Final Report: 3D Shape Generation and Completion through Point-Voxel Diffusion

Pietro Caforio
Technical University of Munich
pietro.caforio@tum.de

Simon Benedict
Technical University of Munich
go72wov@mytum.de

Tobias Palzer
Technical University of Munich
tobias.palzer@tum.de

Abstract

This project report describes enhancements to the Point-Voxel Diffusion (PVD) model for 3D shape generation by introducing three key modifications: (1) enabling shape completion using RGB images instead of depth data, (2) incorporating text-conditional generation, and (3) allowing text-conditional part editing. These extensions make PVD more versatile and improve its applicability across different datasets and user interactions.

1. Introduction

Generative modeling of 3D shapes has significant potential in vision, graphics, and content creation. To maximize this potential, two critical aspects must be addressed: ensuring the model produces realistic, high-quality shapes and providing users with the ability to manipulate and refine these shapes interactively.

The first aspect is the generation of shapes that are not only realistic but also free from artifacts, ensuring the final output is of the highest quality. The second aspect focuses on user interaction. A powerful generative model should allow users to manipulate and refine the generated shapes according to their needs. For instance, users might want to adjust details, explore variations, or refine a coarse input to achieve a polished result. This flexibility enables users to tailor the output to specific requirements.

Point-Voxel Diffusion (PVD) [16] is a novel approach for probabilistic generative modeling of 3D shapes. Unlike most existing models that deterministically translate a latent vector to a shape, PVD uses a unified, probabilistic formulation for both unconditional shape generation and conditional, multi-modal shape completion. It integrates denoising diffusion models with a hybrid point-voxel representation of 3D shapes. The model operates by progressively reversing a diffusion process that transforms observed point cloud data back to Gaussian noise. This process is trained by optimizing a variational lower bound to the (conditional) likelihood function.

The model has two modes of operation. First, it is able to unconditionally generate a shape by denoising a point cloud sampled from Gaussian Noise. This allows the model to generate point clouds of a certain category (e.g. car, airplane). The second mode of operation is completion. For this, the model takes a partial point cloud and a depth map as input. The depth map is used to generate additional points to the partial point cloud by backprojection. The model then uses the same 3DCNN architecture as in the unconditional generation to complete the shape. It thus learns to perform generation conditioned on the partial point cloud of an object.

We propose to expand PVD with the following methods:

1. We enable completion based on RGB image instead of depth information. This makes the model applicable to a broader range of datasets, since depth data is not always available.
2. We enable text conditional generation. This allows for a guidance of the model to generate shapes conditioned on user input.
3. We enable text conditional part editing. This allows to select parts of the object to remove and regenerate conditioned on user input.

We show that our work allows for wider application of PVD as well as text conditional generation and completion. Our implementation can be found here: https://github.com/PietroCaforio/PVD-mod .

2. Related Work

Prior works in point cloud generation include auto-encoding, single-view reconstruction, and adversarial generation. Notable examples are PointFlow [15] and Learning Gradient Fields [2], which adopt probabilistic approaches. Traditional methods often use heuristic loss functions like Chamfer Distance (CD) and Earth Mover's Distance (EMD).

Voxel-based models, such as Generative and Discriminative Voxel Modeling [1], and point cloud-based models, like PointNet [12] and PointCNN [8], face limitations like high memory requirements and permutation invariance issues. The Point-Voxel CNN [10] addresses these by leveraging spatial correlations in point cloud data. Point-voxel CNN [9] shows how point-voxels can be leveraged for the efficient employment of 3DCNNs.

As of late, diffusion based generative models have shown major advances in the 2D space [6, 13] as well as on 3D point clouds [2, 11]. A well performing 3D diffusion model is LION [14], which performs diffusion of 3D objects in the latent space.

3. Data

The original model was trained using the Shapenet dataset [3], which contains over 50,000 objects across 55 categories. For the first modification, we continued to use the original dataset, as it contains images as well as their camera parameters needed for shape completion. It does not, however contain textual descriptions, required for conditioning the model, thus leaving us with two options: Either manually annotating (a part of) the Shapenet dataset, or use an already annotated one.

Objaverse 1.0 [4] contains over 800,000 objects, which are annotated with categories, textual descriptions, as well as additional tags. It should be noted however, that the quality of these annotations varies, as not every object contains tags, or a useful description. The frequency distribution can be seen in Fig. 1. For this purpose, we filtered objaverse, by initially only downloading objects with more than 4 tags. We also included the ability to filter out problematic authors, as we found one author responsible for many objects unsuitable for our purpose in the "chair" class.

Lastly, we further filtered the dataset to remove model outliers. The tags downloaded are created by the community, and can often describe vastly different objects, which is problematic when generating objects of only one class.

