OmniObject3D: Large-Vocabulary 3D Object Dataset for Realistic Perception, Reconstruction and Generation Supplementary Materials

Tong Wu^{1,2}, Jiarui Zhang^{1,3}, Xiao Fu¹, Yuxin Wang^{1,4}, Jiawei Ren⁵, Liang Pan⁵, Wayne Wu¹, Lei Yang^{1,3}, Jiaqi Wang¹, Chen Qian¹, Dahua Lin^{1,2⊠}, Ziwei Liu^{5⊠}

¹Shanghai Artificial Intelligence Laboratory, ²The Chinese University of Hong Kong, ³SenseTime Research,
⁴Hong Kong University of Science and Technology, ⁵S-Lab, Nanyang Technological University

A. Additional Information of OmniObject3D

We first provide a full category list with the number of objects for each class in Figure S1. Most of the categories have [10, 40] objects. The dataset includes objects that have undergone common manipulations, as shown in Figure S3 (b). For each object, the raw data includes a textured 3D mesh and several surrounding videos. To demonstrate the completeness and high quality of our scanned objects, we compare the quality between the COLMAP sparse reconstruction and the textured mesh from the scanner in Figure \$3 (c). Given a high-fidelity 3D scan, we can render realistic and high-resolution multi-view images with modern graphics engines like the Blender [14], where we also save the corresponding depth and normal maps (Figure S2) for different research usage. We also provide the users with posed frames from the real-captured videos following [62]. We leverage the calibration board and COLMAP [65] to recover the poses of selected frames with a real-world scale. as described in the main text, and then we develop a matting pipeline based on a two-stage U²-Net [61] model together with a post-processing FBA [19] model. In detail, we first utilize the Rembg ¹ tool on image frames to remove backgrounds from different categories and choose 3,000 good results as the pseudo segmentation labels. We then refine our pipeline by fine-tuning with the pseudo labels to boost its segmentation ability on objects. We show some examples and failure cases of our segmentation pipeline in Figure \$3 (a).

B. Related Works

We have briefly discussed the related works for the four benchmarks in the main text, and we conduct a more comprehensive discussion here.

Robust Point Cloud Perception. Robustness to out-of-distribution (OOD) data has been an important topic in point

cloud perception since point clouds are widely employed in safety-critical applications, e.g., autonomous driving. In particular, OOD styles (e.g., different styles in CAD models and real-world objects) and OOD corruptions (e.g., missing points) are two main challenges to point cloud perception robustness. A line of work [11, 36, 59, 78] evaluates the robustness to OOD corruptions by adding corruptions like random jittering and rotation to clean test sets. Recent work works [63,71] further systematically anatomize the corruptions and propose a standard corruption test suite. However, they fail to take account of OOD styles. Another line of work [2, 62] evaluates the sim-to-real domain gap by testing models trained on clean synthetic datasets (e.g., ModelNet-40 [81]) on noisy real-world test sets (e.g., ScanObjectNN [73]). However, the sim-to-real gap couples OOD styles and OOD corruptions at the same time, which makes the results hard to analyze. In this work, we use OmniObject3D dataset to provide high-quality real-world point cloud to measure the OOD style robustness, and apply systematic corruptions on top of it to measure the OOD corruptions robustness. We hence provide the first point cloud perception benchmark that allows fine-grained evaluation of the robustness on both OOD styles and corruptions.

Neural Radiance Field. Neural radiance field (NeRF) [48] represents a scene with a fully-connected deep network (MLPs), which takes in hundreds of sampled points along each camera ray and outputs the predicted color and density. Novel views of the scene are synthesized by projecting the colors and densities into an image via volume rendering. Inspired by the success of NeRF, a massive follow-up effort has been made to improve its quality [4,5,47,75], and efficiency [8,20,50,70]. A branch of works [9,40,62,77,89] has also explored the generalization ability of NeRF-based frameworks. PixelNeRF [89], MVSNeRF [9], IBRNet [77], and NeuRay [40] reconstruct the radiance field with a mere forward pass during inference via training on cross-scenes. NeRFormer [62], IBRNet [77], and GNT [74] leverage

¹https://github.com/danielgatis/rembg

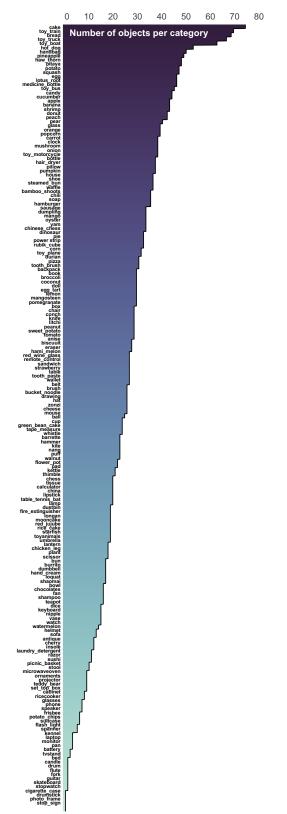


Figure S1. A full class list with number of objects per category.

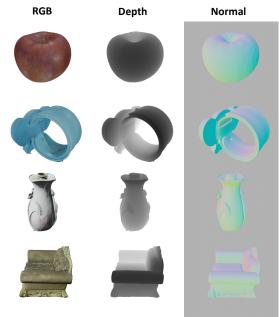


Figure S2. Examples of the Blender [14] rendered results.

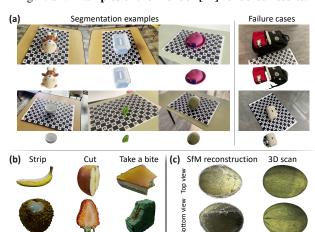


Figure S3. Examples of the segmentation (a), manipulation (b), and reconstruction (c). In (c), the missing bottom of the SfM reconstruction from video frames is due to its touch with the table.

Transformers for generalizable NeRF.

