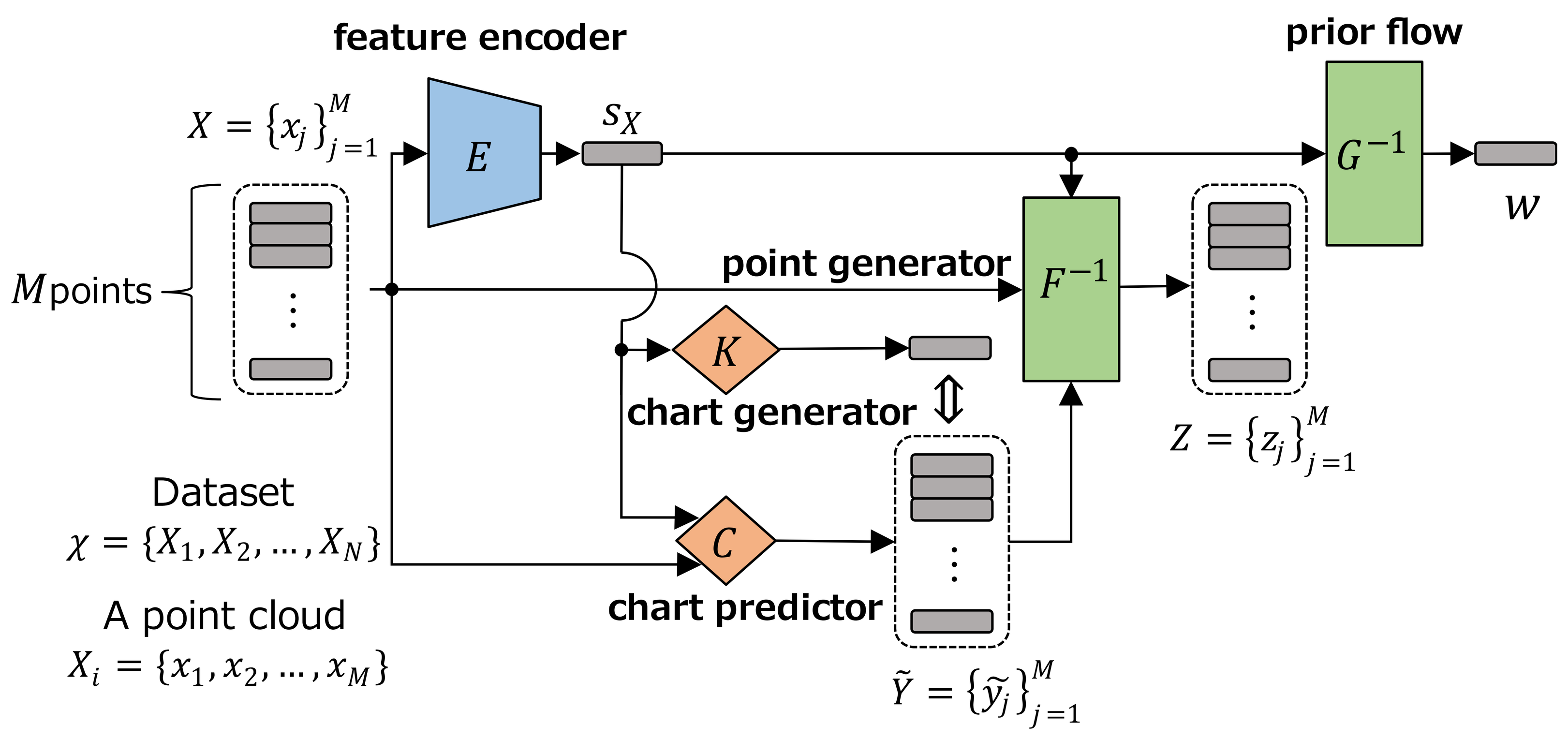
Proposal

- We employed flow-based generative model using multiple latent labels, each of which is
  - assigned to a continuous subset of a given point cloud in unsupervised manner.
  - corresponding to a map, similarly to a chart of a manifold.
- A set of charts forms an atlas that covers the entire point cloud.



Proposed Method

Architecture and data flow



- (C) The chart predictor infers the label y that corresponds to the chart that the point x belongs to.
- (F) The point generator learns a set of maps, given the label as a condition.
- (K) The chart generator learns the posterior $p_K(y|s_X)$ of the label y for generation task.