First, we download an initial dataset. We then turn the object into a pointcloud of size 6144 points by sampling the points randomly using barycentric coordinates of the triangle faces. Next, for each object, its average Hausdorff distance with 50 random samples is calculated. The 15% of objects with the highest average distance are then filtered out, on the assumption that they are most likely to contain ill-formed objects inconsistent with the rest of the data. Figure 2 shows examples of objects with high average distance.

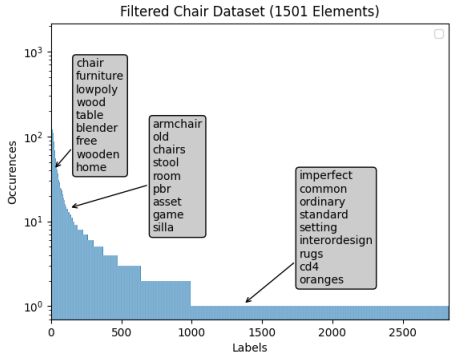


Figure 1. Distribution of tags of chair objects in objaverse: The distribution roughly follows an exponential distribution, with a select few having a high amount of occurrences, whereas the vast majority of tags only occurs once.

Figure 2. Objaverse outliers: These samples were taken from the objaverse set by downloading objects tagged as "chair". Each one contains additional elements, which is not desired when simply generating a single chair. The algorithm therefore recognized them by their high average distance from the other instances.

4. Method

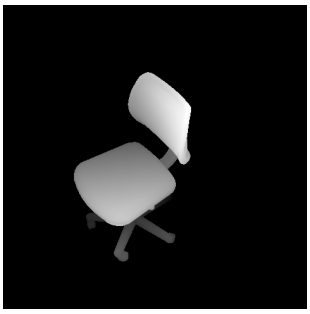
Our work builds upon the Point-Voxel Diffusion (PVD) [16] model, aiming to achieve shape completion using RGB images rather than depth data as well as integrating text-conditional generation and enabling text-conditional part editing within the model.

4.1. RGB instead of depth Data

The approach used by PVD for handling depth images in the completion task leverages the utilities of the Open3D library to sample a partial point cloud from the depth image, which is then used for the completion process. To extend the model's capability to handle RGB images for completion,

(a) RGB input Image

(b) Depth prediction



(c) Depth-map ground truth

Figure 3. SegFormer depth-map estimation example

we modified PVD's dataloader to integrate a SegFormer-based monocular depth estimation model [7].

This model infers a depth map from a single-view RGB image, which is then used with Open3D's PointCloud generation functionality to sample points .

This adaptation enables the model to perform completions using single-view RGB images with minimal performance loss, despite the sampled point cloud being limited to a single view, potentially offering a weaker prior for reconstruction.

Figure 4 shows an example completion prediction from the model.

4.2. Text-Conditional Generation

The primary goal here is to enable the model to incorporate text conditioning during the denoising process of Diffusion, allowing it to generate data samples that align with the meaning of a given text prompt. Typically, this kind of task is achieved by using techniques such as triplet loss to create an embedding space where the semantic features of text and point clouds are closely aligned in terms of similarity.

In our work, due to limitations in time and resources, we sought to achieve this correlation without implementing complex loss functions or making modifications that would have highly altered the functioning of the base architecture.

Our approach leverages the time-step embedding component of the PVD model's architecture to incorporate the semantics of the text prompt, aiming to allow the model to implicitly learn the relationship between the text and the point cloud denoising process.

We introduced a BERT [5] model that, when given a tokenized text prompt, generates an embedding vector at runtime. This vector is then concatenated with the time-step embedding used in the PVD diffusion process. To facilitate this concatenation, the BERT model's embedding is first compressed into a latent feature using an MLP (Multi-Layer Perceptron).

4.3. Text-Conditional Part Editing

With the text-conditional generative model in place, we experimented to see if the text conditioning could be transferred to a text-conditional reconstruction task. The concept here is to provide the model with a partial point cloud (e.g., a chair missing its backrest) along with a text prompt (e.g., "fancy chair backrest") and use the text-conditioned model to perform reverse diffusion on the partial chair. The text prompt embedding would then guide the denoising process, influencing the reconstruction of the missing part in a way that aligns semantically with the text (e.g., reconstructing the missing backrest as a fancy backrest).

5. Results

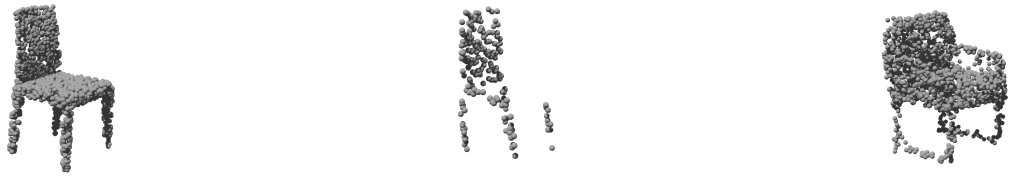
We conducted our experiments on a NVIDIA RTX 3090 GPU. We used the training script provided in the original repository. The provided losses are also calculated with the loss modules of PVD.