Neural Surface Reconstruction. Implicit Neural Representations (INR) [3, 12, 31, 41, 45, 56, 64, 68, 72, 91] of 3D object geometry and appearance with neural networks have attracted increasing attention in recent years. Some approaches [35, 38, 53, 88] regard the color of an intersection point between the ray and the surface as the rendered color, namely surface rendering, and they typically rely on accurate object masks. Another trend of recent approaches [15,54,76,80,87] proposes to leverage neural radiance field with implicit surface representations like Signed Distance Function (SDF) for higher-quality and mask-free surface reconstruction from multi-view images. NeuS [76],

Table R1. Point cloud perception robustness analysis on OmniObject3D with different architecture designs. Models are trained on ModelNet-40 dataset. OA on OmniObject3D evaluates the robustness to OOD styles. mean Corruption Error (mCE) on the corrupted OmniObject3D-C evaluates the robustness to OOD corruptions. The blue cells denote best in each row, and the red cells denote the worst.

	OA _{Clean} ↑	OA _{Style} ↑	Scale	Jitter	Drop-G	Drop-L	Add-G	Add-L	Rotate	mCE ↓
DGCNN [78]	0.926	0.448	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
PointNet [59]	0.907	0.466	0.925	0.858	0.976	0.816	1.318	0.921	0.935	0.969
PointNet++ [60]	0.930	0.407	1.104	1.071	1.108	0.886	1.101	1.123	1.031	1.066
RSCNN [39]	0.923	0.393	1.115	1.078	1.144	0.997	1.042	1.079	1.025	1.076
SimpleView [24]	0.939	0.476	0.940	0.951	0.959	1.012	1.043	1.037	0.949	0.990
GDANet [84]	0.934	0.497	0.887	0.933	0.923	0.975	0.884	0.921	0.882	0.920
PAConv [83]	0.936	0.403	1.034	1.101	1.032	1.052	1.159	1.057	1.082	1.073
CurveNet [82]	0.938	0.500	0.930	0.930	0.920	0.869	0.929	0.997	0.907	0.929
PCT [26]	0.930	0.459	0.950	0.986	1.011	0.862	0.921	0.912	1.001	0.940
RPC [63]	0.930	0.472	0.947	0.940	0.967	<u>0.855</u>	0.999	0.909	0.915	0.936

Table R2. Comparisons of 3 single-scene NVS methods on different data types. For all the methods we involve, we can observe that the *Blender* setting performs the best; the *SfM-wo-bg* setting is a little bit worse due to the motion blur and potential inaccuracy in SfM pose estimation; the *SfM-w-bg* setting always achieves the lowest PSNR, as the background in the unbounded scene introduces further challenges.

Method	Data-type	PSNR (†)		
NeRF [48]	SfM-w-bg SfM-wo-bg Blender	22.92 24.70 28.07		
Mip-NeRF [4]	SfM-w-bg SfM-wo-bg Blender	23.29 25.62 31.25		
Plenoxel [20]	SfM-w-bg SfM-wo-bg Blender	14.06 19.18 28.07		

VolSDF [87] reconstruct implicit surfaces with an SDF-based volume rendering scheme, and Voxurf [80] leverages an explicit volumetric representation for acceleration. Since dense camera views of scenes are sometimes unavailable, SparseNeuS [42] and MonoSDF [90] explore surface reconstruction from sparse views. The former exploits generalizable priors cross scenes for a generic surface prediction, while the latter takes advantage of the estimated geometry cues predicted by pretrained networks.

OmniObject3D can serve as a large-scale benchmark with realistic photos and meshes for both training and evaluation. It bears a large vocabulary and high diversity in shape and appearance, offering an opportunity for pursuing more generalizable and robust novel view synthesis and surface reconstruction methods.

3D Object Generation. Recent advances in photorealistic 2D image generations [16, 18, 29, 32–34, 57] inspire the explorations of 3D content generation. Early approaches [21, 28, 43, 69, 79] extend 2D generation frame-

works to 3D voxels with a high computational cost when generating high-resolution contents. Some other works adopt different 3D data formulations, e.g., point cloud [1, 49, 86, 92] and octree [30] to generate coarse geometry. OccNet [46], IM-NET [13] generates the 3D meshes with implicit representation while extracting high-quality surfaces is non-trivial. Encouraged by NeRF [48], extensive works [6, 7, 25, 27, 52, 55, 66, 67, 85, 93] explore 3Daware image synthesis rather than mesh generation. Aiming at generating textured 3D meshes, Textured3DGAN [58] and DIBR [10] deform template meshes, preventing them from complex shapes. PolyGen [51], SurfGen [44], and GET3D [22] generate meshes with arbitrary topology. Distinguishable from others, GET3D generates diverse meshes with rich geometry and textures. With the proposed OmniObject3D dataset, we extend the benchmarks of realistic 3D generation approaches to large vocabulary and massive objects, enabling the exploration of better generation quality and diversity.

C. Additional Experimental Results

C.1. Robust 3D Perception

Following ModelNet-C [63], we perform seven kinds of out-of-distribution (OOD) corruptions for study, including "Scale", "Jitter", "Drop Global/Local", "Add Global/Local", and "Rotate". Please refer to their paper for a detailed illustration of each corruption type. We calculate the error under each corruption and the mean Corruption Error (mCE) is an average of the results. The full evaluation results corresponding to are shown in Table R1.

C.2. Novel View Synthesis

C.2.1 Single-Scene NVS

Implementation Details. We use the official code and default settings by NeRF [48], Mip-NeRF [5], and Plenoxels [20] in this section. For NeRF, we re-weight the fore-

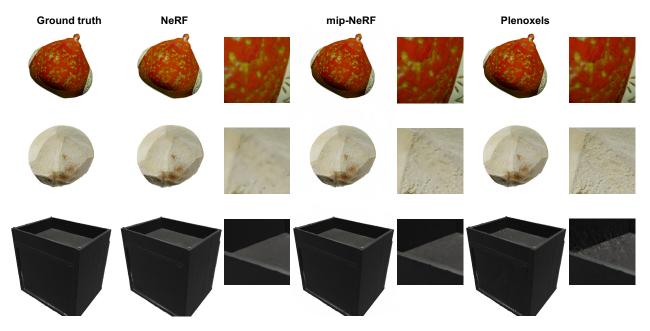


Figure S4. Qualitative comparisons of single-scene NVS methods in different rendered scenes from our dataset.

