Generating point clouds from multiple views with Point-E
(145858) Signal, Image & Video - Project report

Diego Calanzone Riccardo Tedoldi Zeno Sambugaro
University of Trento
Trento, Italy
{diego.calanzone, riccardo.tedoldi}@studenti.unitn.it
zeno.sambugaro@unitn.it

1. Introduction

In deep learning literature, the problem of 3D reconstruction from images is formulated under different perspectives: with voxel grids, object volumes are estimated in a discrete 3d space [17]; with continuous space representations [10], the volume is determined by a decision boundary in the 3d space; with neural radiance fields [5], MLPs are optimized to learn a volumetric render for a specific object, with views as ground truth. Despite [5] overcomes difficulties in optimization for methods such as [17], [10], multiple GPUs are still required to learn such rendering functions, which mostly overfit a single scene or object.

Recently, Nichol et al. [4] introduced diffusion [9] to learn 3D point clouds: a generative model learns 3D priors in a distribution of objects and given a single-view image, it is possible to sample a cloud in 1-2 minutes with a single GPU.

We further develop this line of work with two techniques to condition a diffusion model on multiple views; we conduct experiments with a pipeline to extend text-to-3d with multiple synthetic views; finally, we attempt reconstruct 3D point clouds from photos "in the wild".

1.1. Diffusion models

Generative models based on diffusion [9] approximate a data distribution through progressive noising $x_{1 \rightarrow t}$ of a signal x_0 (forward process) and denoising (backward process) with a neural network ϵ_θ, such that $\hat{x_0} = x_t - \epsilon_\theta(x_t, t)$. Conditional sampling is possible by denoising a random signal with images, text or any other token sequence provided to ϵ_θ.

1.2. 3D Point cloud generation with Point-E

(Figure 1) Given a random cloud $x_0 \in \mathbb{R}^{K \times 6}$ of K points (x, y, z, r, g, b), [4] apply diffusion with a transformer as denoising network ϵ_θ. Once $x_t \in \mathbb{R}^{K \times 6}$ is computed with the forward process, ϵ_θ predicts the added noise

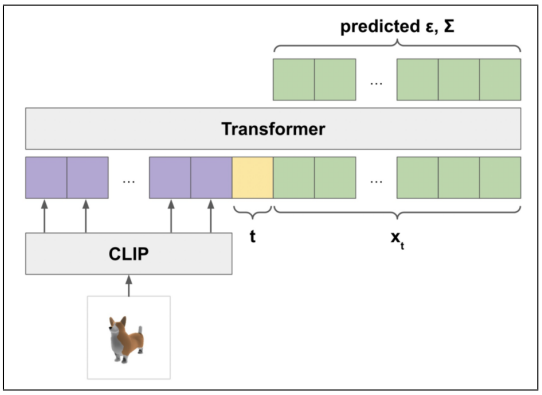

Figure 1. A denoising transformer used in [4] for point cloud diffusion.

ϵ from the concatenation of x_t, t and a conditioning view $v \in \mathbb{R}^{256 \times D'}$, encoded as patches with ViT-L/14 CLIP from [3]. All the input tokens are linearly projected to a standard embedding dimension D.

1.3. Contributions

A single view of an object may not contain enough information for sound 3d reconstruction: e.g. from a top view, it is not possible to accurately estimate the height of an object; reconstruction of occluded parts can introduce blurring or artifacts. We introduce multi-view conditioning for Point-E [4] with two possible methods:

- **Patch concatenation**: CLIP embeddings of multiple views $v_1, ..., v_n \in \mathbb{R}^{256 \times D}$ are horizontally stacked, resulting in a single token sequence o length $N \times 256 \times D$. The denoiser ϵ_θ attends to all views at each step.
- **Stochastic conditioning**: as in [7], a random view $v_i, i \sim U(1, n)$ is drawn at each denoising step. The cost of computation is unchanged as one view at a time is used, however the price to pay is stochasticity yield-

ing different combinations of conditioning views and thus varying outcomes. Increasing the number of denoising steps allows view probabilities to converge.

2. Experiments

2.1. Text-to-3d with Stable Diffusion

We firstly generate a synthetic image from text with Stable Diffusion 2 [16]: a pre-trained, freely accessible text-to-image model based on Latent Diffusion [13]. To remove shadows, background and undesired objects from the generated image, we also provide as input a starting 256×256 image u, which consists in a white background with a central black square to inpaint the generated content. According to the pipeline in Figure 2 (a), the synthetic image is then provided as input to Point-E to obtain a $K \times 6$ point cloud. Point-E includes two diffusion modules: *base*, to generate a $K_1 = 1024$ cloud from visual semantics; *upscale*, which given K_1 points and the conditioning view, it generates the remaining $K_2 = 4096 - K_1$ points for 4K resolution. Results from appendix 4.1 show that Point-E yields clouds consistent with the provided view from fixed perspectives, while it can present color distortions, wrong depths or spots without points for occluded parts.

2.2. Multi-view with Patch Concatenation

We expand the conditioning approach described in paragraph 1.2: multiple views are encoded with the pre-trained image encoder from OpenAI CLIP [3], forming a vector of input tokens $v_1, ..., v_n \in \mathbb{R}^{256xD'}$, with D' a predefined embedding length for CLIP. The sequence of tokens provided to the denoising transformer is thus: $c = (v_1, ..., v_n, t, x_t)$ with $v_i, t, x_t \in \mathbb{R}^D$, composed by the conditioning view patches, a timestep embedding and the noisy cloud of K points after t noising steps; each token is projected to the dimension D with learnable linear projection W_θ. As the denoising network ϵ_θ attends to multiple views at each step, we notice an increase in time of inference by around $\sim$ *50%* for 4 conditioning views[1]. Results from 4.2 highlight two phenomenons:

- Given views of the same object with slight semantic differences (e.g. a hamburger with different slices of stuffing), such details are combined in the reconstructed point-cloud. Semantic compositionality could be further investigate e.g. to compose new scenes.
- With a single conditioning view, occluded object parts are reconstructed according to object priors learned during training (which do not always match the queried object). With multiple views, the generated point cloud holds semantic consistency with the visual description.

2.3. Multi-view with Stochastic Conditioning

As an alternative to the method described above, inspired by [7] we apply stochastic conditioning: a random view from a given set is used for conditioning at each denoising step. As anticipated, there is no difference in time of inference as one view at a time is used, however more denoising steps may be necessary to ensure that all the views have been uniformly seen. Results in appendix 4.2 show this methodology yields similar or reconstructions wrt. 2.2 in simple subjects with occluded parts, given the same amount of denoising steps.

