Neural Point Cloud Diffusion for Disentangled 3D Shape and Appearance Generation

Philipp Schröppel Christopher Wewer Jan Eric Lenssen Eddy Ilg Thomas Brox
schroepp@cs.uni-freiburg.de

MAX PLANCK INSTITUTE FOR INFORMATICS
Computer Vision and Machine Perception Lab

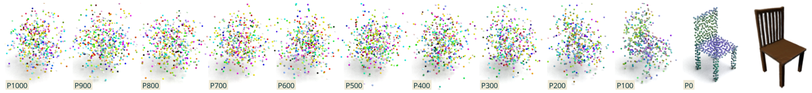
CVPR SEATTLE, WA JUNE 17-21, 2024

Overview

Diffusion for 3D Shape and Appearance Generation:

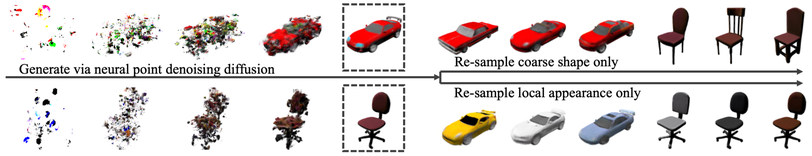
- **Goal:** learn distribution → **generate 3d shape and appearance** of new objects
- **Previous works use different 3D representations**: latent codes (Functa), triplanes (SSD-NeRF), voxel grids (DiffRF)

This work → apply diffusion to a neural point cloud 3D representation:

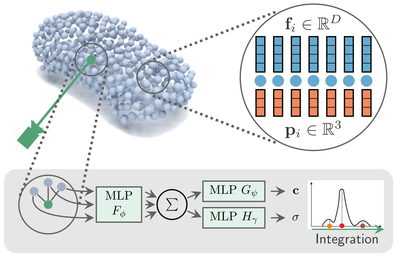


- Points have: **3D position + higher-dimensional feature**
- Can be **rendered to images with Point-NeRF**
- Figure shows diffusion on neural point cloud + Point-NeRF rendering of the final generated neural point cloud

Point positions represent shape, features represent appearance → **This enables disentangled generation:**



Neural Point Cloud Representation and Point-NeRF Rendering



Represent 3D shape and appearance of object by a neural point cloud:

$$\mathcal{P} = (\mathbf{P}, \mathbf{F})$$

→ M neural points with ■ positions $\mathbf{P} \in \mathbb{R}^{M\times 3}$ and ■ features $\mathbf{F} \in \mathbb{R}^{M\times D}$

Can be rendered to images with a Point-NeRF:
Like NeRF, but compute colors $\mathbf{c}$ and densities σ from neighbouring neural points using MLPs F_ϕ and H_γ and G_ψ

Point-NeRF Autodecoder Optimization (on N objects from same category)

Given:

- **Dataset of N objects** from same category
- **Object**: consists of a neural point cloud $\mathcal{P}_j = (\mathbf{P}_j, \mathbf{F}_j)$ and K images with camera parameters

Optimization objective:

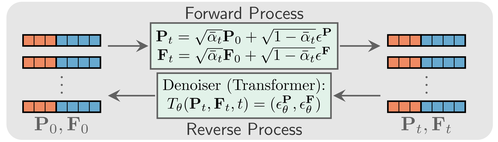
- **Optimize point features $\mathbf{F}$ and MLP parameters ϕ, ψ, γ**
- Objective: **Minimize image reconstruction error for all images of all objects**

Initialization and Regularization:

- **Crucial to avoid many-to-one mappings between features and renderings**
- We use: **Zero initialization**, **TV regularization**, **KL regularization**

Diffusion Model Training

- **Input:** dataset of optimized neural point clouds
- Diffusion model is **trained jointly on point positions and appearance features**



Disentangled Generation (here described for appearance-only generation)

Appearance-only generation with given point positions $\mathbf{P}_0$ works by "masking" point positions and generating only features:

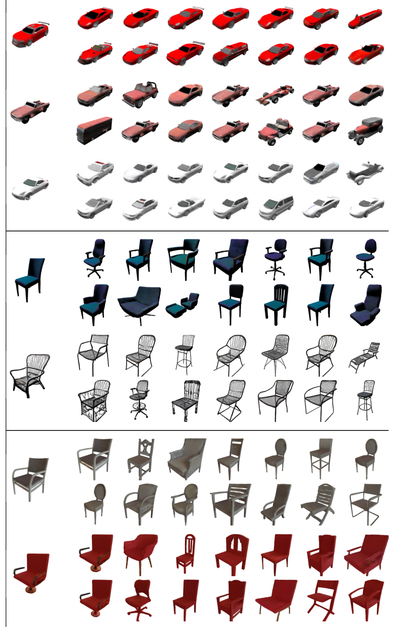
1: $\epsilon^{\mathbf{P}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ $\mathbf{P}_T = \sqrt{\bar{\alpha}_T}\mathbf{P}_0 + \sqrt{1-\bar{\alpha}_T}\epsilon^{\mathbf{P}}$ $\mathbf{F}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $(\epsilon^{\mathbf{P}}_\theta, \epsilon^{\mathbf{F}}_\theta) = T_\theta((\mathbf{P}_t, \mathbf{F}_t), t)$ ▷ **Estimate noise**
4: $\mathbf{P}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\mathbf{P}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\epsilon^{\mathbf{P}}$ ▷ **Compute point positions via forward process**
5: $\mathbf{F}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{F}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon^{\mathbf{F}}_\theta\right) + \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\epsilon$ $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ **Update point features via reverse process**
6: **end for**