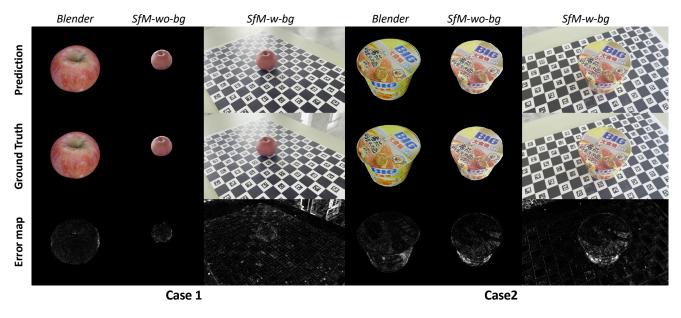


Figure S5. Qualitative comparisons of NVS on the same scenes with different data dypes.

ground and background contents by 1:0.5 to avoid all-black output. For Plenoxel on the SfM data with background, we enable the background model provided by the official code to model the background area.

Qualitative Comparisons of NVS on rendered images.

We describe the performance of three representative methods in the main text, and we provide some qualitative comparisons here in Figure S4, accordingly. Plenoxels are especially good at modelling high-frequency textures (*e.g.*, the coconut), while it is less robust then NeRF and mip-NeRF

when dealing with dark textures and concave geometry, suffering from inaccurate geometry. Our dataset helps to provide a comprehensive evaluation of different methods.

Comparisons of NVS on rendered images and iPhone videos. We conduct qualitative and quantitative evaluations on novel view synthesis with several scenes under different data types, including *SfM-wo-bg*, *SfM-w-bg* and *Blender*. The *SfM-wo-bg* and *SfM-w-bg* settings use images sampled from iPhone videos and camera parameters generated by COLMAP. The difference between them is whether the

Table R3. Cross-scene novel view synthesis results on 10 categories. We evaluate our benchmarks on 3 unseen scenes per category with 3 source views. In each scene, we take 10 test frames widely distributed around the object by FPS sampling strategy.

Method	Train	Metric	toy train	bread	cake	toy boat	hot dog	wallet	pitaya	squash	handbag	apple
		PSNR	15.90	16.80	15.47	16.28	15.84	20.58	18.69	17.81	18.02	19.55
	All*	SSIM	0.501	0.548	0.522	0.519	0.497	0.534	0.490	0.576	0.564	0.681
	All	LPIPS	0.480	0.456	0.480	0.408	0.429	0.449	0.456	0.417	0.444	0.403
		$\mathcal{L}_1^{ ext{depth}}$	0.182	0.155	0.249	0.253	0.127	0.261	0.178	0.187	0.229	0.113
		PSNR	16.14	16.87	14.60	15.65	16.64	20.76	19.09	16.97	18.35	20.40
	Cat.	SSIM	0.515	0.560	0.527	0.444	0.520	0.524	0.505	0.548	0.575	0.709
	Cat.	LPIPS	0.475	0.463	0.488	0.433	0.431	0.464	0.449	0.435	0.444	0.399
MVSNeRF [9]		$\mathcal{L}_1^{ ext{depth}}$	0.175	0.127	0.339	0.477	0.134	0.382	0.237	0.101	0.219	0.112
WYSINCIA [7]		PSNR	23.16	25.82	25.14	23.47	23.91	27.83	25.36	25.68	26.09	30.53
	All*-ft	SSIM	0.717	0.769	0.745	0.736	0.714	0.739	0.710	0.761	0.803	0.845
	All -It	LPIPS	0.281	0.224	0.263	0.228	0.248	0.293	0.227	0.255	0.280	0.215
		$\mathcal{L}_1^{ ext{depth}}$	0.091	0.062	0.081	0.141	0.053	0.078	0.069	0.061	0.130	0.053
		PSNR	22.88	25.58	25.29	23.80	23.44	27.38	25.46	25.40	25.94	30.06
	Catft	SSIM	0.721	0.758	0.748	0.733	0.698	0.722	0.715	0.759	0.803	0.840
	Cat11	LPIPS	0.283	0.243	0.262	0.226	0.280	0.318	0.229	0.277	0.283	0.244
		$\mathcal{L}_1^{ ext{depth}}$	0.122	0.053	0.064	0.096	0.060	0.084	0.071	0.048	0.120	0.046
	All*	PSNR	17.90	19.08	17.09	17.89	17.77	23.13	20.11	20.25	18.36	22.36
		SSIM	0.526	0.599	0.538	0.530	0.516	0.579	0.511	0.632	0.530	0.726
		LPIPS	0.430	0.383	0.422	0.368	0.394	0.426	0.405	0.356	0.451	0.352
		$\mathcal{L}_1^{ ext{depth}}$	0.379	0.327	0.610	0.357	0.338	0.419	0.388	0.392	0.847	0.175
	Cat.	PSNR	17.33	18.30	16.87	17.13	17.83	23.39	19.62	19.05	19.73	21.02
		SSIM	0.502	0.554	0.525	0.491	0.498	0.579	0.485	0.606	0.584	0.684
		LPIPS	0.449	0.415	0.446	0.394	0.413	0.427	0.420	0.376	0.443	0.371
IBRNet [77]		$\mathcal{L}_1^{ ext{depth}}$	0.417	0.394	0.392	0.169	0.096	0.234	0.177	0.352	0.336	0.331
ibitatet [77]		PSNR	22.12	27.53	26.28	25.80	22.89	30.03	26.33	29.15	26.74	32.00
	All*-ft	SSIM	0.683	0.829	0.769	0.834	0.686	0.814	0.764	0.845	0.815	0.885
	All · -It	LPIPS	0.298	0.177	0.238	0.152	0.267	0.211	0.199	0.177	0.268	0.163
		$\mathcal{L}_1^{ ext{depth}}$	0.232	0.051	0.079	0.083	0.054	0.036	0.075	0.051	0.073	0.080
		PSNR	21.90	26.47	24.83	22.46	24.74	27.68	26.41	25.37	26.61	30.18
	Catft	SSIM	0.678	0.804	0.739	0.707	0.755	0.727	0.766	0.745	0.813	0.861
	CatIt	LPIPS	0.301	0.195	0.261	0.233	0.210	0.280	0.197	0.254	0.266	0.184
		$\mathcal{L}_1^{ ext{depth}}$	0.225	0.049	0.070	0.101	0.046	0.063	0.062	0.195	0.065	0.111
		PSNR	19.77	21.54	20.77	20.15	20.93	24.73	21.78	23.48	21.30	27.18
	All*	SSIM	0.647	0.701	0.690	0.661	0.671	0.666	0.606	0.748	0.696	0.833
	All	LPIPS	0.377	0.331	0.363	0.315	0.339	0.393	0.370	0.283	0.381	0.269
pixelNeRF [89]		$\mathcal{L}_1^{ ext{depth}}$	0.142	0.131	0.141	0.109	0.073	0.085	0.114	0.065	0.175	0.061
PINCHINGKE [09]		PSNR	19.91	20.93	17.55	20.20	19.63	24.16	20.80	18.59	19.84	24.96
	Cat	SSIM	0.685	0.702	0.622	0.686	0.645	0.662	0.606	0.667	0.657	0.828
	Cat.	LPIPS	0.332	0.330	0.426	0.275	0.348	0.392	0.367	0.342	0.420	0.249
		$\mathcal{L}_1^{ ext{depth}}$	0.136	0.224	0.364	0.119	0.142	0.152	0.243	0.181	0.336	0.054