2.4. Novel view synthesis from generated images

To generate multiple views from a synthetic image, we choose 3d-diffusion from Watson et al. [7]: starting from a single view v and a pair of rotation-translation matrices (R, t), the model generates a new frame of the object for a query pair of matrices (R', t'); the generated view is added to the set of conditioning views to re-iterate the process, for each denoising step a random view is used as conditioning signal. Due to the prohibitive computational cost to train this network on a large set of object priors, we conduct our experiments with pre-trained weights on the Scene Representation Network-Cars dataset provided by [15]. Currently, there is a collaborative effort from open AI communities to train and release a ShapeNet version. Figure 2 (b) shows a scheme of the described pipeline. Despite imperfections in the synthetic views, Point-E can generate a 3d-consistent cloud of the object described in the prompt. We recognize potential limitations in this approach: without a significant overlap between the training sets of respectively the text-to-image and the 3d-diffusion models, the generated views would not contribute significantly to reconstruct the described object.

2.5. Reconstructing 3D from real photos

Shifting from prompt-generated images to real photos, we test a simple pipeline for 3D point cloud generation: to identify the subject of the photo and to isolate it from the background and irrelevant objects, we use a U^{2}Net [19] pre-trained for Salient Object Detection. With respect to models from the baseline, U^{2}Net combines contextual information with local features perceived through receptive fields of multiple sizes; moreover, this architecture allow for near real-time inference on a single GPU[2]. The processed views are provided as input to Point-E with the methodologies described above. As shown in 21, significant gain in

[1] All the experiments have been conducted with a single RTX 3070 Laptop GPU with 8GB of VRAM.

[2] U2-Net (176.3 MB, 30 FPS on GTX 1080Ti GPU) and (lightweight) U2-Net (4.7 MB, 40 FPS)

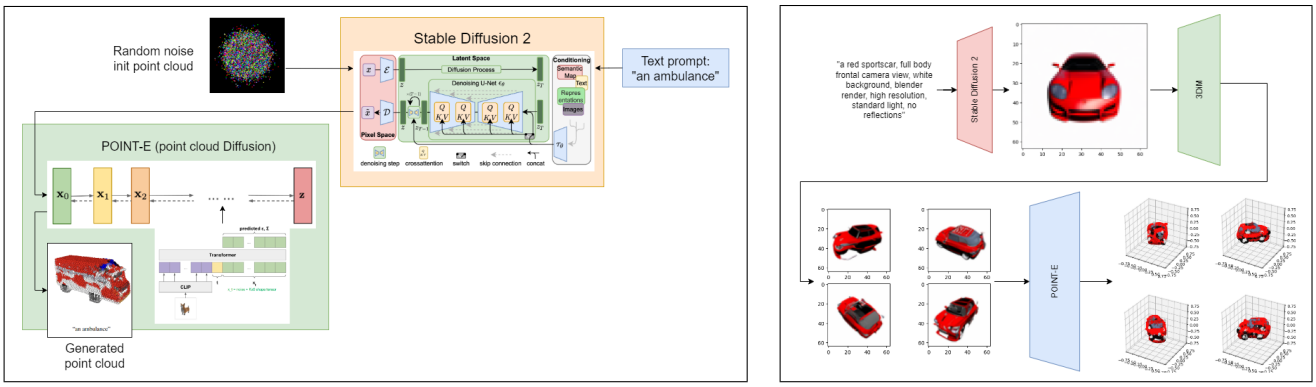

Figure 2. (a) Single-view text-to-3d with Point-E on top of Stable Diffusion 2. (b) Multi-view text-to-3d with with 3DiM in addition, on top of Stable Diffusion 2.

reconstruction quality is obtained with multiple conditioning views, as shape details and proportions are difficult to estimate from e.g. a single, top view.

3. Evaluation

3.1. Dataset

All the generated datasets have been published with code [3]. We considered as sources: ModelNet40 [2] (as included in the work from [4]), ShapeNetV2 and ShapeNet [6]. For each mesh, we stored:

- The *id label* of the object
- A sampled he mesh into a uniform point cloud
- The rendering of the ground truth meshes with different poses that portray the object from different views

In the ModelNet40 and ShapeNetV2 datasets, there were no textures available. In a few situations it may be difficult to reconstruct the 3D object where the shapes in the views are not sharply defined. Therefore, the model can be strongly conditioned by textures in some situations: for objects with missing textures, we computed synthetic colors and patterns.

On ModelNet40 and ShapeNetV2 we added textures with approaches based on spatial position, this allows to better recognize edges and surfaces from different perspectives. We discuss these details in the Appendix 4.3. Most of the meshes found on ShapeNet came with textures. Thus, taking only the one that had textures, in each test batch of ShapeNet we selected for each type of object (guitar, basketball ball,...) 25 random objects of the same category. We iterated this procedure overall in the ShapeNet tests batch.

[3]GitHub: https://github.com/halixness/point-e

The rendered images are without any reflections and the light in the scene is fixed. The dimensions of the multiple views generated from ModelNet40 and ShapeNetV2 are 512x512 whereas 256x256 are the ones from ShapeNet with texture. The number of samples generated from ShapeNet with textures stands at 625, while 95 are the ones produced from ModelNet40 + ShapeNetV2 without textures.

3.2. Metrics

We focused on evaluating the multi-view model with patch concatenation and the one with stochastic conditioning compared to the single-view version with these well-known metrics.

Point cloud divergences: Firstly, we compute the matrix of the pairwise euclidean distance between points in the same point cloud. Next, we determine the divergence through the Wasserstein Distance [12].

Batched P-IS: The acronym stands for Point cloud Inception Score [1] . Based on the inception classifier PointNet++ [11] trained by the authors of Point-e [4] on ModelNet40. Thus, we use the likelihoods provided by this model to compute the inception score [14] over a batch of point clouds generated by our multi-view model. The score is computed as the exponential of the KL divergence.

$$D_{\mathrm{KL}}(P\|Q) = \sum_{x\in\mathcal{X}} P(x)\log\left(\frac{P(x)}{Q(x)}\right)$$

The P-IS is maximized when the distribution of the generated object is close to the distribution of a label and differ from the other labels.