Objective

- The evidence lower bound (ELBO)

$$\log p(X) \geq \mathbb{E}_{q_E(s_X|X)q_C(Y|X,s_X)}\left[\log \frac{p_F(X|Y,s_X)p(Y)p_G(s_X)}{q_C(Y|X,s_X)q_E(s_X|X)}\right]$$
$$= \mathbb{E}_{q_E(s_X|X)}\left[\sum_j \left\{\mathbb{E}_{q_C(y_j|x_j,s_X)}\left[\log p_F(x_j|y_j,s_X)\right] + H[q_C(y_j|x_j,s_X)] - H[q_C(y_j|x_j,s_X)|p(y_j)]\right\}\right] - D_{KL}(q_E(s_X|X)\|p_G(s_X))$$
$$=: \mathcal{L}_{ELBO}(F,C,E,G;X).$$

- The regularization term to assign a map to a continuous subset

$$\mathcal{L}_{MI}(C;X,\mu,\lambda)\sum_j \left\{\mu H\left[\frac{1}{|X|}\sum_{\tilde{x}\in X} q_C(y_j|\tilde{x})\right] - \lambda H[q_C(y_j|x_j)]\right\}.$$

μ, λ: coefficient

- The final objective function to be maximized

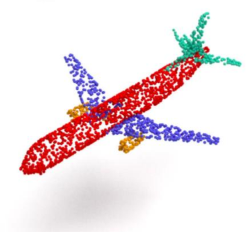
$$\mathcal{L}(F,C,E,G,K;\mathcal{X},\mu,\lambda) = \sum_{X\in\mathcal{X}}\left[\tilde{\mathcal{L}}_{\mathrm{ELBO}} + \mathcal{L}_{MI}\right].$$

Experiments

Generation and Reconstruction

- Dataset: ShapeNet. [Chang+, arXiv2015]
  - Three categories: airplane, chair, and car. [Yang+, ICCV2019]

Unsupervised Segmentation



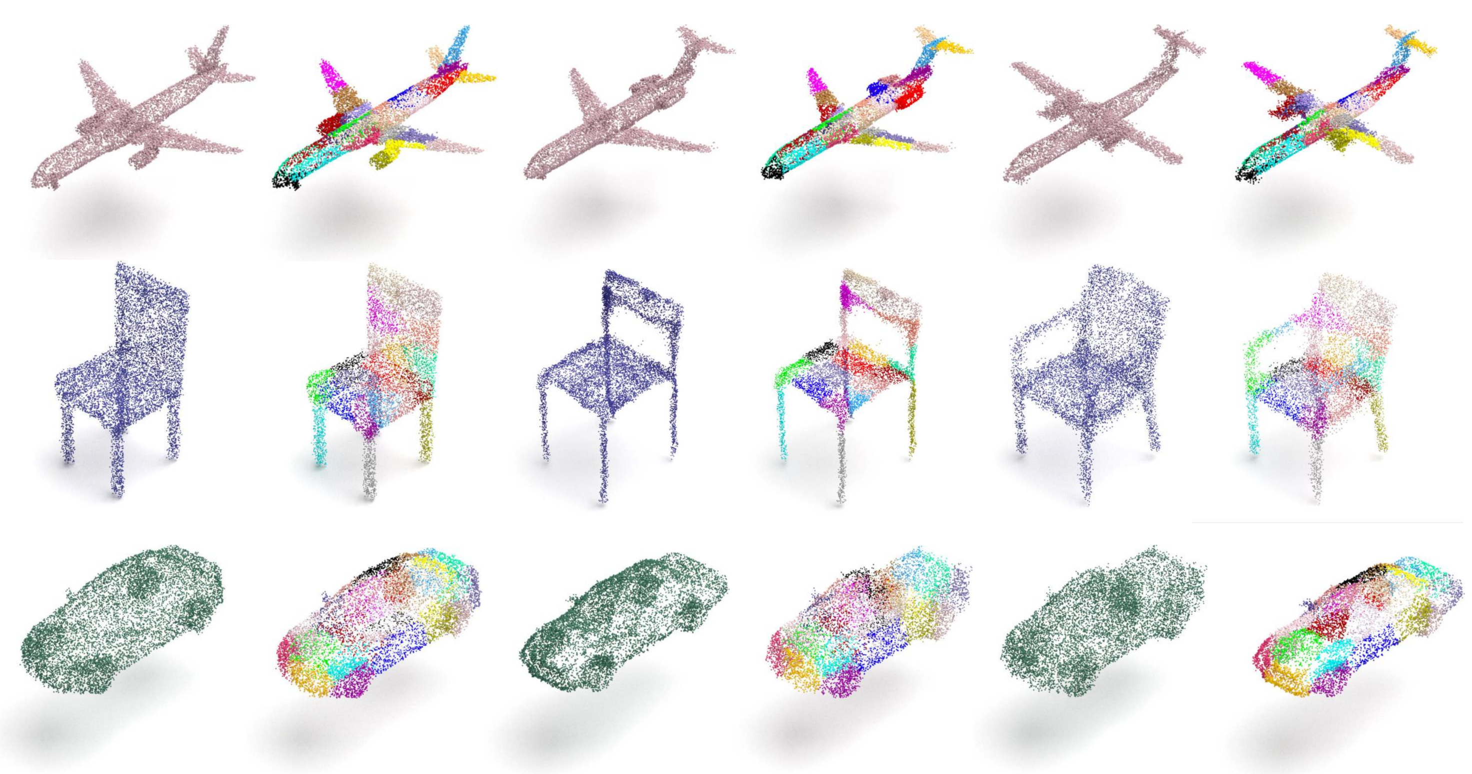
- Dataset: PartDataset. [Yi+, SIGGRAPH Asia2016]
  - Three categories: airplane, chair, and car.
  - divided into four parts.

