Evaluating Latent Generative Paradigms for High-Fidelity 3D Shape Completion from a Single Depth Image

Supplementary Material

Matthias Humt[1,2] Ulrich Hillenbrand[1] Rudolph Triebel[1,3]
[1]German Aerospace Center [2]TU Munich [3]Karlsruhe Institute of Technology
{matthias.humt,ulrich.hillenbrand,rudolph.triebel}@dlr.de

A. Implementation Details

Here, we extend Sec. 3 from the main text to provide further details on the implementation and training. We train our VAEs for 800 epochs with an effective batch size of 512 and a learning rate of 4×10^{-4} on four NVIDIA A100 80GB GPUs in less than a day; a fourth of the compute budget reported by Zhang et al. [26]. This is made possible through the reduction in model size (from $\sim$106 million to $\sim$35 million parameters), utilization of flash-attention [3, 4] (native to PyTorch $\geq$ 2.2), fused CUDA-kernels for NeRF encoding [16], GPU-accelerated farthest-point-sampling[1] (FPS) and `bfloat16` mixed-precision training.

All latent generative models–both diffusion and autoregressive–have approx. the same size as the one in [26] (109-164 million parameters) and are trained for 2000 epochs with an effective batch size of 256 and a learning rate of 10^{-4} on four A100 GPUs in less than two days; which again represents a fourth of the compute used by [26]. We visualized the training progress, measured FID every 25 epochs, and observed the majority of improvement occurring within the first 500 epochs.

We find that while the VAEs are more sensitive to the *range* of representable values, thus requiring `bfloat16` precision, the diffusion models require higher *resolution* and, therefore, must be trained in `float16` precision to prevent divergence.

B. Metrics

As discussed in the main text (Sec. 4.1), there is no clear consensus on the choice of evaluation metrics for 3D generative models, resulting in a great variety of metrics used. Additionally, their exact definitions and implementations can vary significantly. For this reason, this section provides the exact definition (or a reference to it) and additional details and discussion for all metrics used in our experiments.

B.1. Instance-level

These metrics rely on the comparison of individual instances, i.e., there is a one-to-one correspondence between prediction and ground truth, s.a. partial input and (best) completion.

[1] https://github.com/mit-han-lab/pvcnn/tree/master/modules

Volumetric Intersection-over-Union. The well-known *Intersection-over-Union* metric, while ubiquitously used as a bounding-box measure in object detection, can also be defined for 3D volumes to evaluate implicit functions. We follow Mescheder et al. [15] and compute the volumetric IoU for 10^5 query points randomly sampled in a unit cube with additional total padding of 0.1. It is restricted to watertight meshes and insensitive to fine details, especially at values below 50% [23] as well as oversensitive in low-volume regimes such as thin structures and walls [10]. As a result, we primarily rely on other metrics for instance-level 3D shape comparisons.

Chamfer Distance. The (bidirectional, L2 or squared) Chamfer distance (CD) between two sets of points $\mathcal{X}$ and $\mathcal{Y}$ was introduced by Fan et al. [7] and used compute COV and MMD [1] as well as 1-NNA [25] as,

$$\begin{aligned} \text{Chamfer}_{\text{L2}}(\mathcal{X}, \mathcal{Y}) = &\sum_{x \in \mathcal{X}} \min_{y \in \mathcal{Y}} \|x - y\|_2^2 \\ &+ \sum_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} \|x - y\|_2^2, \end{aligned} \tag{1}$$

and later extended to an L1 variant in Mescheder et al. [15] as the mean of an *accuracy* and *completeness* term,

$$\begin{aligned} \text{Chamfer}_{\text{L1}}(\mathcal{X}, \mathcal{Y}) = &\frac{1}{2|\mathcal{X}|} \sum_{x \in \mathcal{X}} \min_{y \in \mathcal{Y}} \|x - y\| \\ &+ \frac{1}{2|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} \|x - y\| \end{aligned} \tag{2}$$

and, as in [7, 15], multiplied by *"1/10 times the maximal edge length of the current object's bounding box"* resulting in a factor of 10.

We employ the L2 variant (eq. 1) when used within other metrics s.a. COV, MMD and 1-NNA–following their original definitions [1, 25]–but with $|\mathcal{X}| = |\mathcal{Y}| = 2048$ farthest-point-samples to increase sensitivity and reduce variance–and the L1 (eq. 2) variant with $|\mathcal{X}| = |\mathcal{Y}| = 10^5$ random samples otherwise. We found 2048 FPS points to approximately resolve details of 10^4 random points while significantly reducing computation time. We use GPU-accelerated implementations of both CD[2] and FPS. All point clouds for

[2] https://github.com/ThibaultGROUEIX/ChamferDistancePytorch

evaluation are sampled from the surface of generated and reference meshes.