Batched P-FID: P-FID is computed from the authors of the Point-e paper through extracting the feature from the last layer before the final ReLU activation. Given those extracted features they compute the Fréchet distance [8] between the generated distribution and the ground truth distribution. According to the original intuition, we compute it

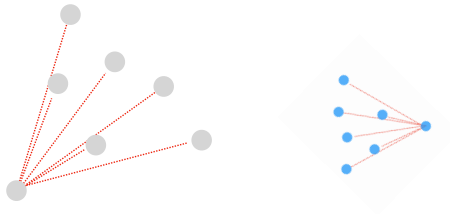

Figure 3. Illustrated in the figure it is shown that even if the point cloud is translated, rotated or scaled in a different size the pattern on the normalized distance distribution still looks exactly the same.

on a random batch of generated samples for sixty iterations. However, our results on P-FID were obtained without extracting features from the last layer.

Batched Chamfer Distance: For each point in clouds S1 and S2, we search the closest point in the other cloud and we perform the sum squared over all the minimum distances between the points [18]. The one below is the mathematical formulation:

$$d_{CD}\left(S_1, S_2\right) = \frac{1}{|S_1|} \sum_{x \in S_1} \min_{y \in S_2} \|x-y\|_2 + \frac{1}{|S_2|} \sum_{y \in S_2} \min_{x \in S_1} \|y-x\|_2$$

This metric has been computed over random batches of generated samples for sixty iterations.

Furthermore, we develop a *novel metric to evaluate the generated point cloud*. Given a ground truth point cloud GT, we measure the distances pairwise between the point in the point cloud. Then we iterate the same process over the generated point cloud. Therefore, we store those measures in the pairwise distance matrix. Those two matrices encode in the columns the details related to the points and where each point is positioned with respect to the other points in the same cloud (Figure 3). Normalizing the distances we switch this set of multidimensional observations into a distribution. When the set of observed distances is similar, the two distributions should overlap. Hence, when the overlap between distributions is pronounced, the point clouds 3Ds also exhibit the same pattern. Through this approach we compare the generated point cloud with ground truth or the multi-view with the single-view.

3.3. Results

We validated the multi-view, patch concatenation variant of Point-E with the standard model conditioned on a single image. We used the same pre-trained weights (Point-E 300M) for both model instances. The metrics have been computed on the evaluations sets with synthetic object textures.

In Figure 4, we report the ***variance on divergence*** in point clouds generated with multi-view with patch concatenation and single-view compared with ground truth on the textureless dataset. The white dots in the box plots represent the outliers. The multi-view model with patch concatenation does not seem to significantly improve convergence towards the ground truth mesh, in a matter of structure. However, the P-IS score improves as multiple views condition the model: this shows better shape consistency with semantic features. We provide an example in the Appendix Figure 19.

The variance on the other metrics discussed can be found in Figure 20 in the Appendix. The results over ***PIS score*** and ***Chamfer distance*** show possible gains with multi-view conditioning: with respect to problematic single-views as shown in 10, better 3d consistency and convergence with the ground truth cloud is achieved. In Figure 22b 22c we can observe that the overlap is better as with multi-view we obtain a more accurate version of the ground truth object.

To validate the model with stochastic conditioning, we generated 650 point clouds with the single-view model and with the model with stochastic conditioning from the views with textures generated by ShapeNet. We can see in Figure 5 and Figure 6 that the point clouds generated with stochastic conditioning are better compared to the once generated from a single view.

3.3.1 Limitations

The experiments have been conducted using the pre-trained 300M model. We would expect equal or better results with the pre-trained 1B model. There is the possibility that the results might be biased: because we tested the model with only a few objects (one hundred) or because of imperfections dictated by differential poses of the imported objects (ex. Figure 19). Moreover, we must consider that these tests were performed on datasets with synthetic textures. Therefore we expect further improvements with rendered views of correctly colored and contoured objects (as seen in Figure 21). We recognize a computational inefficiency in conditioning with a concatenation of image patches, as the time of inference increases. With stochastic conditioning, the inference cost is unchanged, although more denoising steps may be required to attend to all conditioning views sufficiently.

3.4. Conclusion

Text-to-3d generation with single synthetic images leaves room for large improvement. For complex shapes, learning object priors with scarce input data can result in approximate outcomes. For better 3d consistency, we propose two variants of the denoising diffusion process with further experiments to conduct. Metrics based on neural feature ex-

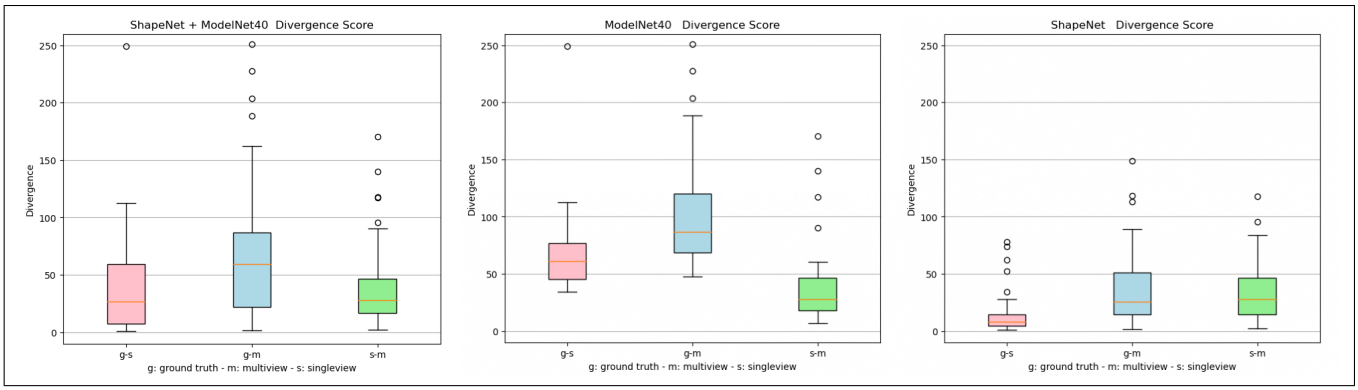

Figure 4. This plot shows the divergence computed based on the Wasserstein distance. Specifically with multi-view with patch concatenation and single-view over the textureless dataset ModelNet and ShapeNetV2..