5.1. RGB instead of Depth Data

The sampling of the additional points to the partial point cloud on the basis of a depth prediction from RGB yields almost identical results. After 1800 epochs, the default model has a loss of 0.1949, the RGB-only model achieves a loss of 0.2275. A prediction of a depth map based on RGB data can be seen in figure 3. We note, that we were only able to train both of these models for 1800 epochs because of GPU time limitations.

5.2. Text-Conditional Generation

As can be seen in figure 5, the model can be conditioned on the textual input of the objaverse description to generate multiple varieties "chair"-class instances. We can see, the model shapes correspond to the described shapes. Even more abstract prompts like "autoprogettazion", a term used by italian designers", are used by the model as additional guidance, in this case to make the chair more fancy. However, it is not possible to distinguish the material (e.g.



(a) Ground Truth (b) Partial sampled Point Cloud (c) Model's Completion

Figure 4. Completion example

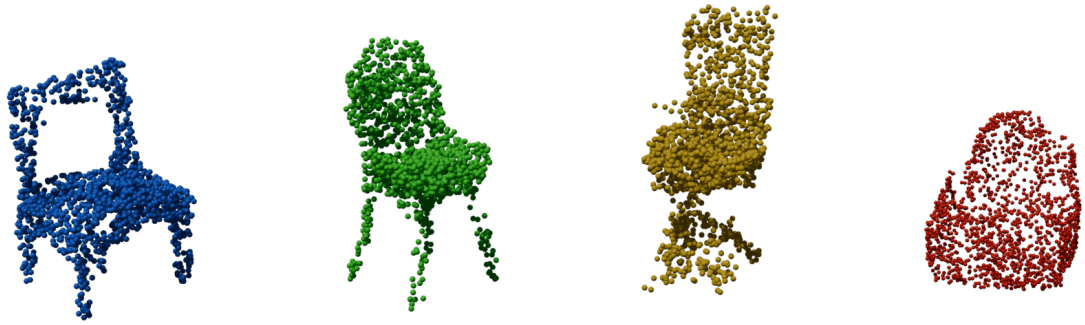


Figure 5. Text-conditioned chairs generated through the following text prompts (going from left to right): "wooden cafe classic dirt furniture old", "structure scan enzo mari autoprogettazione", "school desk furniture lowpoly", "model animation realistic cusion"

"wooden") due to the lack of textures as well as the limited detail of pointclouds. We note, that we were only able to train this model for 1200 epochs, as our computational resources were limited.

5.3. Text-Conditional Part editing

The model hasn't demonstrated significant transfer to the part-editing task when framed as reconstructing the missing portion of the input point cloud. Specifically, we observed that the model tends to apply reverse diffusion to the entire partial input point cloud, deforming it into an entirely new object (e.g. chair). This issue can be addressed by fixing the given points of the input point cloud.

Additionally, another problem arose from our initial use of a simple automatic approach to create the part-missing point clouds by removing a cube of points. Instead, the removal process should be guided by semantic segmentation of the object.

6. Conclusion

We have proposed several modifications to the Point-Voxel Diffusion model to enable a broader usability, by making it usable with RGB-data, and also enabling text-conditional generation. We note, that for better analysis of the method, an experiment suite with a higher amount of resources should be conducted, as our modified models could not be trained to a final state. For the part-editing task, we experimented with a baseline approach that didn't yield significant

results. However, we identified some potential refinements that could lead to successful transfer of the model for part-editing.

References

[1] Andrew Brock, Theodore Lim, JM Ritchie, and Nick Weston. Generative and discriminative voxel modeling with convolutional neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 409–425, 2016. 2

[2] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 364–381. Springer, 2020. 2

[3] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository, 2015. 2

[4] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3d objects, 2023. 2

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. 3

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pages 6840–6851. Curran Associates, Inc., 2020. 2

[7] Doyeon Kim, Woonghyun Ga, Pyunghwan Ahn, Donggyu Joo, Sehwan Chun, and Junmo Kim. Global-local path networks for monocular depth estimation with vertical cutdepth. *CoRR*, abs/2201.07436, 2022. 3

[8] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on $\mathcal{X}$-transformed points. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 820–830, 2018. 2

[9] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 2

[10] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 149–159, 2019. 2

[11] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. *arXiv preprint arXiv:2103.01458*, 2021. 2

[12] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660, 2017. 2

[13] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022. 2

[14] Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, Karsten Kreis, et al. Lion: Latent point diffusion models for 3d shape generation. *Advances in Neural Information Processing Systems*, 35:10021–10039, 2022. 2

[15] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4530–4539, 2019. 2

[16] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion, 2021. 1, 2