CS348n: Neural Representations and Generative Models for 3D Geometry

Learning Gradient Fields for Shape Generation

Ruojin Cai*, Guandao Yang*, Hadar Averbuch-Elor, Zekun Hao,
Serge Belongie, Noah Snavely, and Bharath Hariharan

Date: 31 January 2022
Presenter: Yikai Li
Instructor: Leonidas Guibas

What is this paper trying to achieve?

Given: A dataset of points captured of objects of various classes

Goal: Create a **generative** model for point clouds that represent these objects

Motivation

Point clouds are only **part** of the picture

- Samples of physical objects
- Want to build a model from samples

Why is a **model** useful?

- Densify sparse data (LiDAR)
- Object completion (Occlusion)
- Determine other properties

Prior Works

Point Cloud Modelling

- AtlasNet
- PointFlow
- PointGrow
- Point Cloud GAN
- Achlioptas et al

+ No post-processing required
+ Computationally efficient

- Fixed point cloud size
- Heuristic distance functions
- Assumed ordering
- Unstable Training

3D Representations

Voxels

- Wu et al
- Girdhar et al

Meshes

- Scape
- Dyna

Implicit Representation

- DeepSDF
- OccupancyNet

+ Model 3D distribution directly

- Require Signed Distance Function GT
- Computationally Expensive

Energy-Based Modelling

- Score Matching
- Denoising Score Matching
- Sliced Score Matching
- Song et al

This paper builds on this approach

+ No need to normalize probability
+ Computationally Efficient

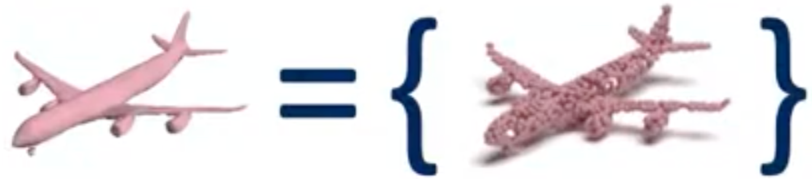
Contributions

- Novel generative model of objects given their point clouds
- Useful because of robustness to sparsity and occlusion
- Prior works are computationally expensive and unstable
- Key insights
  - Can generate implicit surfaces without generating meshes
  - Can generate new objects by interpolating latent space
- State of the art performance on auto-encoding and generation of point clouds

Problem Setting

Set of Shapes $\mathcal{X} = \{X^{(i)}\}_{i=1}^{N}$

Set of Points for a shape $X^{(i)} = \{x_j^i\}_{j=1}^{M_i}$



Surface S

Uniform Distribution on Surface $P_S(x)$

Gaussian Kernel Approximation $Q_{\sigma,S}(x) = \int_{\mathbf{s}\in\mathbb{R}^3} P_S(s)\mathcal{N}(x; s, \sigma^2 I)$

Problem Setting

Idea: Model the probability of the surface: $Q_{\sigma,S}(x)$

- Approach of prior works (Implicit Representation)
- Computational limitations of this approach

Alternative Idea: Model the gradient of the log-probability: $\nabla_x \log Q_{\sigma,S}(x)$

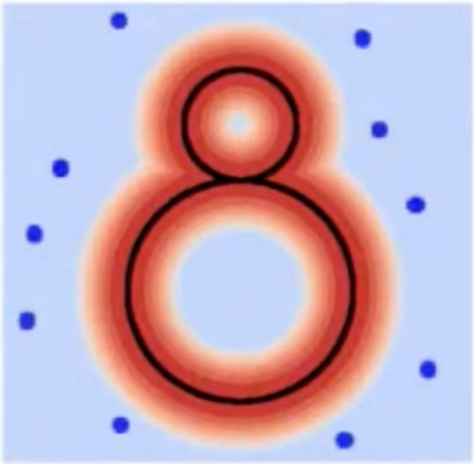
- Start with a general prior distribution (e.g. Gaussian Noise)
- Iteratively move points in direction of steepest ascent
- Can arbitrarily set the number of sampled points
- Can prove direction of gradient is to closest point on surface