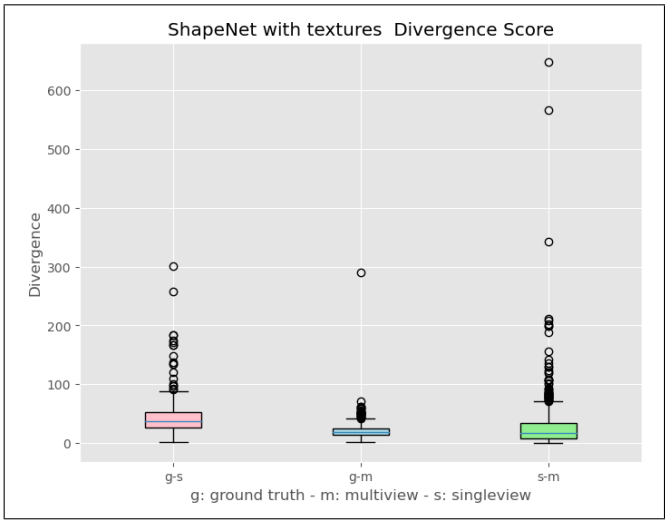

Figure 5. This boxplot shows the variance on divergence computed on the generated clouds on ShapeNet with textures with multi-view with stochastic conditioning and single-view compared with ground truth. The white circles are the outliers.

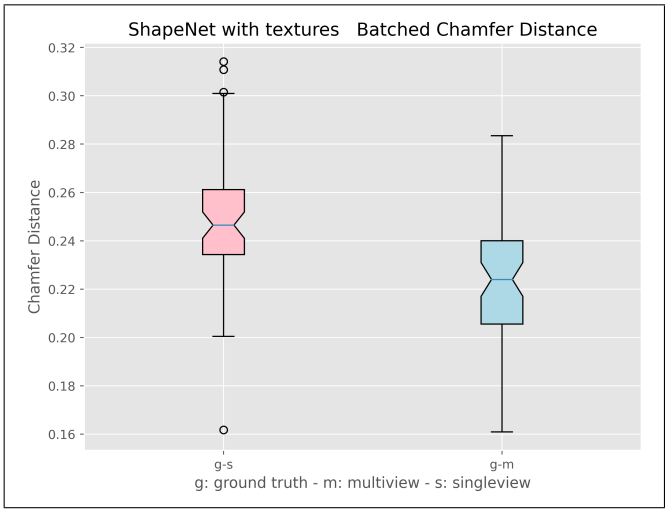

Figure 6. This boxplot shows the variance on the Chamfer distance of the generated point cloud versus the ground truth computed by taking batches of 35 random generated sample and iterating for 60 times.

tractors can result limiting, thus we recognize the necessity for more reliable benchmarks. With patch concatenation, multi-view we observe in specific scenarios equal or better results results compared to the single view version, at the cost of higher computational complexity (potentially solvable with a tradeoff in timesteps/conditioning data with the stochastic conditioning variant (Figure 22).

References

[1] Inception score - Wikipedia — en.wikipedia.org. https://en.wikipedia.org/wiki/Inception_score. [Accessed 14-Feb-2023]. 3

[2] Princeton ModelNet — modelnet.cs.princeton.edu. https://modelnet.cs.princeton.edu. [Accessed 14-Feb-2023]. 3

[3] Chris Hallacy Aditya Ramesh Gabriel Goh Sandhini Agarwal Girish Sastry Amanda Askell Pamela Mishkin Jack Clark Gretchen Krueger Ilya Sutskever Alec Radford, Jong Wook Kim. Learning transferable visual models from natural language supervision. https://arxiv.org/abs/2103.00020, 2021. 1, 2

[4] Prafulla Dhariwal Pamela Mishkin Mark Chen Alex Nichol, Heewoo Jun. Point-e: A system for generating 3d point clouds from complex prompts. https://arxiv.org/abs/2212.08751, 2022. 1, 3

[5] Matthew Tancik Jonathan T. Barron Ravi Ramamoorthi Ren Ng Ben Mildenhall, Pratul P. Srinivasan. Nerf: Representing scenes as neural radiance fields for view synthesis. https://arxiv.org/abs/2003.08934, 2020. 1

[6] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing 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. *CoRR*, abs/1512.03012, 2015. 3

[7] Ricardo Martin-Brualla Jonathan Ho Andrea Tagliasacchi Mohammad Norouzi Daniel Watson, William Chan. Novel view synthesis with diffusion models. https://arxiv.org/abs/2210.04628, 2022. 1, 2

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 3

[9] Niru Maheswaranathan Surya Ganguli Jascha Sohl-Dickstein, Eric A. Weiss. Deep unsupervised learning using nonequilibrium thermodynamics. https://arxiv.org/abs/1503.03585, 2015. 1

[10] Michael Niemeyer Sebastian Nowozin Andreas Geiger Lars Mescheder, Michael Oechsle. Occupancy networks: Learning 3d reconstruction in function space. https://arxiv.org/abs/1812.03828, 2018. 1

[11] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *CoRR*, abs/1706.02413, 2017. 3

[12] Aaditya Ramdas, Nicolas Garcia, and Marco Cuturi. On wasserstein two sample testing and related families of nonparametric tests, 2015. 3

[13] Dominik Lorenz Patrick Esser Björn Ommer Robin Rombach, Andreas Blattmann. High-resolution image synthesis with latent diffusion models. https://arxiv.org/abs/2112.10752, 2021. 2

[14] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. 3

[15] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, 2019. 2

[16] StabilityAI. Stable diffusion 2. https://stability.ai/blog/stable-diffusion-v2-release, 2022. 2

[17] Felix Heide Matthias Nießner Gordon Wetzstein Michael Zollhöfer Vincent Sitzmann, Justus Thies. Deepvoxels: Learning persistent 3d feature embeddings. https://arxiv.org/abs/1812.01024, 2018. 1

[18] Tong Wu, Liang Pan, Junzhe Zhang, Tai Wang, Ziwei Liu, and Dahua Lin. Density-aware chamfer distance as a comprehensive metric for point cloud completion. *CoRR*, abs/2111.12702, 2021. 4

[19] Chenyang Huang Masood Dehghan Osmar R. Zaiane Martin Jagersand Xuebin Qin, Zichen Zhang. U2-net: Going deeper with nested u-structure for salient object detection. https://arxiv.org/abs/2005.09007, 2020. 2

4. Appendix

4.1. Single view text-to-3d with Stable Diffusion 2

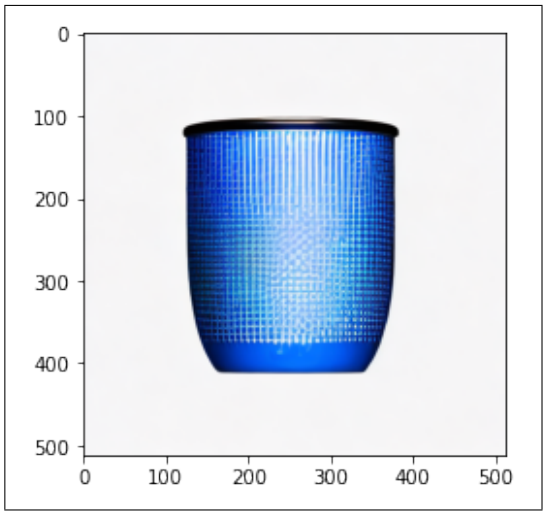

Figure 7. "a blue coffee mug, full body centered camera view, blender render, high resolution, standard light, no reflections" (Stable Diffusion 2)