Results

Generation (1-NNA, closer to 50% is better)

- ChartPointFlow outperformed other point cloud generators.

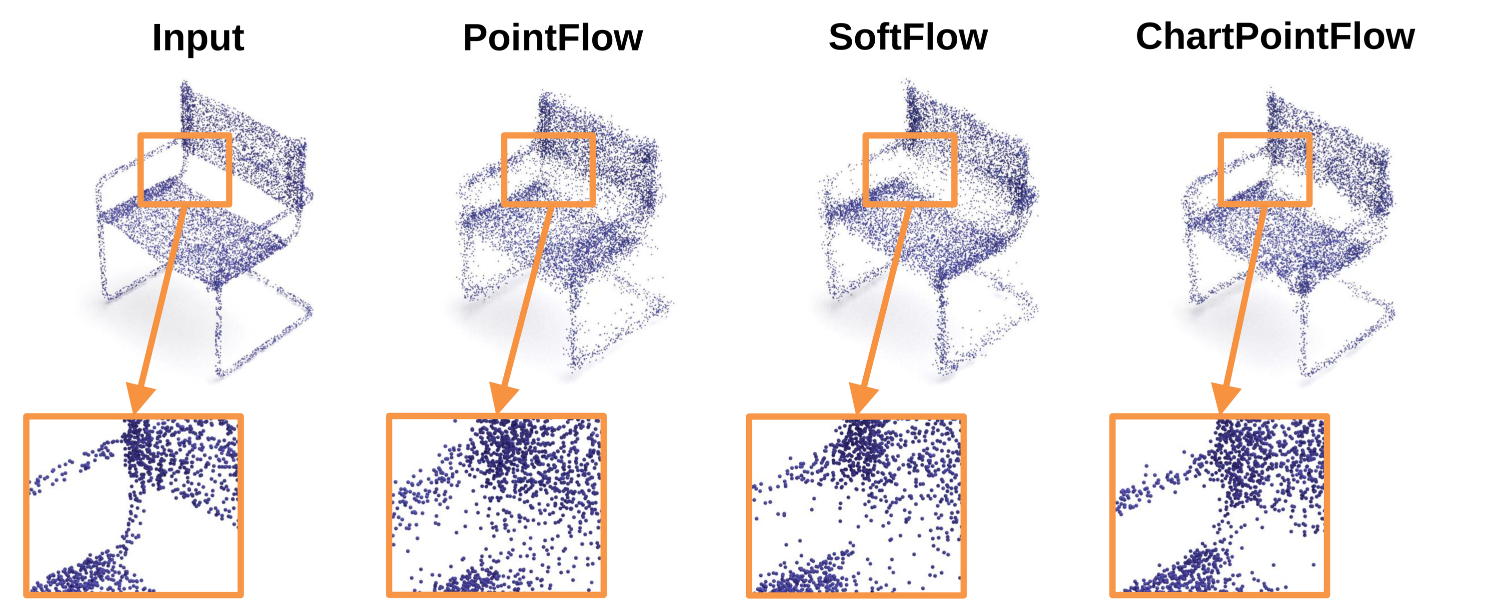
Model	Airplane	Chair	Car
l-GAN (EMD) [Achlioptas+, ICML2018]	85.68	65.56	68.32
PC-GAN [Li+, arXiv2018]	92.32	78.37	90.87
ShapeGF [Cai+, ECCV2020]	81.44	59.60	60.31
PointFlow [Yang+, ICCV2019]	75.06	59.89	62.36
SoftFlow [Kim+, NeurIPS2020]	69.44	63.51	64.71
ChartPointFlow (proposed)	**65.08**	**58.31**	**58.68**