Neural Network: $g_\theta(x, \sigma) \cong \nabla_x \log Q_{\sigma,S}(x)$

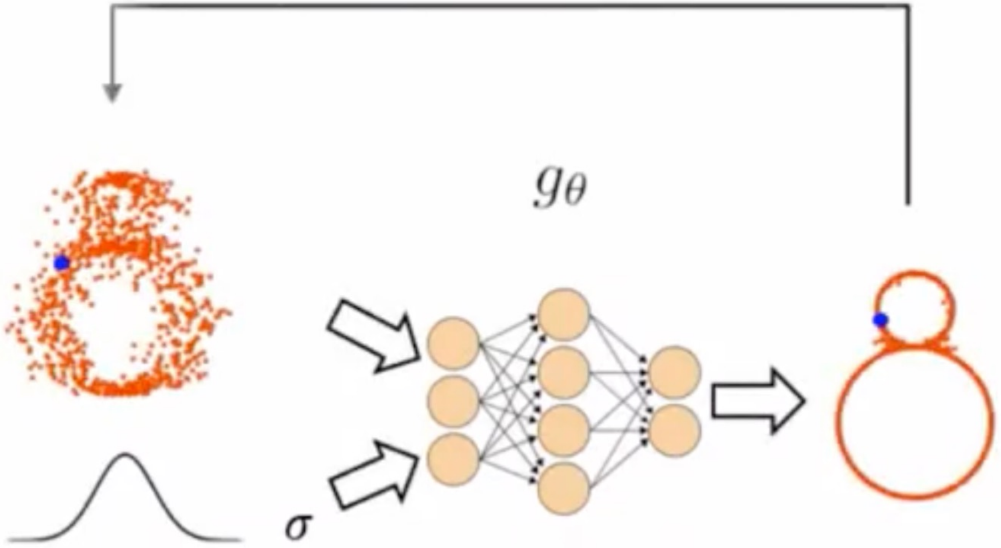
Method

Point Cloud Sampling

- Start with a random distribution go in direction of gradient
- Add random noise (Annealed Langevin Dynamics)
- 10 Iterations going from 1 to 0.01 of Gaussian noise



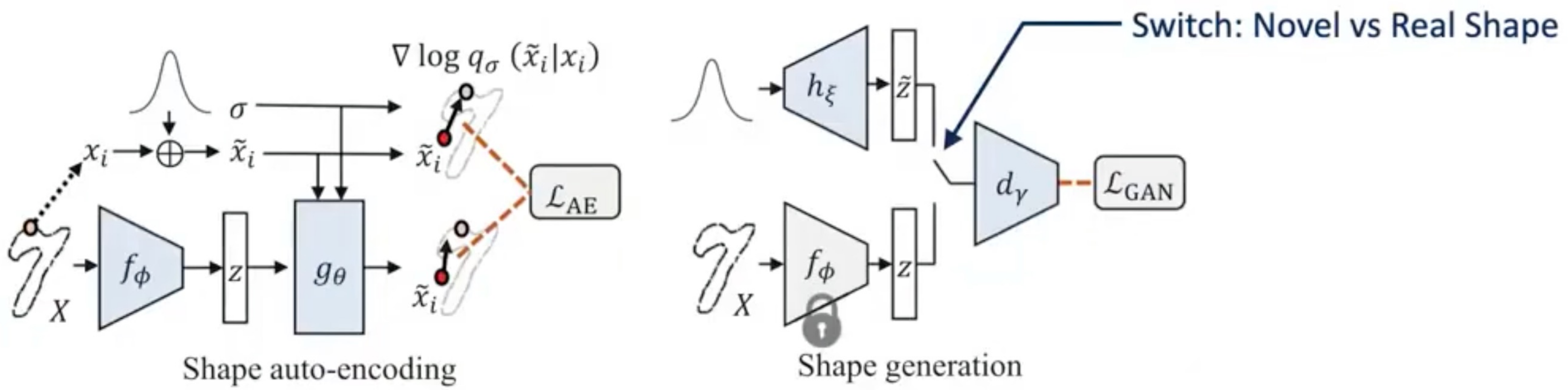
Probability



Method

Generating Multiple Shapes

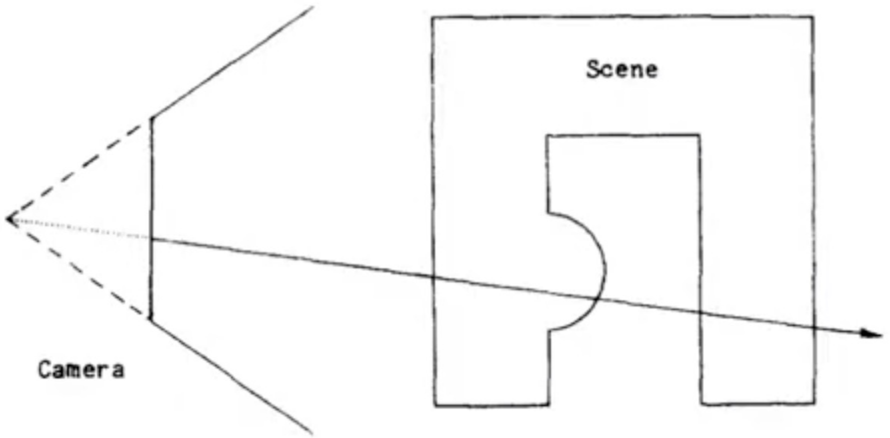
- Auto-encoder encodes input to latent variable Z
- Train using a Generative Adversarial Network (GAN)
- Generate realistic novel shapes using GAN latent variable $\tilde{Z}$