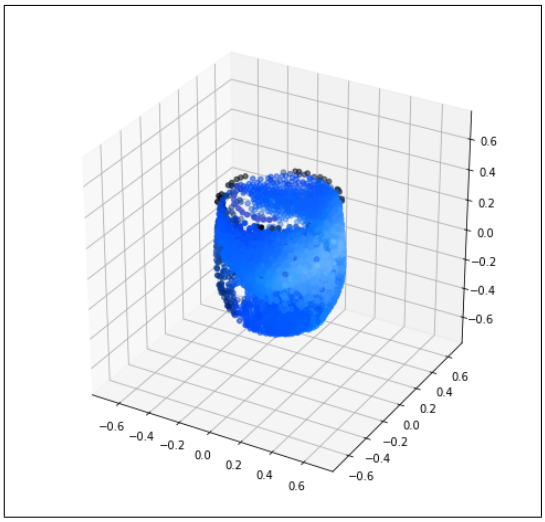

Figure 8. Point cloud generated from 7. The occluded part of the object results in sparse points.

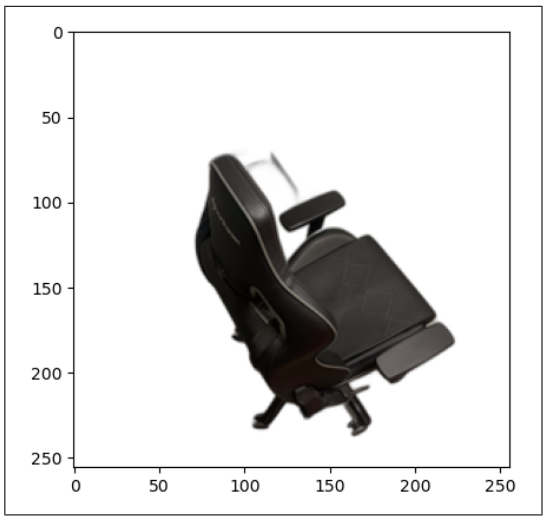

Figure 9. Top view photo of a gaming chair.

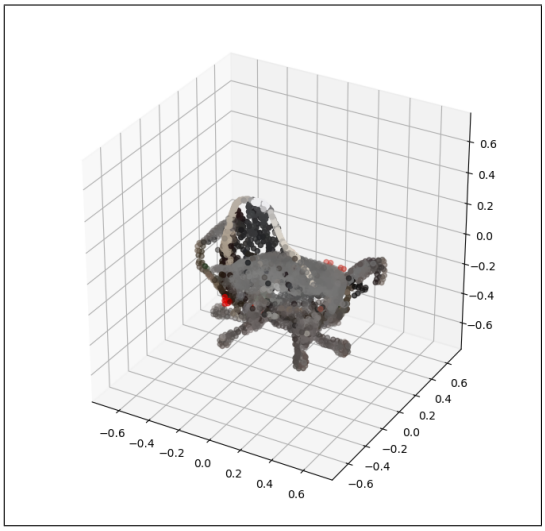

Figure 10. Point cloud generated from 9. From the provided view it is not possible to correctly estimate height and (partial) color of the object.

4.2. Multi view text-to-3d with Stable Diffusion 2

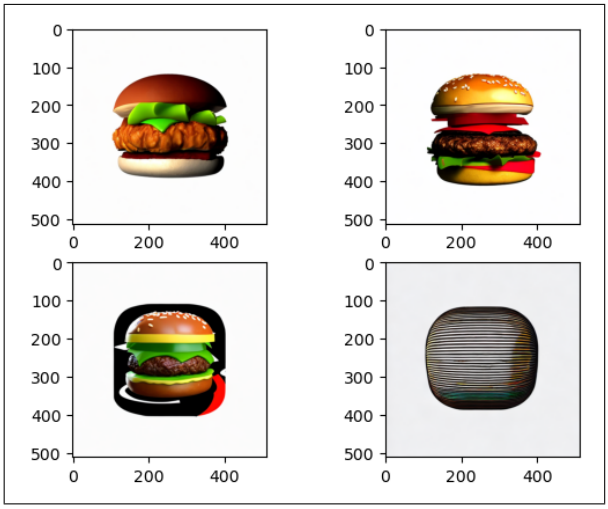

Figure 11. ”a hamburger, full body frontal/bottom/top/left camera view, blender render, high resolution, standard light, no reflections” (Stable Diffusion 2)

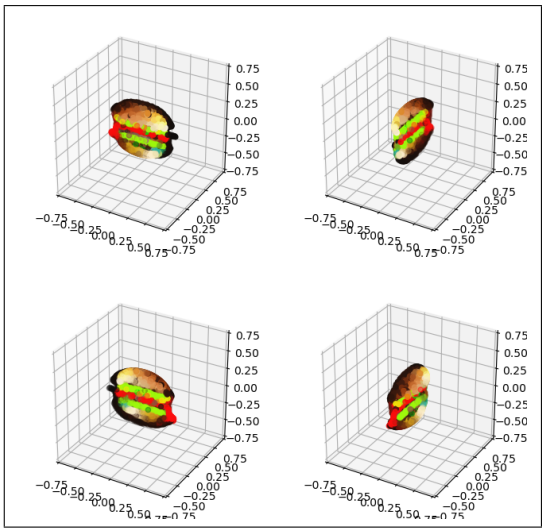

Figure 12. Point cloud generated from 11

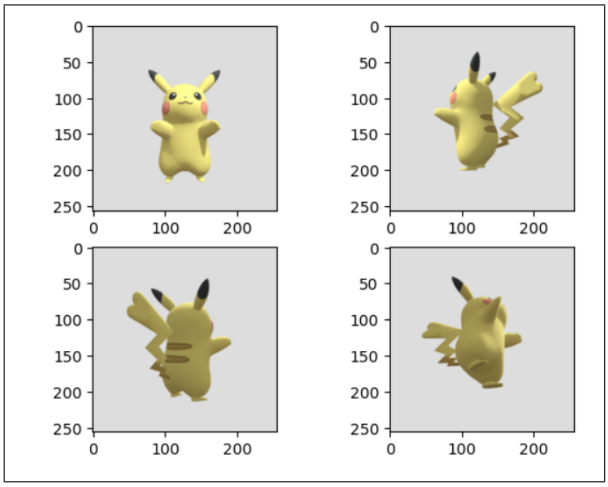

Figure 13. Rendered views of Pikachu (fantasy character) with Blender.