Generation examples by ChartPointFlow (each color represents a chart).

Reconstruction

- State-of-the-art (see the manuscript for numerical performances).
- The large improvement in the chair category, especially compared to models using flows, namely PointFlow and SoftFlow.

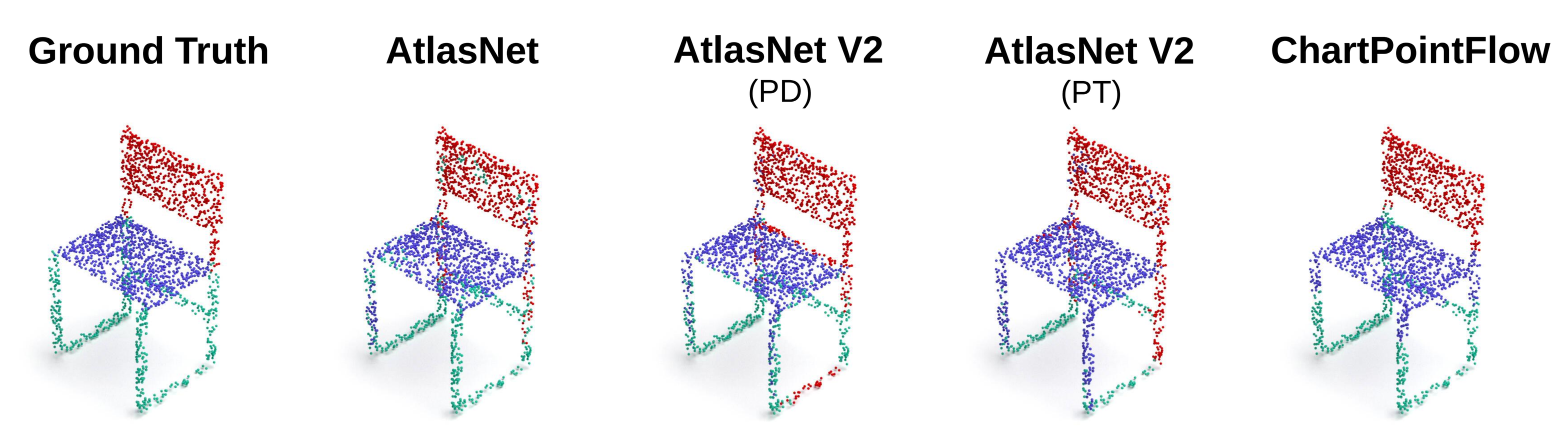


Reconstruction examples by each model.

Unsupervised Segmentation (NMI / purity, larger is better)

- ChartPointFlow outperformed AtlasNets for both criteria in all categories, except for the purity of the airplane.

Model	Airplane	Chair	Car
AtlasNet [Groueix+, CVPR2018]	0.22 / 0.76	0.23 / 0.74	0.11 / 0.71
AtlasNet V2 (PD) [Deprelle, NeurIPS2019]	0.25 / 0.79	0.24 / 0.75	0.13 / 0.72
AtlasNet V2 (PT) [Deprelle, NeurIPS2019]	0.27 / **0.80**	0.24 / 0.74	0.17 / 0.73
ChartPointFlow (proposed)	**0.30 / 0.80**	**0.35 / 0.86**	**0.19 / 0.79**



Results of unsupervised segmentation.