Method

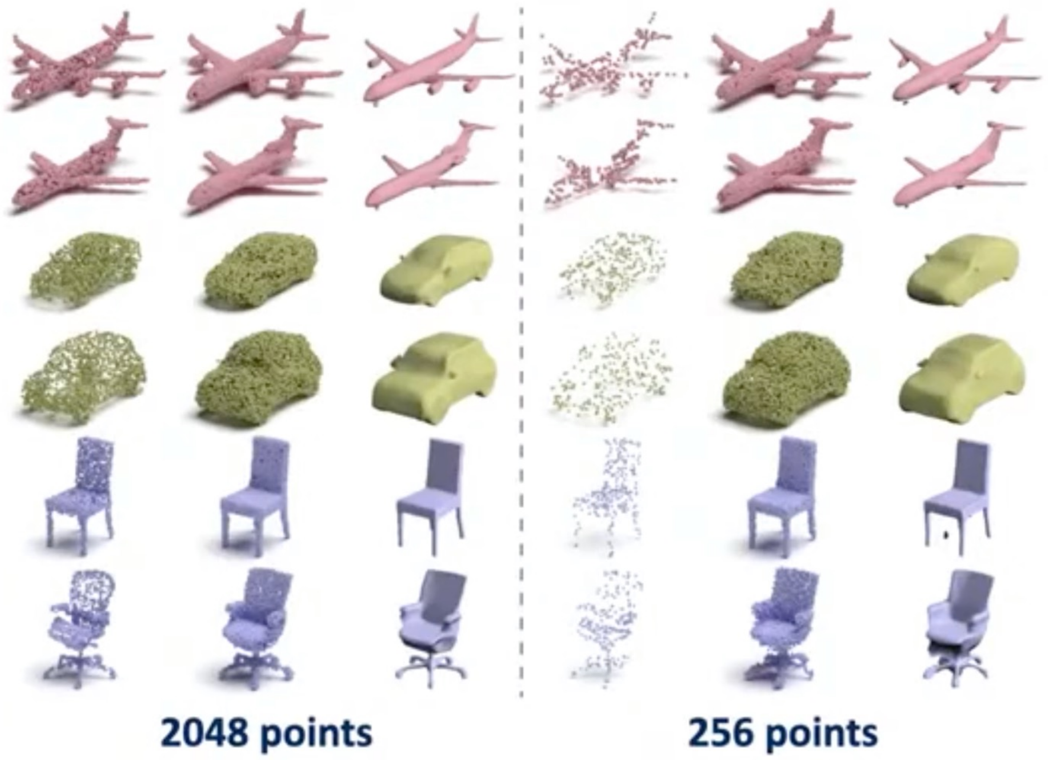
Implicit Surface Generation

- Key insight: Surface is the region of max likelihood
- Zero Gradient condition on surface
- For small variance, approximates distance field around surface



Experimental Results

Qualitative Results



Experimental Results

ShapeNet Auto-Encoding

Dataset	Metric	l-GAN [1]		AtlasNet [23]		PF [60]	Ours	Oracle
		CD	EMD	Sphere	Patches			
MNIST-CP	CD	8.204	–	7.274	4.926	17.894	**2.669**	1.012
	EMD	40.610	–	19.920	15.970	8.705	**7.341**	4.875
Airplane	CD	1.020	1.196	1.002	0.969	1.208	**0.96**	0.837
	EMD	4.089	2.577	2.672	2.612	2.757	**2.562**	2.062
Chair	CD	9.279	11.21	6.564	6.693	10.120	**5.599**	3.201
	EMD	8.235	6.053	5.790	5.509	6.434	**4.917**	3.297
Car	CD	5.802	6.486	5.392	5.441	6.531	**5.328**	3.904
	EMD	5.790	4.780	4.587	4.570	5.138	**4.409**	3.251
ShapeNet	CD	7.120	8.850	5.301	**5.121**	7.551	5.154	3.031
	EMD	7.950	5.260	5.553	5.493	5.176	**4.603**	3.103

Takeaways

- Best performance in almost all cases
- Operates at theoretical optimum

Experimental Results

Point Cloud Upsampling

 # Runtime points

 # Training points

N	CD					EMD				
	2048	1024	512	256	128	2048	1024	512	256	128
$10K$	0.993	1.057	0.999	1.136	1.688	2.463	2.608	2.589	3.042	3.715
$3K$	1.080	1.059	1.003	1.142	1.753	2.533	2.586	2.557	2.997	3.878
$1K$	–	–	1.021	1.149	1.691	–	–	2.565	2.943	3.633

Takeaways

- Able to operate on extremely sparse data
- Able to train using sparse data without loosing too much performance

Experimental Results

Generation Results

Discussion of results

- Performance suggests accurate & uniform distribution
  - Qualitatively & Quantitatively
  - Close to best possible performance
- Robust to sparsity in training and runtime
- Outperforms AtlasNet on ShapeNet data (close to optimal)
- Shape Generation results look believable
  - Comparable results to PointFlow but half the training time

Critique / Limitations / Open Issues

- No runtime analysis
- Rotation and Translation invariance?
- How does this work in a real scene?
- How does it deal with noise?
- How robust is it to real data?
- “Limited to reconstruction of single objects” [2]
- “struggle to generalize to scenes outside of the training distribution” [2]

Contributions (Recap)

- Novel generative model of objects given their point clouds
- Useful because of robustness to sparsity and occlusion
- Prior works are computationally expensive and unstable
- Key insights
  - Can generate implicit surfaces without generating meshes
  - Can generate new objects by interpolating latent space
- State of the art performance on auto-encoding and generation of point clouds