Earth Mover's Distance. While frequently recognized as a more precise alternative to CD, existing Earth Mover's Distance (EMD) [7] implementations almost exclusively rely on approximate solutions and thus do not guarantee correctness[3], and are still prohibitively slow for large-scale evaluations, even in their GPU-accelerated form[4]. We, therefore, decide to omit EMD from our evaluation.

F-score, Precision & Recall. First defined as a measure for multi-view 3D reconstruction quality [12] and later introduced to 3D shape completion by Tatarchenko et al. [23], the *F-score* is the harmonic mean of *precision* and *recall*, where precision is the ratio of points in the completion that are close to the ground truth and recall is the ratio of points in the ground truth that are close to the completion. We use the default distance threshold of 0.01 and 10^5 surface samples for all evaluations.

B.2. Set-level

These metrics compare two sets of instances, such as unconditional or class-conditional generations, against the train or test split or multiple completions to a single partial input. As explained in the previous section, all set-level metrics are computed on 2048 FPS points.

Coverage & Minimum Matching Distance. For both *Coverage* (COV) and *Minimum Matching Distance* (MMD) [1], we use the definition exactly as presented in Yang et al. [25]. While neither their definition of CD nor MMD divide by the number of points, their code reveals[5], that this average is indeed taken. In doing so, the influence of the number of points on the metrics is removed. We implement a batched, GPU-accelerated version for efficient paired-distance computation between all point clouds from two sets.

Leave-One-Out 1-Nearest-Neighbor Accuracy. As for COV and MMD, we use the *Leave-One-Out* (LOO) *1-Nearest-Neighbor Accuracy* 1-NNA definition of Yang et al. [25] who proposed it as a more reliable alternative to the former. While for unconditional and class-conditional generative models, a score of 50% denotes peak performance, we point out that for instance-conditioned tasks, such as shape completion, a perfect model would achieve 0%, as the LOO NN to the ground truth shape should always be the generated completion.

Edge Count Difference. We use the definition and implementation[6] by Ibing et al. [11], who also recognized the shortcomings of COV and MMD and propose *Edge Count Difference* (ECD) as another alternative. We found that ECD frequently yields contrary results to all other metrics, thus making it seem less reliable than, e.g., 1-NNA.

Total Mutual Difference. Designed as a *diversity* measure by Wu et al. [24], the *Total Mutual Difference* (TMD) for a partial input is the sum of the LOO CD between 10 completions.

Unidirectional Hausdorff Distance. The *Unidirectional Hausdorff Distance* (UHD) [24], on the other hand, is supposed to measure *fidelity* as the average distance from 10 completions to the partial input.

Fréchet & Kernel Pointcloud Distance. Instead of in metric space, one can also compare point clouds in the higher-dimensional feature space of a pre-trained neural network to potentially capture high-level semantic information. To this end, Shue et al. [21] define a derivative of the Fréchet Inception Distance (FID) [9] as the *Fréchet Pointcloud Distance* (FPD) between two sets of point clouds. Similarly, Zhang et al. [26] propose *Kernel Pointcloud Distance* as a derivative of the *Kernel Inception Distance* (KID) [2]. We use the same 2048 FPS points to compute FPD and KPD as used for all other set-level metrics and our pre-trained VAE to extract point features. We reuse low-level functionality from the `clean-fid` [18] Python package.

Fréchet & Kernel Inception Distance. The *Fréchet Inception Distance* [9] computes the Fréchet distance between two Gaussian distributions in the feature space of the *Inception V3* [22] network pre-trained on the *ImageNet* [5] dataset. Therefore, two implicit assumptions are made: (1) The feature space follows a Gaussian distribution, and (2) the images ingested by the Inception V3 network are identically distributed to the ImageNet dataset. The more these assumptions are violated, the less reliable FID becomes [14].

The second assumption can be somewhat alleviated through the use of a different pre-trained network, potentially trained on a larger and more diverse dataset such as CLIP [19] features from a Vision Transformer [6] as proposed in Kynkäänniemi et al. [14]. We refer to this metric as FID_{CLIP}.

The *Kernel Inception Distance* [2] is a non-parametric alternative to FID, which uses the *Maximum Mean Discrepancy* [8] to compare the feature distributions of two sets of images and therefore relaxes the Gaussian assumption.