background is included. The *Blender* data are rendered by Blender [14]. Since the image resolutions and foreground proportions are different among the data types, we calculate the PSNR metric only in the foreground area for *SfM-wo-bg* data and *Blender* data, whereas for *SfM-w-bg* data, every pixel in the image is included PSNR calculation.

Based on the qualitative comparisons in Figure S5, we observe that for both two selected scenes, the predicted novel view image under the *Blender* setting achieves the best visual quality, resulting in the highest PSNR in Table R2. When comparing the two SfM based data types, we find that the quality of the foreground object from the

SfM-wo-bg data is only slightly better than the other, while the high background error under the *SfM-w-bg* setting leads to a significant drop in performance, as shown in Table R2. The experimental results shed light on how real-captured videos introduce further challenges to NeRF-like methods. We demonstrate that performing robust novel view synthesis with casually captured videos will be an important and practical topic.

Table R4. Unaligned Cross-scene novel view synthesis results of pixelNeRF-U [89] on 10 categories.

Train	Metric	toy train	bread	cake	toy boat	hot dog	wallet	pitaya	squash	handbag	apple
·	PSNR	18.81	19.92	19.86	19.54	19.64	20.31	20.44	20.74	20.79	21.21
	10111	-0.96	-1.62	-0.91	-0.29	-1.29	-4.42	-1.34	-2.74	-0.51	-5.97
All*	SSIM	0.591	0.625	0.636	0.626	0.627	0.628	0.619	0.631	0.635	0.650
All	2211/1	-0.056	-0.076	-0.054	-0.035	-0.044	-0.038	+0.013	-0.117	-0.061	-0.183
	LPIPS	0.432	0.406	0.405	0.398	0.397	0.401	0.405	0.394	0.397	0.390
	LPIPS	-0.055	-0.075	-0.042	-0.083	-0.058	-0.008	-0.035	-0.111	-0.016	-0.121
	depth	0.145	0.118	0.123	0.132	0.122	0.120	0.119	0.113	0.121	0.117
	$\mathcal{L}_1^{ ext{depth}}$	-0.003	+0.013	+0.018	-0.023	-0.049	-0.035	-0.005	-0.048	+0.054	-0.056
	PSNR	19.36	19.03	18.46	18.45	18.53	19.41	19.51	19.34	19.38	19.58
	FSINK	-0.55	-1.90	-0.91	-1.75	-1.10	-4.75	-1.29	-0.75	-0.46	-5.38
Cat.	SSIM	0.637	0.636	0.626	0.624	0.623	0.625	0.616	0.614	0.618	0.631
Cat.	SSIM	-0.048	-0.066	+0.004	-0.062	-0.022	-0.037	+0.010	-0.053	-0.039	-0.197
	LPIPS	0.392	0.402	0.415	0.400	0.396	0.399	0.403	0.404	0.408	0.404
	LPIPS	-0.060	-0.072	+0.011	-0.125	-0.048	-0.007	-0.036	-0.062	+0.012	-0.155
	depth	0.172	0.219	0.260	0.262	0.247	0.251	0.252	0.286	0.293	0.276
	$\mathcal{L}_1^{ ext{depth}}$	-0.036	+0.005	+0.104	-0.143	-0.105	-0.099	-0.009	-0.105	+0.043	-0.222

C.2.2 Cross-Scene NVS

Implementation Details. We use the official codes to evaluate three benchmarks on 10 categories, i.e., toy train, bread, cake, toy boat, hot dog, wallet, pitaya, squash, handbag, and apple. We split three scenes from each category as a test-set, and the remaining scenes are used as a train-set. During training, we randomly sample rays from scenes in the train-set of each category and use Adam [37] optimizer. For a fair comparison, we evaluate these methods with the same source views, i.e., 3 views from nearby 30 views (explained in Sec. C.3.2) by FPS sampling. Then in a scene with 100 rendered views, we exclude these 3 source views and select 10 test views from the remaining 97 views by FPS criteria again. For MVSNeRF, we pretrain the 'All*' with total 300k iterations, and the 'Cat.' with 20k to 40k iterations depending on the number of scenes. In finetuning stage, we take 3 views as input and additional 13 views sampling for per-scene optimization. Each scene is finetuned for 15k iterations. For IBRNet, we pretrain the 'All*' with 300k iterations, and the 'Cat.' with 50k iterations. After cross-scene training, we further finetune the model with 15k iterations on each test scene. For pixelNeRF, we train the 'All*' with 400k iterations, and the 'Cat.' with 12k to 30k iterations depending on the number of scenes. All methods sample rays within a tight foreground bounding box around the object.