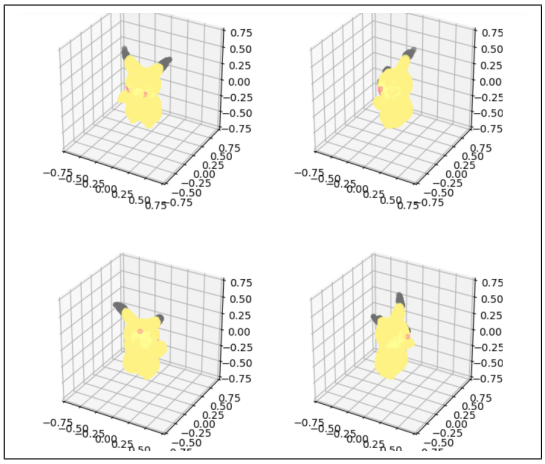

Figure 14. Point cloud generated from the top-left view in 13

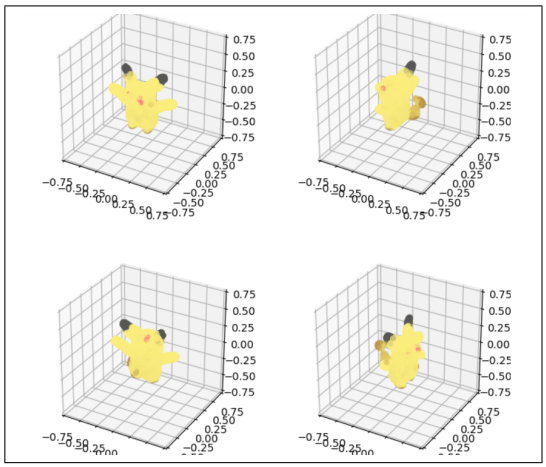

Figure 15. Multi-view, patch concatenation 3D reconstruction from 13.

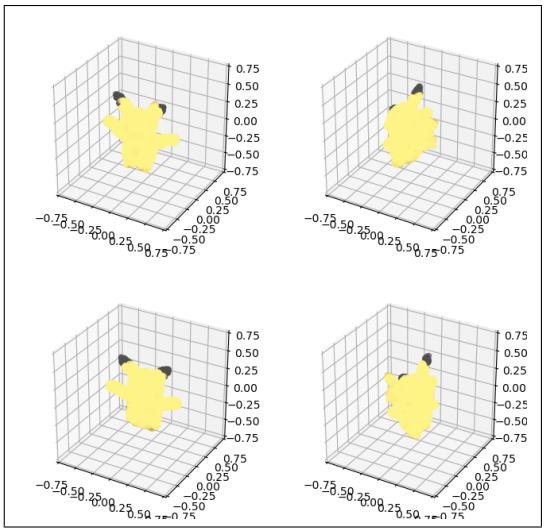

Figure 16. Multi-view, stochastic conditioning 3D reconstruction from 13.

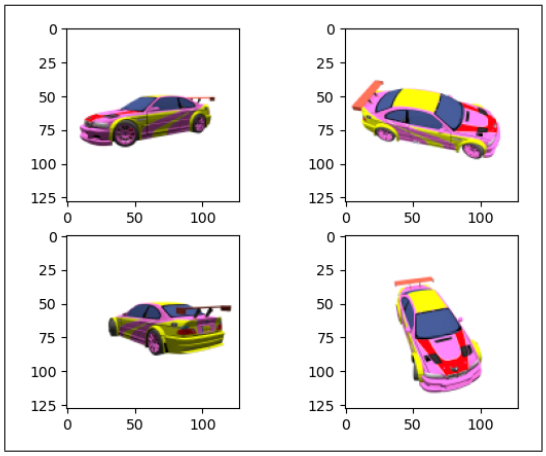

Figure 17. Rendered views from SRNCars with PyTorch3D.

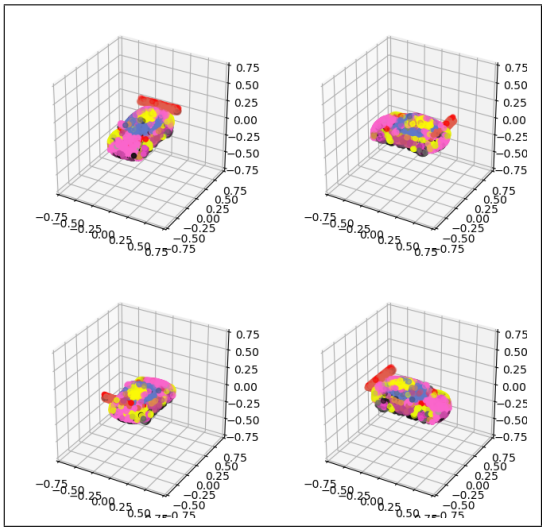

Figure 18. Multi-view, patch concatenation 3D reconstruction from 17.