To measure the perceptual quality of 3D data, FID and KID are adapted to the 3D domain by Zheng et al. [27] and Zhang et al. [26] respectively through rendering of shaded images from 20 uniformly distributed viewpoints around the object. *Shading-image-based* FID and KID are the average FID and KID across all views.

[3] https://github.com/facebookresearch/pytorch3d/issues/211

[4] https://github.com/Colin97/MSN-Point-Cloud-Completion/tree/master/emd

[5] https://github.com/stevenygd/PointFlow/blob/master/metrics/evaluation_metrics.py

[6] https://github.com/GregorKobsik/Octree-Transformer/blob/master/evaluation/evaluation.py

FID decompositions. Finally, Sajjadi et al. [20] propose a decomposition of FID into *Precision* and *Recall*, improved upon by Kynkäänniemi et al. [13], which is the definition we use throughout this work.

Naeem et al. [17] acknowledge the improvements made by Kynkäänniemi et al. [13] but find remaining failure cases of the improved precision and recall formulations and therefore propose *Density* and *Coverage* as drop-in replacements.

We further propose to also decompose FPD to obtain an even more detailed view of the generative performance of 3D data.

B.3. Recommendations

Based on our extensive empirical evaluation and literature review, we recommend the following metrics for the evaluation of 3D generative models in general and the shape completion task in particular:

- For *instance-level* evaluation, we only recommend the F1-score but highly recommend the precision and recall decomposition. All other metrics in this category, like CD, EMD, and IoU, feature at least one highly problematic aspect, as discussed in their dedicated sections.
- For *set-level* evaluation, we strongly recommend KPD and FPD, especially with a task-specific feature extractor (ideally a VAE), but shading-image-based FID and KID are viable alternatives. For both FID and FPD, we recommend the (improved) precision and recall decomposition to gain valuable insights into the origin of the observed performance. The only non-feature-based metric we recommend is 1-NNA.

C. Additional Results

C.1. Quantitative Results

	Normal Consistency [15] ↑	IoU ↑
VAE	**95.966**	**93.635**
VQ-VAE	92.065	85.453

Table 1. Reconstruction quality; class average. Watertight meshes only. Extends Tab. 1.

C.2. Qualitative Results

	D=32	kmeans	N=16k	revive	sample	IoU ↑
FSQ						81.2
LFQ						79.9
VQ						78.9
	✓					81.3
		✓				85.9
		✓	✓			89.2
		✓	✓	✓		88.8
		✓	✓		✓	89.3

Table 2. VQ-VAE ablations.

		COV↑	MMD↓	ECD ↓	Dens.↑	Cov.↑
Chair	LAS-Diff.	45.79	**3.522**	80	**1.30**	0.89
	3DS2VS	**51.55**	3.531	26	0.79	0.83
	Ours	50.81	3.588	**7**	1.30	**0.91**
Plane	LAS-Diff.	38.12	1.249	164	0.44	0.32
	3DS2VS	48.27	1.059	37	0.46	0.46
	Ours	**50.00**	**1.058**	**10**	**0.72**	**0.56**
Car	LAS-Diff.	28.57	**0.992**	**483**	0.27	**0.36**
	3DS2VS	25.50	1.231	2036	0.22	0.18
	Ours	**37.38**	1.088	538	**0.28**	0.30
Table	LAS-Diff.	49.88	**3.111**	136	1.00	0.86
	3DS2VS	50.94	3.249	20	0.93	0.83
	Ours	**52.82**	3.187	**16**	**1.20**	**0.87**
Rifle	LAS-Diff.	32.49	0.950	180	0.53	0.22
	3DS2VS	45.15	**0.847**	39	0.71	0.42
	Ours	**46.41**	0.895	**12**	**0.81**	**0.45**
Mean	LAS-Diff.	38.97	1.965	209	0.71	0.53
	3DS2VS	44.28	1.983	432	0.62	0.54
	Ours	**47.49**	**1.963**	**117**	**0.86**	**0.62**

Table 3. Comparison of *class-conditional* generative models. MMD$\times 10^3$. Extends Tab. 2.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018.

[2] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

[3] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

[4] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[6] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[7] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.

	Diffusion (VAE)	AR (VQ-VAE)
$\text{FID}_{\text{CLIP}} \downarrow$	3.597	**3.581**
Density $\uparrow$	**0.303**	0.293
Coverage $\uparrow$	**0.330**	0.292
1-NNA $\uparrow$	**63.938**	68.137
FPD $\downarrow$	**74.420**	79.425
KPD $\downarrow$	**4.198**	4.919
Precision $\uparrow$	**56.558**	56.045
Recall $\uparrow$	**59.653**	54.394
COV $\uparrow$	**48.331**	45.419
MMD$\times 10^3 \downarrow$	2.382	**2.344**
ECD $\downarrow$	**60.020**	124.568
Density $\uparrow$	**1.026**	1.013
Coverage $\uparrow$	**0.749**	0.723

Table 4. Comparison of diffusion and autoregressive *unconditional* generative shape modeling on continuous (VAE) and discrete (VQ-VAE) latents. Extends Tab. 3.