Detailed Comparisons. The full evaluation results are presented in Table R3. We additionally provide qualitative comparisons of 4 cases, each with rendered RGB and depth, as shown in Figure S6 (we leave an extra 15 pixels of each edge). We evaluate depth within the foreground masks. From the visualization, it may seem that methods w/ 'Cat.' generate more accurate contour than that w/ 'All*', contra-

dicting the statement that methods w/ 'All*' can learn a better geometric cue than that w/ 'Cat.' in the main context. However, we find that within the masks, the depth of the former is generally more precise than that of the latter, obviously illustrated by "pitaya" (the third case) in pixelNeRF. It may raise an interesting research topic of how generic methods can perform both accurately in shape contour and geometry. After slightly finetuning MVSNeRF and IBRNet on a test scene, these methods achieve comparable performance with scene-specific methods, e.g., NeRF.

Results on Unaligned Coordinate System. We additionally provide a more challenging setting by evaluating Cross-Scene NVS on an unaligned coordinate system rather than in a perfectly predefined canonical space. Specifically, we examine pixelNeRF-U [89], where the coordinate system of each object is randomly rotated by $\theta (\sim 60^{\circ} \cdot \mathcal{N}(0,1))$ in three axes and translated by $[0.5, 0.5, 0.5] \cdot \mathcal{N}(0, 1)$. As detailedly illustrated in Table R4 and Figure S7, the PSNR drops with All*:22.16 \to 21.20, Cat.: 20.65 \to 19.58, particularly for apple and wallet, and the geometry also suffers except for bread, cake, and handbag, resulting in a gennerally more blurry and irregular-shaped appearance. We infer that since xyz is fed into the network, the coordinates will implicitly store category-specific priors, e.g., a specific sampled 3D location in canonical space will learn the prior of head or tail (other elements) of toy train. Thus the misalignment will tend to impair this learned variance of the rigid scene. In our experiment, we manually perform nonalignment in a regular mathematical manner, we believe this impairment will become more severe when applied to a naturally-unaligned coordinate system.

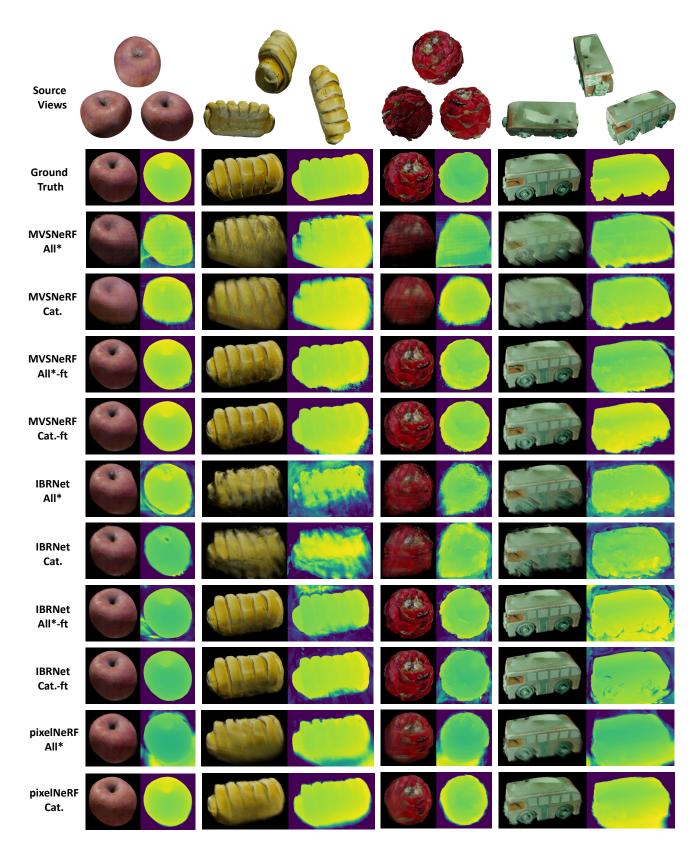


Figure S6. Qualitative comparisons of several cross-scene NVS methods in different scenes from our dataset.

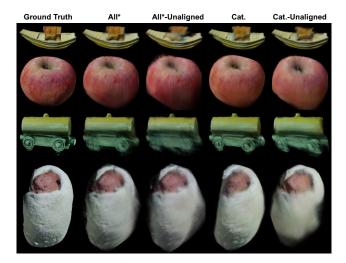


Figure S7. Qualitative comparison of pixelNeRF-U and pixel-NeRF. The former shows a more blurry and irregular-shaped appearance.

C.3. Neural Surface Reconstruction

C.3.1 Dense-View Surface Reconstruction

Implementation Details. We use the publicly available code for NeuS [76] and VolSDF [87], and we use the code provided by the authors for Voxurf [80], training with for a standard number of iteration on each of them. For all the methods, we do not involve the mask loss as supervision. Each scene is trained on 100 views. We use the Chamfer Distance between the reconstructed surface and the ground truth mesh for evaluation. The distance is calculated in a normalized space (all coordinates lying within [-1,1]). We clip the distance by 0.1 to alleviate the huge effect of outliers. We will release the standard evaluation code.

Qualitative Comparisons. In the main text, we split the categories into three difficulty levels, namely *hard*, *medium*, and *easy*. Figure S8 shows some examples from each level. We observe that the "hard" examples usually suffer from dark and low-texture appearance (*e.g.*, the pan), concave geometry (*e.g.*, the vase and the kennel), and complex or thin structures (*e.g.*, the durian, the fork, and the toy train). The "medium" and "easy" cases usually have a simple geometry with proper texture. The wide exploration of geometry and textures of the dataset helps to provide a comprehensive evaluation of different methods.

C.3.2 Sparse-View Surface Reconstruction

Implementation Details. For NeuS [76] and MonoSDF [90], we use FPS sampling to sample 3 views from all the 100 views. We train 10k iterations for NeuS and 500 epochs for MonoSDF, both being reduced from the original setting due to the few-view input. For

Table R5. Sparse-view surface reconstruction results with a range of views.