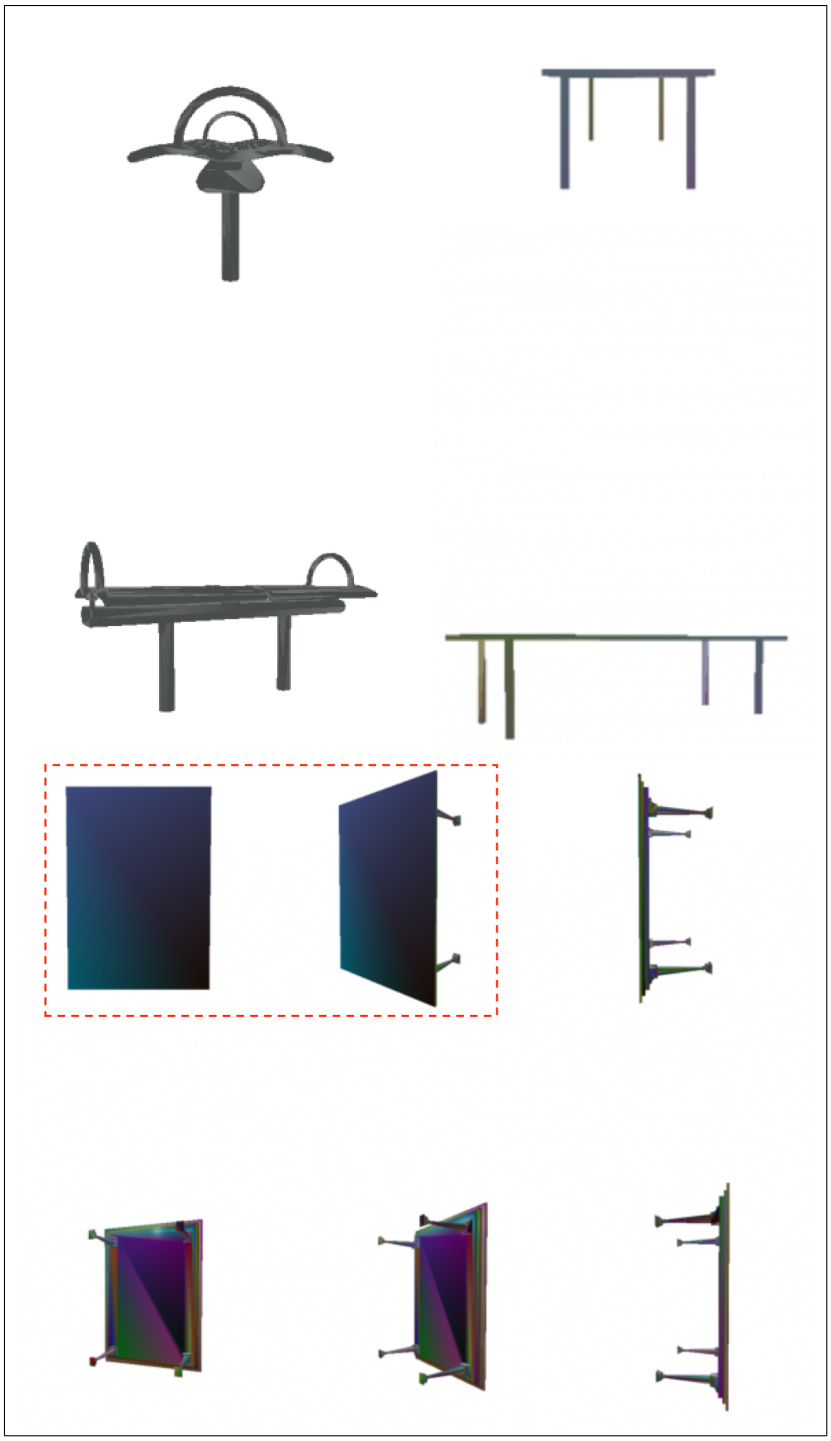

Figure 19. Dissimilar views of the same object.

4.3. Texture tinge

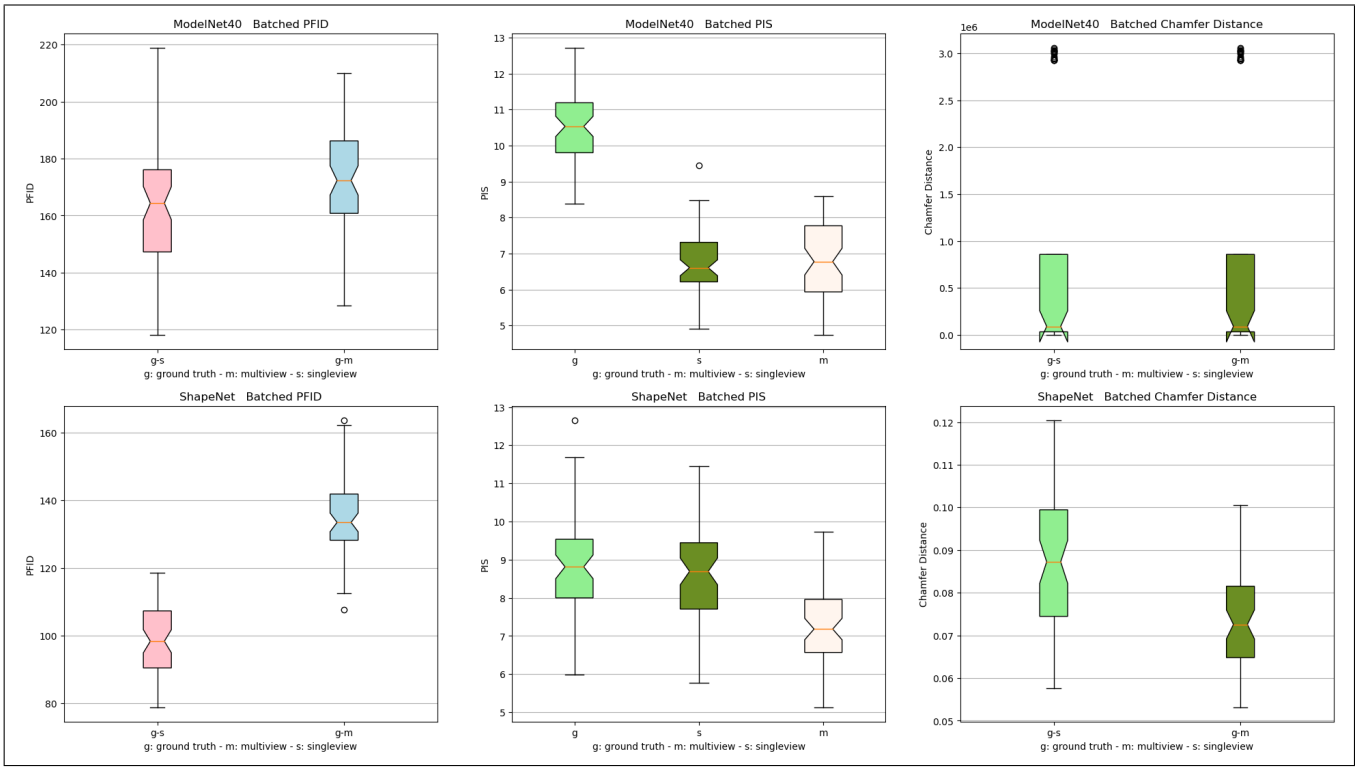

Figure 20. This plot shows the variance over the P-FID, the P-IS and the Chamfer distance. Specifically with multi-view with patch concatenation and single-view over the dataset with synthetic textures ModelNet and ShapeNetV2.