→ **Intuitively like masked image inpainting, similar to RePaint**

Disentangled Generation Results (on ShapeNet SRN Cars, SRN Chairs, PhotoShape Chairs)

Appearance-only generation:

Shape-only generation:



Evaluation of Unconditional Generation Quality

Comparison to disentanglement-capable approaches:

Model	ShapeNet SRN Cars FID↓	ShapeNet SRN Cars KID/10^{-3}↓	ShapeNet SRN Chairs FID↓	ShapeNet SRN Chairs KID/10^{-3}↓	PhotoShape Chairs FID↓	PhotoShape Chairs KID/10^{-3}↓
GRAF	40.95	19.15	37.19	17.85	34.49	17.13
D3D	62.34	41.60	45.73	24.33	59.80	36.07
NPCD (Ours)	**28.38**	**17.62**	**9.87**	**3.62**	**14.45**	**5.40**

→ our approach outperforms previous approaches

Comparison to 3D diffusion approaches:

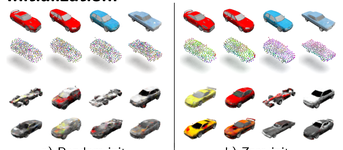
Model	PhotoShape Chairs FID↓	PhotoShape Chairs KID/10^{-3}↓
DiffRF	15.95	7.93
NPCD (Ours)	**14.45**	**5.40**

Model	SRN Cars FID↓	SRN Cars KID/10^{-3}↓
Functa	80.3	-
SSDNeRF	**11.08**	**3.47**
NPCD (Ours)	28.38	17.62

→ Our approach performs better than Functa and DiffRF and worse than SSDNeRF, but enables disentangled generation

Analyses on Initialization and Regularization

Initialization:



Regularization:

Many-to-one mappings:

→ *how far are features that represent the same appearance away from each other?*
→ We measure this by computing per-point mean cosine similarities between optimized neural point features of 10 training examples for 100 different seeds:

Init.	Reg.	Cosine sim.
Rand.	✗	0.0306
Zero	✗	0.7695
Zero	TV	0.9355
Zero	KL	0.9480
Zero	TV,KL	0.9470

→ zero initialization and regularizations reduce many-to-one mappings (as shown by Figures on the left and Table above) and improve generation quality (as shown by Table below)

Setting	Init.	Reg.	ShapeNet SRN Cars PSNR↑	FIDrec↓	KIDrec↓	FID↓	KID↓	ShapeNet SRN Chairs PSNR↑	FIDrec↓	KIDrec↓	FID↓	KID↓
a) Initialization												
Random initialization	Rand.	✗	29.24	37.24	25.37	125.51	97.82	-	-	-	-	-
Zero initialization	Zero	✗	31.32	18.96	11.17	53.55	35.37	34.91	10.37	4.85	39.19	23.64
b) Regularization												
No regularization	Zero	✗	31.32	18.96	11.17	53.55	35.37	34.91	10.37	4.85	39.19	23.64
TV regularization	Zero	TV	29.72	22.42	13.71	45.90	28.70	32.38	14.10	6.70	32.87	17.49
KL regularization	Zero	KL	30.02	24.93	15.60	55.01	35.86	34.20	8.37	3.17	18.13	8.17
TV+KL regularization	Zero	TV,KL	29.70	26.12	16.44	43.92	26.53	33.62	8.58	3.34	17.17	7.44
c) Model size												
40M parameters	Zero	TV,KL	29.70	26.12	16.44	43.92	26.53	33.62	8.58	3.34	17.17	7.44
300M parameters	Zero	TV,KL	29.70	26.12	16.44	28.38	17.62	33.62	8.58	3.34	9.87	3.62

Summary

→ We apply **diffusion for 3D shape and appearance** generation to a **neural point cloud representation**
→ **This enables disentangled generation**, as point positions represent coarse shape and features represent appearance
→ We provide analyses and **insights into many-to-one mappings in auto-decoded latent spaces** and how to tame them for the application of diffusion models
→ Project page and code: https://neural-point-cloud-diffusion.github.io/

Acknowledgements

The research leading to these results is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under the projects 401269959 and 417962828.