	Chamfer Distance $\times 10^3 (\downarrow)$								
Method	2 views	3 views	5 views	8 views					
NeuS [76]	41.06	27.3	12.65	7.96					
MonoSDF [90]	45.35	34.68	23.02	18.97					

SparseNeuS [42], we fix the first three examples in each category as the testing set and skip them when training. We conduct FPS among the nearest 30 camera poses from a random reference view at inference time. The fine-tuning stage of SparseNeuS is not stable: the training usually collapses before convergence, and the issue also exists for the officially used DTU dataset. So we report the results via direct inference for all the experiments.

Detailed Comparisons. In Table 6 of the main text, we surprisingly find that NeuS can serve as a strong baseline under the sparse-view setting without bells and whistles. MonoSDF is enhanced by depth and normal estimations from pre-trained networks [17], and it claims a superior performance on DTU with only 3 views as input. However, MonoSDF does not seem to perform as well as NeuS in OmniObject3D.

As shown in Figure S9, the NeuS baseline with FPS sampling is especially good at dealing with thin structures: the wide-spread views together with the black backgrounds help to bound the geometry well. However, the depth estimation is especially inaccurate in these scenarios, which is probably caused by the gap between the training and testing images of the depth estimation network. Nevertheless, it shows great performance in some cases for maintaining a coherent shape and adding some geometry details. It is an interesting problem to explore a robust usage of the estimated geometry cues under different circumstances.

Sparse-view surface reconstruction with a range of view numbers. In addition to the default setting of 3 views, we try a range of views (*i.e.*, 2, 3, 5, 8 views) with FPS sampling for NeuS [76] and MonoSDF [90], and the results are shown in Table. **R5**. For NeuS, we observe a significant improvement in accuracy as the view number increases from 2 to 8, but the 8-view setting (7.96) is still worse than the 100-view setting (6.09) with a clear margin. For MonoSDF, the improvement begins to slow down when lifting from 5 views to 8 views. This problem is probably due to the inaccurate depth guidance, as described above.

View Selection Range for Cost Volume Initialization. In MVSNeRF [9], due to occlusions, initialized local cost volume feature is inconsistent with large viewpoint changes, causing poor geometry extracted from the global density field. One naïve solution is to decrease the interval distance between source views. Although the constructed local fea-

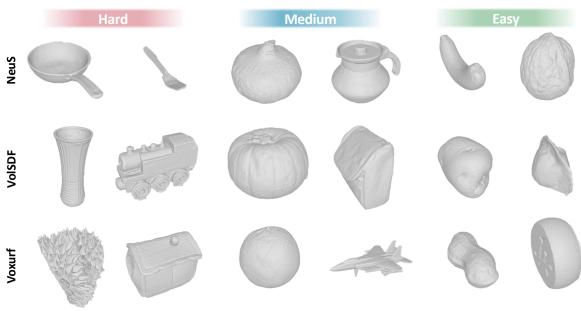


Figure S8. Examples from different difficulty levels in surface reconstruction.

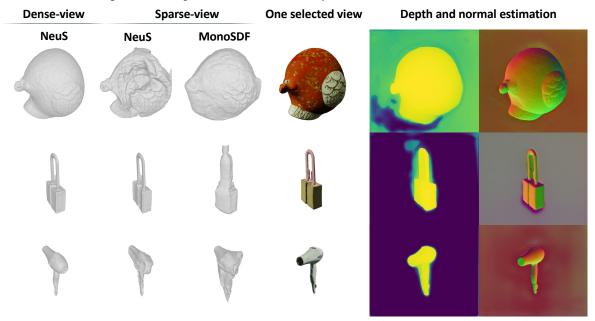


Figure S9. A comparison of sparse-view surface reconstruction between NeuS and MonoSDF. The estimated depth and normal maps used by MonoSDF are shown on the right.

ture will accordingly be more consistent as the occlusion region reduces, it will encode less source context. To make a trade-off between feature consistency and richness of encoded information, we conduct a comparison on how the extracted mesh will perform with the number of the nearest source views in FPS on 15 random categories from three levels of "difficulty". We filter the categories with averaged CD \geq 0.04, whose geometries are too poor to rely on. Finally, we remain 5 classes as shown in Figure S10. The geometric quality shows a fluctuating trend of decreasing and

then increasing with regard to the view range. As a result, we pick up "30" as a proper view selection range. Similarly, we find that "30" can also be applied to SparseNeuS [42] for cascaded geometry volume construction.

C.4. 3D Object Generation

Implementation Details. We use the official code by GET3D [23] to train all the models. We prepare the multiview image dataset by rendering 24 inward-facing multiview images per object with Blender [14]. For the large

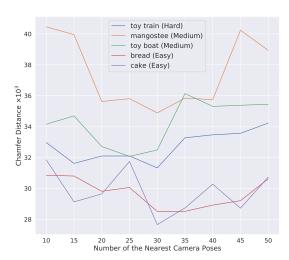


Figure S10. Geometric quality with regard to view selection range.

subset with 100 categories, we train 7k iterations with MSE loss and Adam optimizer; we train 3k iterations on smaller subsets (*e.g.*, *furniture*, *fruits*, and *toys*).

Additional Experimental Results and Discussions. We study the semantic distribution in the main text, where we use KMeans to cluster 100 random categories into 8 groups, as shown in Figure S11. We can observe that Group 2 has the largest number of categories, while they suffer from a high inner-group divergence (e.g., the peanut, handbag, mushroom, and hot dog). In contrast, Group 1 contains many fruits, vegetables, and some other categories that are similar in shape. The high inner-group similarity enables them to enhance the learning of each other, and Group 1 is finally able to dominate the generation distribution. The Group-level analysis reveals how cross-class relationships affect the generation distribution, which is a critical factor for generative models trained with large vocabulary datasets like OmniObject3D. We also provide the distribution of the four subsets used in this section in Figure S12.

Finally, we provide disentangled interpolation results in Figure S13 with geometry latent code and texture latent code, respectively. In the first row, the texture changes with a fixed shape, and the semantic changes accordingly. In the second row, when the geometry changes, the texture is fixed at first while encountering a substantial change along with the geometry at the end. This indicates that the two factors are not fully disentangled, and the geometry code can sometimes affect the texture since the category, geometry, and texture are highly correlated with each other in the dataset. Meanwhile, we observe that complex textures (*e.g.*, the cover of a book) usually fail to be well generated, which is another challenging problem to be explored in the future.