	VQ-VAE		VAE
	Diffusion	Autoregressive	Diffusion
$\text{FID}_{\text{CLIP}} \downarrow$	4.675	**3.319**	3.154
Density $\uparrow$	0.189	**0.301**	0.338
Coverage $\uparrow$	0.195	**0.306**	0.350
COV $\uparrow$	46.129	**47.159**	48.278
MMD$\times 10^3 \downarrow$	2.459	**2.314**	2.349
ECD $\downarrow$	128.000	**102.317**	73.034
Density $\uparrow$	**1.076**	0.977	1.009
Coverage $\uparrow$	**0.746**	0.721	0.746

Table 5. Comparison of diffusion and autoregressive *class-conditional* generative shape modeling on the same latent space. Extends Tab. 4.

Figure 1. Real-world examples using depth data from a Kinect sensor. From left to right: **input**, **ground truth**, **generative** (best), and **discriminative**.

[8] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[9] 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. *Advances in neural information processing systems*, 30, 2017.

[10] Matthias Humt, Dominik Winkelbauer, and Ulrich Hillenbrand. Shape completion with prediction of uncertain regions. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1215–1221. IEEE, 2023.

[11] Moritz Ibing, Isaak Lim, and Leif Kobbelt. 3d shape generation with grid-based implicit functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13559–13568, 2021.

[12] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36 (4):1–13, 2017.

[13] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Neural Information Processing Systems*, 2019.

[14] Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. The role of imagenet classes in fr\'echet inception distance. *arXiv preprint arXiv:2203.06026*, 2022.

[15] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019.

[16] Thomas Müller. tiny-cuda-nn. GitHub repository, 2021. Version 1.7, BSD-3-Clause.

[17] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *International Conference on Machine Learning (ICML)*, 2020.

[18] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On

	COV ↑		MMD×10^3 ↓		ECD ↓		Prec. ↑		Rec. ↑		TMD ↑	UHD ↓
	D	**G**	**D**	**G**	**D**	**G**	**D**	**G**	**D**	**G**		
Chair	62.63	**66.77**	2.671	**2.460**	118	**95**	80.35	**87.74**	48.89	**85.82**	3.685	6.590
Plane	58.91	**60.64**	0.937	**0.817**	119	**73**	75.74	**85.40**	14.85	**48.27**	2.301	4.939
Car	31.38	**44.86**	1.213	**1.033**	1445	**395**	27.37	**49.27**	12.02	**54.47**	2.846	5.551
Table	62.12	**65.41**	2.437	**2.334**	36	**16**	84.82	**95.76**	64.12	**79.18**	4.660	5.570
Rifle	49.79	**53.16**	**0.697**	0.698	83	**74**	75.11	**91.14**	**40.93**	33.76	3.132	4.977
Mean	52.96	**58.17**	1.591	**1.469**	360	**131**	68.68	**81.86**	36.16	**60.30**	3.325	5.525
All	56.36	**60.49**	1.873	**1.785**	**189**	238	77.17	**88.48**	41.09	**70.74**	/	/

Table 6. **Generative (G)** vs. **discriminative (D)** shape completion from a single Kinect depth image. TMD and UHD from 10 generations. Extends Tab. 7.

aliased resizing and surprising subtleties in gan evaluation. In *CVPR*, 2022.

[19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[20] Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lučić, Olivier Bousquet, and Sylvain Gelly. Assessing Generative Models via Precision and Recall. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[21] J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20875–20886, 2023.

[22] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2015.

[23] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3405–3414, 2019.

[24] Rundi Wu, Xuelin Chen, Yixin Zhuang, and Baoquan Chen. Multimodal shape completion via conditional generative adversarial networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 281–296. Springer, 2020.

[25] 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/CVF international conference on computer vision*, pages 4541–4550, 2019.

[26] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions on Graphics (TOG)*, 42(4):1–16, 2023.

[27] Xinyang Zheng, Yang Liu, Pengshuai Wang, and Xin Tong. Sdf-stylegan: Implicit sdf-based stylegan for 3d shape generation. In *Computer Graphics Forum*, pages 52–63. Wiley Online Library, 2022.