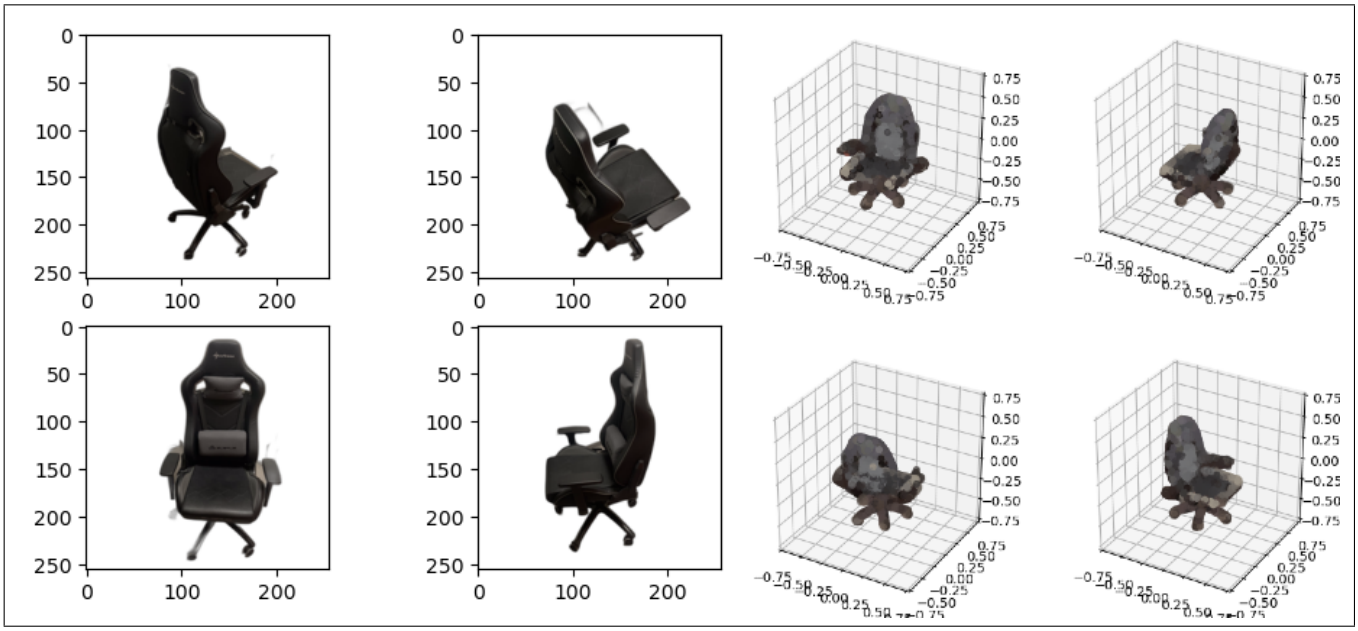

Figure 21. Gaming chair.

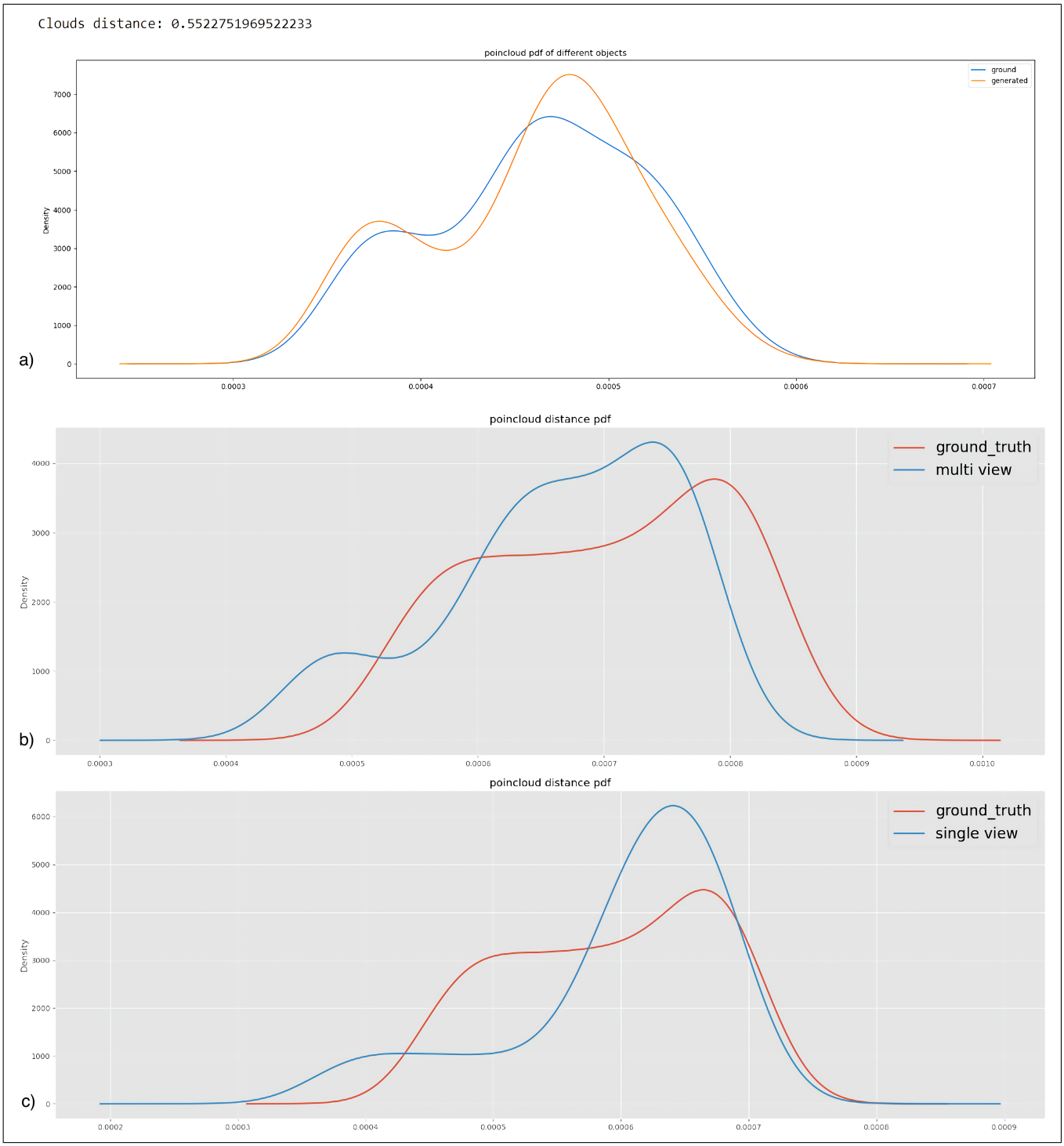

Figure 22. Show the distribution of the pairwise distances matrix of points in the same poincloud: (a) Multi-view with stochastic conditioning, (b) Multi-view with patch concatenation, (c) Single-view.