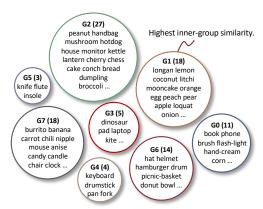


Figure S11. Categories in each group after the KMeans clustering. Categories in Group 1 are highly similar to each other, while those in Group 2 bear a high inner-group divergence.

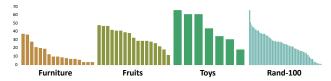


Figure S12. Distributions of the four subsets.



Figure S13. **Shape Interpolation.** In the first row, we keep the latent code of geometry fixed and interpolate the latent code of texture; in the second row, we keep the latent code of texture fixed and interpolate the latent code of geometry.

References

- [1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *Proceedings of the Interna*tional Conference on Machine learning (ICML), pages 40– 49, 2018. 3
- [2] Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7822–7831, 2021. 1
- [3] Matan Atzmon, Niv Haim, Lior Yariv, Ofer Israelov, Haggai Maron, and Yaron Lipman. Controlling neural level sets. In *Advances in Neural Information Processing Systems (NIPS)*, volume 32, 2019. 2
- [4] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neu-

- ral radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5855–5864, 2021. 1, 3
- [5] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5470–5479, 2022. 1, 3
- [6] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 16123–16133, 2022.
- [7] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 5799–5809, 2021. 3
- [8] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Proceedings* of the European Conference on Computer Vision (ECCV), 2022.
- [9] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference* on Computer Vision (ICCV), pages 14124–14133, 2021. 1, 5, 8
- [10] Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. In *Advances in Neural Information Processing Systems (NIPS)*, volume 32, 2019. 3
- [11] Yunlu Chen, Vincent Tao Hu, Efstratios Gavves, Thomas Mensink, Pascal Mettes, Pengwan Yang, and Cees GM Snoek. Pointmixup: Augmentation for point clouds. In *Proceedings of the European Conference on Computer Vision* (ECCV), pages 330–345, 2020. 1
- [12] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (CVPR), pages 5939–5948, 2019. 2
- [13] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF* Conference on Computer Vision and Pattern Recognition (CVPR), pages 5939–5948, 2019. 3
- [14] Blender Online Community. Blender a 3d modelling and rendering package. 2018. 1, 2, 5, 9
- [15] François Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. Improving neural implicit surfaces geometry with patch warping. arXiv.org, 2112.09648, 2021. 2
- [16] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In Advances in Neural Information Processing Systems (NIPS), volume 34, pages 8780– 8794, 2021. 3

- [17] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10786–10796, 2021. 8
- [18] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 12873–12883, 2021.
- [19] Marco Forte and François Pitié. F, b, alpha matting. arXiv.org, 2003.07711, 2020.
- [20] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5501–5510, 2022. 1, 3
- [21] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3d shape induction from 2d views of multiple objects. In *Proceedings* of the International Conference on 3D Vision (3DV), pages 402–411, 2017. 3
- [22] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. arXiv.org, 2209.11163, 2022. 3
- [23] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. In Advances in Neural Information Processing Systems (NIPS), 2022. 9
- [24] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. Revisiting point cloud shape classification with a simple and effective baseline. In *Proceedings of the International Conference on Machine learning (ICML)*, pages 3809–3820, 2021. 3
- [25] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. Proceedings of the International Conference on Learning Representations (ICLR), 2022. 3
- [26] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, 2021. 3
- [27] Zekun Hao, Arun Mallya, Serge Belongie, and Ming-Yu Liu. Gancraft: Unsupervised 3d neural rendering of minecraft worlds. In *Proceedings of the IEEE/CVF International Con*ference on Computer Vision (ICCV), pages 14072–14082, 2021. 3
- [28] Philipp Henzler, Niloy J Mitra, and Tobias Ritschel. Escaping plato's cave: 3d shape from adversarial rendering. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 9984–9993, 2019. 3
- [29] Xun Huang, Arun Mallya, Ting-Chun Wang, and Ming-Yu Liu. Multimodal conditional image synthesis with productof-experts gans. In *Proceedings of the European Conference* on Computer Vision (ECCV), pages 91–109, 2022. 3

- [30] Moritz Ibing, Gregor Kobsik, and Leif Kobbelt. Octree transformer: Autoregressive 3d shape generation on hierarchically structured sequences. *arXiv.org*, 2111.12480, 2022. 3
- [31] Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1251–1261, 2020. 2
- [32] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Aliasfree generative adversarial networks. In Advances in Neural Information Processing Systems (NIPS), volume 34, pages 852–863, 2021. 3
- [33] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 4401–4410, 2019. 3
- [34] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8110–8119, 2020. 3
- [35] Petr Kellnhofer, Lars C Jebe, Andrew Jones, Ryan Spicer, Kari Pulli, and Gordon Wetzstein. Neural lumigraph rendering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 4287– 4297, 2021. 2
- [36] Sihyeon Kim, Sanghyeok Lee, Dasol Hwang, Jaewon Lee, Seong Jae Hwang, and Hyunwoo J Kim. Point cloud augmentation with weighted local transformations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 548–557, 2021. 1
- [37] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv.org, 1412.6980, 2014. 6
- [38] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 2019–2028, 2020. 2
- [39] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF Conference* on Computer Vision and Pattern Recognition (CVPR), pages 8895–8904, 2019. 3
- [40] Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 7824– 7833, 2022.
- [41] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv.org*, 1906.07751, 2019. 2
- [42] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. Sparseneus: Fast generalizable neu-

- ral surface reconstruction from sparse views. *arXiv.org*, 2206.05737, 2022. 3, 8, 9
- [43] Sebastian Lunz, Yingzhen Li, Andrew Fitzgibbon, and Nate Kushman. Inverse graphics gan: Learning to generate 3d shapes from unstructured 2d data. arXiv.org, 2002.12674, 2020. 3
- [44] Andrew Luo, Tianqin Li, Wen-Hao Zhang, and Tai Sing Lee. Surfgen: Adversarial 3d shape synthesis with explicit surface discriminators. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16238–16248, 2021. 3
- [45] 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 (CVPR)*, pages 4460–4470, 2019.
- [46] 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 (CVPR)*, pages 4460–4470, 2019. 3
- [47] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P Srinivasan, and Jonathan T Barron. Nerf in the dark: High dynamic range view synthesis from noisy raw images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 16190– 16199, 2022. 1
- [48] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 405–421, 2020. 1, 3
- [49] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy J. Mitra, and Leonidas J. Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. arXiv.org, 1908.00575, 2019. 3
- [50] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics, 2022. 1
- [51] Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. Polygen: An autoregressive generative model of 3d meshes. In *Proceedings of the International Conference* on Machine learning (ICML), pages 7220–7229, 2020. 3
- [52] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11453–11464, 2021. 3
- [53] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 3504–3515, 2020. 2
- [54] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance

- fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5589–5599, 2021. 2
- [55] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. Stylesdf: High-resolution 3d-consistent image and geometry generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 13503–13513, 2022. 3
- [56] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 165–174, 2019.
- [57] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference* on Computer Vision and Pattern Recognition (CVPR), pages 2337–2346, 2019. 3
- [58] Dario Pavllo, Jonas Kohler, Thomas Hofmann, and Aurelien Lucchi. Learning generative models of textured 3d meshes from real-world images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13879–13889, 2021. 3
- [59] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: deep learning on point sets for 3d classification and segmentation. corr abs/1612.00593 (2016). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660, 2017. 1, 3
- [60] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NIPS)*, volume 30, 2017. 3
- [61] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. *Pattern Recognition*, 106:107404, 2020.
- [62] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (ICCV), pages 10901–10911, 2021. 1
- [63] Jiawei Ren, Liang Pan, and Ziwei Liu. Benchmarking and analyzing point cloud classification under corruptions. In Proceedings of the International Conference on Machine learning (ICML), 2022. 1, 3
- [64] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 2304–2314, 2019.
- [65] Johannes L Schonberger and Jan-Michael Frahm. Structurefrom-motion revisited. In *Proceedings of the IEEE/CVF* Conference on Computer Vision and Pattern Recognition (CVPR), pages 4104–4113, 2016.

- [66] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *Advances in Neural Information Processing* Systems (NIPS), volume 33, pages 20154–20166, 2020. 3
- [67] Katja Schwarz, Axel Sauer, Michael Niemeyer, Yiyi Liao, and Andreas Geiger. Voxgraf: Fast 3d-aware image synthesis with sparse voxel grids. arXiv.org, 2206.07695, 2022. 3
- [68] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In Advances in Neural Information Processing Systems (NIPS), volume 32, 2019. 2
- [69] Edward J Smith and David Meger. Improved adversarial systems for 3d object generation and reconstruction. In Proceedings of the Conference on Robot Learning (CoRL), pages 87–96, 2017. 3
- [70] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 5459–5469, 2022. 1
- [71] Saeid Asgari Taghanaki, Jieliang Luo, Ran Zhang, Ye Wang, Pradeep Kumar Jayaraman, and Krishna Murthy Jatavallabhula. Robustpointset: A dataset for benchmarking robustness of point cloud classifiers. arXiv.org, 2011.11572, 2020.
- [72] Briac Toussaint, Maxime Genisson, and Jean-Sébastien Franco. Fast Gradient Descent for Surface Capture Via Differentiable Rendering. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 1–10, 2022. 2
- [73] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1588–1597, 2019. 1
- [74] Mukund Varma, Peihao Wang, Xuxi Chen, Tianlong Chen, Subhashini Venugopalan, Zhangyang Wang, and Madras. Is attention all nerf needs? *arXiv.org*, 2207.13298, 2022. 1
- [75] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 5481–5490, 2022. 1
- [76] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Advances in Neural Information Processing Systems (NIPS)*, volume 34, pages 27171–27183, 2021. 2, 8
- [77] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Pro*ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 4690–4699, 2021. 1, 5

- [78] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics, 38(5):1–12, 2019. 1, 3
- [79] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In Advances in Neural Information Processing Systems (NIPS), volume 29, 2016. 3
- [80] Tong Wu, Jiaqi Wang, Xingang Pan, Xudong Xu, Christian Theobalt, Ziwei Liu, and Dahua Lin. Voxurf: Voxel-based efficient and accurate neural surface reconstruction. arXiv.org, 2208.12697, 2022. 2, 3, 8
- [81] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 1912–1920, 2015. 1
- [82] Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In *Proceedings of the IEEE/CVF In*ternational Conference on Computer Vision (ICCV), pages 915–924, 2021. 3
- [83] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *Proceedings of* the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 3173–3182, 2021. 3
- [84] Mutian Xu, Junhao Zhang, Zhipeng Zhou, Mingye Xu, Xiaojuan Qi, and Yu Qiao. Learning geometry-disentangled representation for complementary understanding of 3d object point cloud. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, volume 35, pages 3056–3064, 2021. 3
- [85] Yinghao Xu, Sida Peng, Ceyuan Yang, Yujun Shen, and Bolei Zhou. 3d-aware image synthesis via learning structural and textural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18430–18439, 2022. 3
- [86] 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 (ICCV)*, pages 4541–4550, 2019. 3
- [87] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Advances in Neural Information Processing Systems (NIPS)*, volume 34, pages 4805–4815, 2021. 2, 3, 8
- [88] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *Advances in Neural Information Processing Systems (NIPS)*, volume 33, pages 2492–2502, 2020. 2
- [89] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer*

- Vision and Pattern Recognition (CVPR), pages 4578–4587, 2021. 1, 5, 6
- [90] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. In Advances in Neural Information Processing Systems (NIPS), 2022. 3, 8
- [91] Jingyang Zhang, Yao Yao, and Long Quan. Learning signed distance field for multi-view surface reconstruction. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 6525–6534, 2021. 2
- [92] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5826–5835, 2021. 3
- [93] Peng Zhou, Lingxi Xie, Bingbing Ni, and Qi Tian. Cips-3d: A 3d-aware generator of gans based on conditionallyindependent pixel synthesis. arXiv.org, 2110.09788, 2021.